



SUSE Linux Enterprise Desktop 15 SP5

# 管理指南

## 管理指南

### SUSE Linux Enterprise Desktop 15 SP5

本指南介绍系统管理任务，如维护、监视和自定义初始安装的系统。

出版日期：2025 年 12 月 11 日

<https://documentation.suse.com> 

版权所有 © 2006–2025 SUSE LLC 和贡献者。保留所有权利。

根据 GNU 自由文档许可 (GNU Free Documentation License) 版本 1.2 或 (根据您的选择) 版本 1.3 中的条款，在此授予您复制、分发和/或修改本文档的权限；本版权声明和许可附带不可变部分。许可版本 1.2 的副本包含在题为“GNU Free Documentation License”的部分。

有关 SUSE 商标, 请参见 <http://www.suse.com/company/legal/>。所有第三方商标均是其各自所有者的财产。商标符号 (®、™ 等) 代表 SUSE 及其关联公司的商标。星号 (\*) 代表第三方商标。

本指南力求涵盖所有细节, 但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

# 目录

## 前言 **xxi**

1 可用文档 **xxi**

2 改进文档 **xxi**

3 文档约定 **xxii**

4 支持 **xxiv**

SUSE Linux Enterprise Desktop 支持声明 **xxiv** • 技术预览 **xxv**

## I 常用任务 **1**

### 1 Bash 和 Bash 脚本 **2**

1.1 什么是“外壳”？ **2**

Bash 配置文件 **2** • 目录结构 **5**

1.2 编写外壳脚本 **9**

1.3 重定向命令事件 **10**

1.4 使用别名 **11**

1.5 在 Bash 中使用变量 **12**

使用参数变量 **13** • 使用变量替代项 **13**

1.6 分组和组合命令 **14**

1.7 使用通用流程构造 **15**

if 控制命令 **15** • 使用 **for** 命令创建循环 **16**

1.8 更多信息 **16**

## 2 sudo 基础知识 18

### 2.1 sudo 基本用法 18

运行单个命令 18 · 启动外壳 19

### 2.2 配置 sudo 20

编辑配置文件 20 · sudoers 基本配置语法 21 · 基本 sudoers 规则 22

### 2.3 sudo 使用案例 24

在无需提供 root 口令的情况下使用 sudo 24 · 对 X.Org 应用程序使用 sudo 26

### 2.4 更多信息 26

## 3 使用 YaST 27

### 3.1 YaST 界面概述 27

### 3.2 有用的组合键 27

## 4 文本模式下的 YaST 29

### 4.1 在模块中导航 30

### 4.2 高级组合键 31

### 4.3 组合键的限制 32

### 4.4 YaST 命令行选项 32

从命令行安装软件包 32 · 使用单个模块 33 · YaST 模块的命令行参数 33

## 5 使用 YaST 更改语言和国家/地区设置 60

### 5.1 更改系统语言 60

使用 YaST 修改系统语言 61 · 切换默认系统语言 62 · 切换标准 X 和 GNOME 应用程序的语言 63

### 5.2 更改国家/地区和时间设置 63

## 6 使用 YaST 管理用户 66

- 6.1 用户和组管理对话框 66
- 6.2 管理用户帐户 68
- 6.3 其他用户帐户选项 69
  - 自动登录和无口令登录 69 • 强制实施口令策略 70 • 管理配额 71
- 6.4 更改本地用户的默认设置 73
- 6.5 将用户指派到组 73
- 6.6 管理组 74
- 6.7 更改用户身份验证方法 75
- 6.8 默认系统用户 76

## 7 YaST 联机更新 79

- 7.1 联机更新对话框 80
- 7.2 安装补丁 81
- 7.3 查看已撤回的补丁 82
- 7.4 自动联机更新 83

## 8 安装或去除软件 85

- 8.1 术语定义 85
- 8.2 注册已安装的系统 86
  - 使用 YaST 注册 86 • 使用 SUSEConnect 注册 87
- 8.3 使用 YaST 软件管理器 87
  - 搜索软件 88 • 安装和去除软件包或软件集 89 • 更新软件包 91 • 软件包依赖项 92 • 处理软件包建议 93
- 8.4 管理软件储存库和服务 94
  - 添加软件储存库 94 • 管理储存库属性 96 • 管理储存库密钥 97

- 8.5 GNOME Package Updater 97
- 8.6 使用 GNOME Software 更新软件包 100

## 9 使用命令行工具管理软件 101

- 9.1 使用 Zypper 101
  - 一般使用 101 · 使用 Zypper 子命令 102 · 使用 Zypper 安装和去除软件 103 · 使用 Zypper 更新软件 108 · 识别使用已删除文件的进程和服务 114 · 使用 Zypper 管理储存库 116 · 使用 Zypper 查询储存库和软件包 118 · 显示生命周期信息 120 · 配置 Zypper 121 · 查错 121 · Btrfs 文件系统上的 Zypper 回滚功能 122 · 更多信息 122
- 9.2 RPM - 软件包管理器 122
  - 校验软件包真实性 123 · 管理软件包：安装、更新和卸装 123 · 增量 RPM 软件包 124 · RPM 查询 125 · 安装和编译源软件包 128 · 使用 build 编译 RPM 软件包 130 · 用于 RPM 存档和 RPM 数据库的工具 130

## 10 通过 Snapper 进行系统恢复和快照管理 132

- 10.1 默认设置 133
  - 默认设置 134 · 快照类型 134 · 从快照中排除的目录 135 · 自定义设置 135
- 10.2 使用 Snapper 撤销更改 139
  - 撤销 YaST 和 Zypper 更改 140 · 使用 Snapper 恢复文件 145
- 10.3 通过从快照引导来执行系统回滚 147
  - 回滚后的快照 148 · 访问和识别快照引导项 149 · 限制 151
- 10.4 在用户主目录中启用 Snapper 152
  - 安装 pam\_snapper 和创建用户 152 · 去除用户 153 · 手动启用主目录中的快照 153
- 10.5 创建并修改 Snapper 配置 154
  - 管理现有配置 155

|      |                   |               |                 |                 |                |
|------|-------------------|---------------|-----------------|-----------------|----------------|
| 10.6 | 手动创建和管理快照 159     |               |                 |                 |                |
|      | 快照元数据 159         | • 创建快照 161    | • 修改快照元数据 162   | • 删除快照 162      |                |
| 10.7 | 快照的自动清理 164       |               |                 |                 |                |
|      | 清理编号快照 164        | • 清理时间线快照 166 | • 清理无差异的快照对 167 | • 清理手动创建的快照 167 | • 添加磁盘配额支持 168 |
| 10.8 | 显示快照使用的独占磁盘空间 169 |               |                 |                 |                |
| 10.9 | 常见问题解答 171        |               |                 |                 |                |

## 11 使用 KLP 的在线内核修补 173

|      |                      |                 |
|------|----------------------|-----------------|
| 11.1 | 内核实时修补的优势 173        |                 |
| 11.2 | 内核实时修补概述 173         |                 |
|      | 内核实时修补范围 174         | • 内核实时修补局限性 175 |
| 11.3 | 使用 YaST 激活内核实时修补 175 |                 |
| 11.4 | 从命令行激活内核实时修补 176     |                 |
| 11.5 | 执行内核实时修补 176         |                 |
|      | 检查实时补丁的失效日期 177      |                 |
| 11.6 | 内核实时修补问题查错 178       |                 |
|      | 手动降级补丁 178           |                 |

## 12 用户空间实时修补 179

|      |                |            |                  |
|------|----------------|------------|------------------|
| 12.1 | 关于用户空间实时修补 179 |            |                  |
|      | 先决条件 179       | • 支持的库 180 | • 使用 libpulp 180 |
| 12.2 | 更多信息 182       |            |                  |

## 13 事务更新 183

|      |             |
|------|-------------|
| 13.1 | 技术预览的限制 183 |
|------|-------------|

- 13.2 启用 `transactional-update` 185
- 13.3 管理自动更新 185
- 13.4 `transactional-update` 命令 186
- 13.5 查错 188

## 14 使用 VNC 的远程图形会话 189

- 14.1 `vncviewer` 客户端 189
  - 使用 `vncviewer` CLI 进行连接 189 · 使用 `vncviewer` GUI 进行连接 190 · 未加密连接通知 190
- 14.2 Remmina：远程桌面客户端 190
  - 安装 190 · 主窗口 191 · 添加远程会话 191 · 启动远程会话 193 · 编辑、复制和删除保存的会话 194 · 从命令行运行远程会话 194
- 14.3 在 VNC 服务器上配置一次性会话 195
  - 可用配置 196 · 启动一次性 VNC 会话 197 · 配置一次性 VNC 会话 197
- 14.4 配置永久 VNC 服务器会话 198
  - 使用 `vncserver` 启动的 VNC 会话 198 · 使用 `vncmanager` 启动的 VNC 会话 200
- 14.5 在 VNC 服务器上配置加密 203
- 14.6 与 Wayland 的兼容性 205

## 15 使用 RSync 复制文件 206

- 15.1 概念概述 206
- 15.2 基本语法 206
- 15.3 在本地复制文件和目录 207
- 15.4 远程复制文件和目录 208

15.5 配置和使用 Rsync 服务器 208

15.6 更多信息 211

## II 引导 LINUX 系统 212

# 16 引导过程简介 213

16.1 术语 213

16.2 Linux 引导进程 214

初始化和引导加载程序阶段 214 • 内核阶段 215 • Init on initramfs 阶段 218 • systemd 阶段 220

# 17 UEFI (统一可扩展固件接口) 221

17.1 安全引导 221

在 SUSE Linux Enterprise Desktop 上实施 222 • MOK (机器拥有者密钥) 224 • 引导自定义内核 224 • 使用非内置驱动程序 227 • 功能和限制 227

17.2 更多信息 228

# 18 引导加载程序 GRUB 2 229

18.1 GRUB Legacy 与 GRUB 2 之间的主要差异 229

18.2 配置文件结构 229

文件 /boot/grub2/grub.cfg 230 • 文件 /etc/default/grub 231 • /etc/grub.d 中的脚本 234 • BIOS 驱动器与 Linux 设备之间的映射 235 • 在引导过程中编辑菜单项 235 • 设置引导命令 237 • 引导菜单项的授权访问权限 238

18.3 使用 YaST 配置引导加载程序 239

引导加载程序位置和引导代码选项 240 • 调整磁盘顺序 242 • 配置高级选项 242

18.4 有用的 GRUB 2 命令 245

18.5 更多信息 247

## 19 systemd 守护程序 248

19.1 systemd 概念 248

单元文件 249

19.2 基本用法 249

管理正在运行的系统中的服务 250 • 永久启用/禁用服务 252

19.3 系统启动和目标管理 253

目标与运行级别的比较 253 • 调试系统的启动 257 • System V 兼容性 260

19.4 使用 YaST 管理服务 261

19.5 自定义 systemd 262

自定义单元文件 262 • 创建插入式文件 263 • 将 xinetd 服务转换为 systemd 264 • 创建自定义目标 265

19.6 高级用法 266

清理临时目录 266 • 系统日志 267 • 快照 267 • 装载内核模块 268 • 装载服务之前执行操作 268 • 内核控制组 (cgroup) 269 • 终止服务 (发送信号) 270 • 有关 D-Bus 服务的重要说明 271 • 调试服务 271

19.7 systemd 计时器单元 273

systemd 计时器类型 273 • systemd 计时器和服务单元 273 • 实际示例 273 • 管理 systemd 计时器 275

19.8 更多信息 275

## III 系统 277

## 20 64 位系统环境中的 32 位和 64 位应用程序 278

20.1 运行时支持 278

20.2 内核规范 279

## 21 **journalctl**: 查询 systemd 日志 280

21.1 将日志设为永久 280

21.2 **journalctl**: 有用的开关 281

21.3 过滤日志输出 282

    根据引导编号过滤 282 · 根据时间间隔过滤 282 · 根据字段过滤 283

21.4 调查 systemd 错误 284

21.5 Journald 配置 285

    更改日志大小限制 286 · 将日志转发到 /dev/ttyX 286 · 将日志转发到系统日志工具 286

21.6 使用 YaST 过滤 systemd 日志 287

21.7 在 GNOME 中查看日志 287

## 22 **update-alternatives**: 管理命令和文件的多个版本 288

22.1 概览 288

22.2 使用案例 289

22.3 获取备选项的概览 290

22.4 查看有关特定备选项的细节 290

22.5 设置备选项的默认版本 291

22.6 安装自定义备选项 292

22.7 定义依赖备选项 294

## 23 基本网络知识 296

- 23.1 IP 地址和路由 298
  - IP 地址 298 • 网络掩码和路由 299
- 23.2 IPv6 - 新一代互联网 300
  - 优点 301 • 地址类型和结构 302 • IPv4 与 IPv6 并存 306 • 配置 IPv6 307 • 更多信息 307
- 23.3 名称解析 308
- 23.4 使用 YaST 配置网络连接 309
  - 使用 YaST 配置网卡 309
- 23.5 NetworkManager 319
  - NetworkManager 和 **wicked** 320 • NetworkManager 功能和配置文件 321 • 控制和锁定 NetworkManager 功能 321
- 23.6 手动配置网络连接 321
  - wicked** 网络配置 321 • 配置文件 329 • 测试配置 340 • 单元文件和启动脚本 344
- 23.7 设置绑定设备 345
  - 绑定端口的热插拔 347
- 23.8 设置组合设备以进行网络协作 349
  - 用例：网络协作的负载平衡 352 • 用例：使用网络协作实现故障转移 353 • 用例：组合设备上的 VLAN 354

## 24 打印机操作 357

- 24.1 CUPS 工作流程 358
- 24.2 连接打印机的方法和协议 359
- 24.3 安装软件 359
- 24.4 网络打印机 360
- 24.5 使用命令行工具配置 CUPS 361

|      |                                      |     |
|------|--------------------------------------|-----|
| 24.6 | 从命令行打印                               | 362 |
| 24.7 | SUSE Linux Enterprise Desktop 中的特殊功能 | 363 |
|      | CUPS 和防火墙                            | 363 |
|      | · 浏览网络打印机                            | 363 |
|      | · 各个软件包中的 PPD 文                      |     |
|      | 件                                    | 364 |
| 24.8 | 查错                                   | 365 |
|      | 不支持标准打印机语言的打印机                       | 365 |
|      | · 没有合适的 PPD 文件可用于                    |     |
|      | PostScript 打印机                       | 365 |
|      | · 网络打印机连接                            | 366 |
|      | · 打印输出有缺                             |     |
|      | 陷但不显示错误消息                            | 368 |
|      | · 禁用的队列                              | 368 |
|      | · CUPS 浏览：删除                         |     |
|      | 打印作业                                 | 369 |
|      | · 有问题的打印作业和数据传输错误                    | 369 |
|      | · 调试                                 |     |
|      | CUPS                                 | 370 |
|      | · 更多信息                               | 370 |

## 25 图形用户界面 371

|      |   |     |
|------|---|-----|
| 25.1 | X Window 系统                                     | 371 |
| 25.2 | 安装和配置字体   | 371 |
|      | 显示安装的字体   | 372 |
|      | · 查看字体  | 373 |
|      | · 查询字体  | 373 |
|      | · 安装字   |     |
|      | 体   | 374 |
|      | · 配置字体的外观                                       | 375 |
| 25.3 | 适用于管理员的 GNOME 配置                                | 384 |
|      | dconf 系统  | 384 |
|      | · 系统范围的配置                                       | 384 |
|      | · 在 Wayland 上管理平                                |     |
|      | 铺   | 385 |
|      | · 更多信息  | 387 |
| 25.4 | 使用 SUSE Prime 在 Intel 和 NVIDIA Optimus GPU 之间切换 | 387 |
|      | 先决条件  | 388 |
|      | · 安装和使用 SUSE Prime                              | 388 |
|      | · 安装 NVIDIA 驱动程                                 |     |
|      | 序   | 389 |

## 26 使用 FUSE 访问文件系统 390

|      |            |     |
|------|------------|-----|
| 26.1 | 配置 FUSE    | 390 |
| 26.2 | 挂载 NTFS 分区 | 390 |
| 26.3 | 更多信息       | 391 |

## 27 安装多个内核版本 392

### 27.1 启用和配置多版本支持 392

自动删除未使用的内核 393 · 用例：仅在重引导后删除旧内核 394 · 用例：保留较旧内核作为后备 394 · 用例：保留特定的内核版本 395

### 27.2 使用 YaST 安装/去除多个内核版本 395

### 27.3 使用 Zypper 安装/去除多个内核版本 396

## 28 管理内核模块 399

### 28.1 使用 lsmod 和 modinfo 列出装载的模块 399

### 28.2 添加和去除内核模块 400

引导时自动装载内核模块 400 · 使用 modprobe 将内核模块列入黑名单 401

## 29 使用 udev 进行动态内核设备管理 403

### 29.1 /dev 目录 403

### 29.2 内核 uevents 和 udev 403

### 29.3 驱动程序、内核模块和设备 404

### 29.4 引导和启动设备设置 404

### 29.5 监视正在运行的 udev 守护程序 405

### 29.6 使用 udev 规则影响内核设备事件处理 406

在 udev 规则中使用运算符 408 · 在 udev 规则中使用替换项 408 · 使用 udev 匹配键 409 · 使用 udev 赋值键 411

### 29.7 永久设备命名 412

### 29.8 udev 使用的文件 413

### 29.9 更多信息 414

## 30 特殊系统功能 415

- 30.1 特殊软件包的相关信息 415
  - bash 软件包和 /etc/profile 415 • cron 软件包 416 • 停止 Cron 状态消息 417 • 日志文件：软件包 logrotate 417 • **locate** 命令 417 • **ulimit** 命令 417 • **free** 命令 419 • 手册页和信息页 419 • 使用 **man** 命令选择手册页 419 • GNU Emacs 的设置 420
- 30.2 虚拟控制台 421
- 30.3 键盘映射 421
- 30.4 语言和国家/地区特定的设置 422
  - 系统范围的区域设置 423 • 一些示例 424 • ~/i18n 中的区域设置 425 • 语言支持的设置 425 • 更多信息 426

## 31 使用 NetworkManager 427

- 31.1 NetworkManager 的用例 427
- 31.2 启用或禁用 NetworkManager 427
- 31.3 配置网络连接 428
  - 管理有线网络连接 430 • 管理无线网络连接 430 • 启用无线强制网络门户检测 431 • 将 Wi-Fi/蓝牙网卡配置为接入点 431 • NetworkManager 和 VPN 432
- 31.4 NetworkManager 和安全性 433
  - 用户和系统连接 434 • 存储口令和身份凭证 434 • 防火墙区域 434
- 31.5 常见问题解答 435
- 31.6 查错 436
- 31.7 更多信息 437

IV 硬件配置 438

**32 设置系统键盘布局 439**

**33 设置声卡 440**

**34 设置打印机 443**

34.1 配置打印机 443

使用 YaST 添加驱动程序 444 · 编辑本地打印机配置 445

34.2 使用 YaST 配置网络打印 446

使用 CUPS 446 · 使用 CUPS 之外的打印服务器 447

34.3 通过网络共享打印机 447

**35 设置扫描仪 449**

35.1 配置 HP 多功能一体机 449

35.2 通过网络共享扫描仪 450

35.3 通过网络进行扫描 450

**36 电源管理 451**

36.1 省电功能 451

36.2 高级配置和电源接口 (ACPI) 452

控制 CPU 性能 452 · 查错 453

36.3 硬盘的休眠 454

36.4 查错 456

CPU 频率功能不工作 456

**37 持久内存 457**

37.1 简介 457

37.2 术语 459

|      |  |
|------|--|
| 37.3 | 使用案例 461   |
|      | 将 PMEM 与 DAX 搭配使用 461 • 将 PMEM 与 BTT 搭配使用 462  |
| 37.4 | 用于管理持久内存的工具 462  |
| 37.5 | 设置持久内存 463   |
|      | 查看可用的 NVDIMM 存储 463 • 将存储配置为使用 DAX 的单个 PMEM 名称空间 465 • 创建使用 BTT 的 PMEM 名称空间 467 • 将文件系统日志置于 PMEM/BTT 上 468 |
| 37.6 | 更多信息 469   |

## V 服务 470

# 38 使用 YaST 进行服务管理 471

## 39 使用 NTP 同步时间 473

|      |   |
|------|---|
| 39.1 | 使用 YaST 配置 NTP 客户端 473                    |
|      | NTP 守护程序启动 474 • 配置源的类型 474 • 配置时间服务器 475 |
| 39.2 | 手动配置网络中的 NTP 476                          |
| 39.3 | 在运行时使用 <b>chronyc</b> 配置 chronyd 477      |
| 39.4 | 运行时的动态时间同步 478                            |
| 39.5 | 设置本地参考时钟 478                              |
| 39.6 | 与外部时间参考 (ETR) 的时钟同步 479                   |

## VI 查错 480

# 40 帮助和文档 481

|      |                         |
|------|-------------------------|
| 40.1 | 文档目录 481                |
|      | SUSE 手册 482 • 软件包文档 482 |
| 40.2 | 手册页 483                 |

40.3 信息页 484

40.4 联机资源 484

## 41 收集系统信息以供支持所用 486

41.1 显示当前系统信息 486

41.2 使用 Supportconfig 收集系统信息 487

    创建服务请求编号 487 · 上载目标 488 · 使用 YaST 创建 supportconfig 存档 488 · 从命令行创建 supportconfig 存档 491 · 了解 **supportconfig** 的输出 491 · 常用的 supportconfig 选项 493 · 存档内容概述 494

41.3 将信息提交到全球技术支持部门 497

41.4 分析系统信息 499

    SCA 命令行工具 500 · SCA 设备 501 · 开发自定义分析模式 512

41.5 在安装期间收集信息 513

41.6 内核模块支持 513

    技术背景 514 · 使用不支持的模块 514

41.7 更多信息 515

## 42 常见问题及其解决方案 517

42.1 查找和收集信息 517

42.2 引导问题 519

    GRUB 2 引导加载程序无法装载 520 · 不显示登录或提示界面 520 · 无图形登录 521 · 无法挂载 Btrfs 根分区 521 · 强制检查根分区 522 · 禁用交换以启用引导 522 · GRUB 2 在双引导系统上重引导时失败 522

42.3 登录问题 522

    有效的用户名和口令组合失败 522 · 有效的用户名和口令不被接受 524 · 登录至加密的主分区失败 525 · GNOME 桌面有问题 526

42.4 网络问题 527

NetworkManager 问题 530

42.5 数据问题 531

管理分区映像 531 · 使用救援系统 532

**A 网络示例 539**

**B GNU licenses 540**

# 前言

## 1 可用文档

### 联机文档

可在 <https://documentation.suse.com> 上查看我们的联机文档。您可浏览或下载各种格式的文档。



### 注意：最新更新

最新的更新通常会在本文档的英文版中提供。

### 发行说明

有关发行说明，请参见 <https://www.suse.com/releasenotes/>。

### 在您的系统中

要以脱机方式使用，请参见安装的系统中 `/usr/share/doc` 下的文档。许多命令的 **手册页** 中也对相应命令进行了详细说明。要查看手册页，请运行 `man` 后跟特定的命令名。如果系统上未安装 `man` 命令，请使用 `sudo zypper install man` 加以安装。

## 2 改进文档

欢迎您提供针对本文档的反馈及改进建议。您可以通过以下渠道提供反馈：

### 服务请求和支持

有关产品可用的服务和支持选项，请参见 <http://www.suse.com/support/>。

要创建服务请求，需在 SUSE Customer Center 中注册订阅的 SUSE 产品。请转到 <https://scc.suse.com/support/requests> 并登录，然后点击新建。

### Bug 报告

在 <https://bugzilla.suse.com/> 中报告文档问题。

要简化此过程，请单击本文档 HTML 版本中的标题旁边的报告问题图标。这样会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。

需要一个 Bugzilla 帐户。

## 贡献

要帮助改进本文档，请单击本文档 HTML 版本中的标题旁边的 Edit Source document（编辑源文档）图标。然后您会转到 GitHub 上的源代码，可以在其中提出拉取请求。

需要一个 GitHub 帐户。



## 注意：Edit source document（编辑源文档）仅适用于英语版本

Edit source document（编辑源文档）图标仅适用于每个文档的英语版本。对于所有其他语言，请改用报告问题图标。

有关本文档使用的文档环境的详细信息，请参见储存库的 README（网址：<https://github.com/SUSE/doc-sle>）。

## 邮件

您也可以将有关本文档中的错误以及相关反馈发送至 [doc-team@suse.com](mailto:doc-team@suse.com)。请在其中包含文档标题、产品版本和文档发布日期。此外，请包含相关的章节号和标题（或者提供 URL），并提供问题的简要说明。

## 3 文档约定

本文档中使用了以下通知和排版约定：

- /etc/passwd: 目录名称和文件名
- PLACEHOLDER: 将 PLACEHOLDER 替换为实际值
- PATH: 环境变量
- ls、--help: 命令、选项和参数

- user: 用户或组的名称
- package\_name: 软件包的名称
- **Alt** 、 **Alt + F1** : 按键或组合键。按键以大写字母显示，与键盘上的一样。
- 文件、文件 > 另存为：菜单项，按钮
- Chapter 1, “Example chapter” : 对本指南中其他章节的交叉引用。
- 必须使用 root 特权运行的命令。您往往还可以在这些命令前加上 sudo 命令，以非特权用户身份来运行它们。

```
# command  
> sudo command
```

- 可以由非特权用户运行的命令。

```
> command
```

- 注意



### 警告：警报通知

在继续操作之前，您必须了解的不可或缺的信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。



### 重要：重要通知

在继续操作之前，您必须了解的重要信息。



### 注意：注意通知

额外信息，例如有关软件版本差异的信息。



## 提示：提示通知

有用信息，例如指导方针或实用性建议。

- 精简通知



额外信息，例如有关软件版本差异的信息。



有用信息，例如指导方针或实用性建议。

## 4 支持

下面提供了 SUSE Linux Enterprise Desktop 的支持声明和有关技术预览的一般信息。有关产品生命周期的细节，请参见 <https://www.suse.com/lifecycle>。

如果您有权获得支持，可在 <https://documentation.suse.com/sles-15/html/SLES-all/cha-adm-support.html> 中查找有关如何收集支持票据所需信息的细节。

### 4.1 SUSE Linux Enterprise Desktop 支持声明

要获得支持，您需要一个适当的 SUSE 订阅。要查看为您提供的具体支持服务，请转到 <https://www.suse.com/support/> 并选择您的产品。

支持级别的定义如下：

#### L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

#### L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题领域，并针对级别 1 不能解决的问题提供解决方法，或作为级别 3 的准备级别。

### L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约的客户与合作伙伴，SUSE Linux Enterprise Desktop 将为除以下软件包外的其他所有软件包提供 L3 支持：

- 技术预览。
- 声音、图形、字体和作品。
- 需要额外客户合同的软件包。
- 名称以 `-devel` 结尾的软件包（包含头文件和类似的开发人员资源）只能与其主软件包一起获得支持。

SUSE 仅支持使用原始软件包，即，未发生更改且未重新编译的软件包。

## 4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种软件包、堆栈或功能。随附这些技术预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈。如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们可能在功能上不完整、不稳定，或者不适合生产用途。
- 技术预览不受支持。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- SUSE 可能会发现某个预览不符合客户或市场需求，或者未遵循企业标准。技术预览可能会随时从产品中删除。SUSE 不承诺未来将提供此类技术的受支持版本。

有关产品随附的技术预览的概述，请参见 <https://www.suse.com/releasenotes> 上的发行说明。

# | 常用任务

- 1 Bash 和 Bash 脚本 2
- 2 **sudo** 基础知识 18
- 3 使用 YaST 27
- 4 文本模式下的 YaST 29
- 5 使用 YaST 更改语言和国家/地区设置 60
- 6 使用 YaST 管理用户 66
- 7 YaST 联机更新 79
- 8 安装或去除软件 85
- 9 使用命令行工具管理软件 101
- 10 通过 Snapper 进行系统恢复和快照管理 132
- 11 使用 KLP 的在线内核修补 173
- 12 用户空间实时修补 179
- 13 事务更新 183
- 14 使用 VNC 的远程图形会话 189
- 15 使用 RSync 复制文件 206

# 1 Bash 和 Bash 脚本

现今，许多人都在使用装有 GNOME 之类图形用户界面 (GUI) 的计算机。尽管 GUI 可提供许多功能，但执行自动化任务时，这些功能会受到限制。外壳是对 GUI 的有效补充，本章将会概述外壳（以 Bash 外壳为例）的数个方面。

## 1.1 什么是“外壳”？

通常来说，Linux 外壳就是 Bash (Bourne again Shell)。在本章中提到“外壳”时，指的是 Bash。除了 Bash 以外，还存在其他一些外壳（ash、csh、ksh、zsh 等），每种外壳都具备不同的功能和特征。如果需要关于其他外壳的更多信息，请在 YaST 中搜索外壳。

### 1.1.1 Bash 配置文件

外壳可调用为：

- 1. 交互式登录外壳：** 使用 `--login` 选项调用 Bash 来登录计算机时，或通过 SSH 登录到远程计算机时，需要使用此方式。
- 2. “普通”交互式外壳：** 在启动 xterm、konsole、gnome-terminal 或类似的命令行界面 (CLI) 工具时通常会调用此外壳。
- 3. 非交互式外壳：** 当在命令行调用外壳脚本时调用此外壳。

各个外壳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。



#### 提示

Bash 会根据运行配置文件的外壳类型按特定顺序查找其配置文件。有关详细信息，请参见 Bash 手册页 (`man 1 bash`)。搜索标题 `INVOCATION`。

表 1.1：登录外壳的 BASH 配置文件

| 文件                              | 说明                                      |
|---------------------------------|---|
| <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> | 如果要扩展 <code>/etc/profile</code> ，请使用此文件 |
| <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                       |

登录外壳还会获取表 1.2 “非登录外壳的 Bash 配置文件” 中所列的配置文件。

表 1.2：非登录外壳的 BASH 配置文件

|                                     |                             |
|-------------------------------------|-----------------------------|
| <code>/etc/bash.bashrc</code>       | 请勿修改此文件，否则下次更新时，您的修改可能会被损坏。 |
| <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               |

Bash 还会使用其他多个文件：

表 1.3：用于 BASH 的特殊文件

| 文件                           | 说明  |
|------------------------------|---|
| <code>~/.bash_history</code> | 包含已键入的所有命令的列表   |
| <code>~/.bash_logout</code>  | 注销时执行   |
| <code>~/.alias</code>        | 用户为常用命令定义的别名。有关定义别名的更多细节，请参见 <code>man 1 alias</code> 。 |

## 非登录外壳

下面两个特殊外壳会阻止用户登录到系统：/bin/false 和 /sbin/nologin。当用户尝试登录到系统时，这两个外壳都将失败且不会显示任何提示。这是针对系统用户有意设计的一种安全措施，虽然新式 Linux 操作系统提供了更有效的工具（例如 PAM 和 AppArmor）来控制系统访问。

SUSE Linux Enterprise Desktop 上的默认行为是将 /bin/bash 指派给人类用户，将 /bin/false 或 /sbin/nologin 指派给系统用户。但由于历史原因，nobody 用户拥有 /bin/bash，因为该用户过去是默认用做系统用户的最低特权用户。但是，如果有多个系统用户使用 nobody，则使用 nobody 而获得的任何一点安全性都将失去。应该可以将它更改为 /sbin/nologin；最快的测试方法是进行此更改，然后看看这样是否中断了任何服务或应用程序。

使用以下命令列出 /etc/passwd 中已指派给所有用户（系统用户和人类用户）的外壳。输出视您系统上的服务和用户而异：

```
> sort -t: -k 7 /etc/passwd | awk -F: '{print $1"\t" $7}' | column -t
tux          /bin/bash
nobody       /bin/bash
root         /bin/bash
avahi         /bin/false
chrony        /bin/false
dhcpd         /bin/false
dnsmasq       /bin/false
ftpsecure     /bin/false
lightdm       /bin/false
mysql         /bin/false
postfix       /bin/false
rtkit         /bin/false
sshd          /bin/false
tftp          /bin/false
unbound       /bin/false
bin           /sbin/nologin
daemon        /sbin/nologin
ftp           /sbin/nologin
lp            /sbin/nologin
mail          /sbin/nologin
man           /sbin/nologin
```

|                  |                   |
|------------------|-------------------|
| nscd             | /sbin/nologin     |
| polkitd          | /sbin/nologin     |
| pulse            | /sbin/nologin     |
| qemu             | /sbin/nologin     |
| radvd            | /sbin/nologin     |
| rpc              | /sbin/nologin     |
| statd            | /sbin/nologin     |
| svn              | /sbin/nologin     |
| systemd-coredump | /sbin/nologin     |
| systemd-network  | /sbin/nologin     |
| systemd-timesync | /sbin/nologin     |
| usbmux           | /sbin/nologin     |
| vnc              | /sbin/nologin     |
| wwwrun           | /sbin/nologin     |
| messagebus       | /usr/bin/false    |
| scard            | /usr/sbin/nologin |

## 1.1.2 目录结构

下表简要介绍 Linux 系统上最重要的较高级别目录。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 1.4：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。                              |

| 目录              | 内容  |
|-----------------|---|
| <u>/home</u>    | 存储系统上具有帐户的所有用户的用户主目录。但是, <u>root</u> 的主目录不在 <u>/home</u> 中, 而是在 <u>/root</u> 中。 |
| <u>/lib</u>     | 基本共享库和内核模块。   |
| <u>/media</u>   | 可卸媒体的挂载点。   |
| <u>/mnt</u>     | 临时挂载文件系统的挂载点。   |
| <u>/opt</u>     | 附加应用程序软件包。  |
| <u>/root</u>    | 超级用户 <u>root</u> 的主目录。  |
| <u>/sbin</u>    | 基本系统二进制文件。  |
| <u>/srv</u>     | 系统提供的服务的数据。   |
| <u>/tmp</u>     | 临时文件。   |
| <u>/usr</u>     | 具有只读数据的辅助层次结构。  |
| <u>/var</u>     | 变量数据, 如日志文件。  |
| <u>/windows</u> | 只在系统上同时安装了 Microsoft Windows* 和 Linux 时可用。包含 Windows 数据。                        |

以下列表提供有关这些目录中有哪些文件和子目录的更多详细信息, 并给出一些示例:

### /bin

包含 root 和其他用户都可使用的基本 shell 命令。这些命令包括

ls、mkdir、cp、mv、rm 和 rmdir。/bin 还包含 Bash, 后者是 SUSE Linux Enterprise Desktop 中的默认外壳。

### /boot

包含引导所需的数据, 如引导加载程序、内核以及内核开始执行用户模式程序之前使用的其他数据。

## /dev

存储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 子目录包含可在引导过程中执行的 LSB init 脚本。

## /home/USERNAME

存储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拥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形式存放在此处，例如 .gconf/ 和 .config。



### 注意：网络环境中的用户主目录

如果您是在网络环境中工作，您的用户主目录可能会映射到文件系统中除 /home 之外的其他目录。

## /lib

包含引导系统和运行 root 文件系统中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、闪存盘和数码相机（如果它们使用 USB）等可卸媒体的挂载点。/media 通常包含除系统硬盘之外的各类驱动器。可卸媒体插入或连接到系统并挂载之后，您便可从此处访问该媒体。

## /mnt

此目录提供临时挂载的文件系统的挂载点。root 可以在此处挂载文件系统。

## /opt

保留用于安装第三方软件。在此处可以找到可选软件和较大附加产品程序软件包。

## /root

root 用户的主目录。root 的个人数据存储在此处。

## /run

systemd 和各个组件使用的 tmpfs 目录。/var/run 是指向 /run 的符号链接。

## /sbin

如 s 所表明的，该目录存储超级用户的实用程序。/sbin 不仅包含 /bin 中的二进制文件，还包含引导、还原和恢复系统所需的其他二进制文件。

## /srv

存储系统提供的服务（如 FTP 和 HTTP）的数据。

## /tmp

此目录由需要临时存储文件的程序使用。

### **!** 重要：在引导时清理 /tmp

无法保证存储在 /tmp 中的数据在系统重引导后仍然存在。这取决于 /etc/tmpfiles.d/tmp.conf 中的设置及其他因素。

## /usr

/usr 与用户无关，而是 Unix 系统资源 (Unix system resource) 的缩写。/usr 中的数据是静态只读数据，可以在符合 Filesystem Hierarchy Standard (FHS) 的各个主机之间共享。此目录包含所有应用程序（包括 GNOME 等图形桌面），并且会在文件系统中创建次要层次。/usr 包含 /usr/bin、/usr/sbin、/usr/local 和 /usr/share/doc 等多个子目录。

### /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

存储系统的各种文档文件和发行描述。在 manual 子目录中可以找到此手册的联机版本。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` 下可以找到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`，通常用其存储该软件包的自述文件，有时还存储示例、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，其中提供了关于众多 Linux 软件安装和操作任务的附加文档。

## /var

`/usr` 用于存储静态只读数据，而 `/var` 用于存储系统运行期间写入的可变数据，例如日志文件或假脱机数据。如需大致了解 `/var/log/` 中包含的最重要的日志文件，请参见表 42.1 “日志文件”。

## /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。对于 NTFS，SUSE Linux Enterprise Desktop 还包含写访问权限支持。但是，NTFS-3g 文件系统的驱动程序只拥有部分功能。

## 1.2 编写外壳脚本

使用外壳脚本可以方便地完成各种任务：收集数据、在文本中搜索单词或短语，以及执行其他有用的操作。以下示例显示用于打印文本的小外壳脚本：

### 例 1.1：用于打印文本的外壳脚本

```
#!/bin/sh ①
# Output the following line: ②
echo "Hello World" ③
```

- ① 第一行以 **Shebang** 字符 (`#!`) 开头，表示此文件为脚本。在 **Shebang** 后面指定的解释器将执行该脚本。在本例中，指定的解释器为 `/bin/sh`。
- ② 第二行是一个以哈希符号开头的注释。我们建议对难以理解的行提供注释。提供适当的注释可以记住该行的用途和功能。另外，其他阅读者可以更好地理解您的脚本。在开发社区中，注释被视为一种良好的做法。

- ③ 第三行使用内置命令 `echo` 打印相应文本。

在运行此脚本之前，必须满足几个先决条件：

1. 每个脚本都应包含 Shebang 行（如上面的示例所示）。如果该行缺失，您需要手动调用解释器。
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找到的目录中。外壳中的搜索路径由环境变量 `PATH` 确定。普通用户对 `/usr/bin` 没有写入权限。因此，建议将脚本保存在用户目录 `~/bin/` 中。在上例中使用名称 `hello.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
```

如果满足上述所有先决条件，则可以采用如下方式执行此脚本：

1. **作为绝对路径：** 可以使用绝对路径执行脚本。在本例中为 `~/bin/hello.sh`。
2. **所有位置：** 如果 `PATH` 环境变量包含脚本所在目录，则可以使用 `hello.sh` 来执行该脚本。

## 1.3 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：

### Command > File

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

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

### Command > File

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
```

### Command < File

读取该文件作为给定命令的输入。例如，read 命令会将此文件的内容读入变量：

```
> read a < foo
```

### Command1 | Command2

将左侧命令的输出重定向为右侧命令的输入。例如，cat 命令会输出文件 /proc/cpuinfo 的内容。grep 使用此输出仅过滤出包含 cpu 的行：

```
> cat /proc/cpuinfo | grep cpu
```

每个通道都有一个文件描述符：0（零）表示标准输入，1表示标准输出，2表示标准错误。您可以在<或>字符前面插入此文件描述符。例如，下行会搜索以 foo 开头的文件，但通过将文件重定向到 /dev/null 来禁止显示错误：

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

## 1.4 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，下行定义了一个别名 lt，它会输出一份较长的列表（选项 -l），将列表按修改时间排序（-t），并按排好序的倒序打印（-r）：

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

要查看所有别名定义，请使用 alias。使用 unalias 和相应的别名删除别名。

## 1.5 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。如果需要知道变量的值，请将变量名称作为参数插入：

```
> printenv PATH
```

也可以使用 **echo** 查看变量（无论是全局或本地变量）：

```
> echo $PATH
```

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
```

下表包含可以在外壳脚本中使用的常见环境变量：

表 1.5：有用的环境变量

|             |  |
|-------------|--|
| <u>HOME</u> | 当前用户的用户主目录                               |
| <u>HOST</u> | 当前主机名                                    |
| <u>LANG</u> | 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 <u>C</u> |
| <u>PATH</u> | 外壳的搜索路径，冒号分隔的目录列表                        |
| <u>PS1</u>  | 指定在每个命令前打印的普通提示符                         |

|             |                    |
|-------------|--------------------|
| <u>PS2</u>  | 指定在执行多行命令时打印的辅助提示符 |
| <u>PWD</u>  | 当前工作目录             |
| <u>USER</u> | 当前用户               |

## 1.5.1 使用参数变量

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参数，您需要定位参数。这些参数为 \$1（表示第一个参数）、\$2（表示第二个参数），以此类推。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 的所有参数：

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

如果使用上述参数执行此脚本，将获取：

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

## 1.5.2 使用变量替代项

变量替换将一个模式应用于变量的内容（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
```

### `${VAR/pattern_1/pattern_2}`

将 PATTERN\_1 中 VAR 的内容替代为 PATTERN\_2:

```
> file=/home/tux/book/book.tar.bz2
> echo ${file/tux/wilber}
/home/wilber/book/book.tar.bz2
```

## 1.6 分组和组合命令

外壳允许您对命令执行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:

### Command1 ; Command2

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
```

## Command1 && Command2

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
```

## Command1 || Command2

当左侧命令失败时运行右侧命令（逻辑运算符 OR）。下行只会在于 /home/tux/foo 中创建目录失败时，才会在 /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参数。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
```

# 1.7 使用通用流程构造

为了控制脚本的流程，外壳可以使用 while、if、for 和 case 构造语句。

## 1.7.1 if 控制命令

if 命令用于检查表达式。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
```

测试表达式也可以缩写在方括号中：

```
if [ -e /tmp/foo.txt ] ; then
    echo "Found foo.txt"
fi
```

在 <https://bash.cyberciti.biz/guide/If..else..fi> 上可以找到更多有用表达式。

## 1.7.2 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，以下代码会列显关于当前工作目录中 PNG 文件的特定信息：

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

## 1.8 更多信息

关于 Bash 的重要信息在手册页 `man bash` 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

## 2 sudo 基础知识

运行某些命令需要 root 特权。但是，出于安全原因及避免出错考虑，我们不建议以 root 身份登录。更安全的做法是先以普通用户身份登录，再使用 **sudo** 以提升的特权来运行命令。

在 SUSE Linux Enterprise Desktop 上，**sudo** 配置为与 **su** 的工作方式类似。但是，**sudo** 提供了一种灵活机制，可让用户使用其他任何用户的特权运行命令。这样，便可为某些用户和组指派具有特定特权的角色。例如，可以允许组 **users** 的成员使用用户 **wilber** 的特权运行命令。您可以通过禁止任何命令选项，来进一步限制对命令的访问。虽然 **su** 始终需要 root 口令才能使用 PAM 进行身份验证，但是您可以将 **sudo** 配置为使用您自己的身份凭证进行身份验证。这意味着用户无需共享 root 口令，因而提高了安全性。

### 2.1 sudo 基本用法

以下章节将会介绍 **sudo** 的基本用法。

#### 2.1.1 运行单个命令

作为普通的用户，您可以在命令前加上 **sudo** 来以 root 身份运行任何命令。系统会提示您提供 root 口令。如果身份验证成功，便会以 root 身份运行命令：

```
> id -un ①
tux
> sudo id -un
root's password: ②
root
> id -un
tux ③
> sudo id -un
④
root
```

① **id -un** 命令会打印当前用户的登录名。

- ② 在输入过程中不会显示口令，无论是明文还是屏蔽字符都不显示。
- ③ 只有以 **sudo** 开头的命令才会以提升的特权运行。
- ④ 提升的特权会持续一段时间，因此您不必再次提供 **root** 口令。



## 提示：I/O 重定向

使用 **sudo** 时，I/O 重定向不起作用：

```
> sudo echo s > /proc/sysrq-trigger
bash: /proc/sysrq-trigger: Permission denied
> sudo cat < /proc/1/maps
bash: /proc/1/maps: Permission denied
```

在上面的示例中，只有 **echo** 和 **cat** 命令才以提升的特权运行。重定向将由用户的外壳以用户特权执行。要以提升的特权执行重定向，请根据[第 2.1.2 节 “启动外壳”](#) 中所述启动外壳，或使用 **dd** 实用程序：

```
echo s | sudo dd of=/proc/sysrq-trigger
sudo dd if=/proc/1/maps | cat
```

## 2.1.2 启动外壳

使用 **sudo** 来以提升的特权运行命令的做法并不总是可行。虽然可以使用 **sudo bash** 命令，但还是建议您使用以下其中一种内置机制来启动外壳：

**sudo -s (<command>)**

启动 **SHELL** 环境变量所指定的外壳或目标用户的默认外壳。如果指定了某个命令，该命令会传递到外壳（使用 **-c** 选项）。否则，外壳将以交互模式运行。

```
tux:~ > sudo -s
root's password:
root:/home/tux # exit
tux:~ >
```

## `sudo -i (<command>)`

与 `-s` 类似，但会作为登录外壳启动外壳。这表示系统会处理该外壳的启动文件 (`.profile` 等)，并会将当前的工作目录设置为目标用户的主目录。

```
tux:~ > sudo -i  
root's password:  
root:~ # exit  
tux:~ >
```



### 提示：环境变量

默认情况下，`sudo` 不会传播环境变量。可以使用 `env_reset` 选项更改此行为（请参见 [有用的标志和选项](#)）。

## 2.2 配置 `sudo`

`sudo` 提供各种可配置的选项。



### 注意：无法使用 `sudo`

如果您不小心将自己锁定在 `sudo` 之外，可以使用 `su -` 和 `root` 口令来启动 `root` 外壳。要修复该错误，请运行 `visudo`。

### 2.2.1 编辑配置文件

`sudo` 的主要策略配置文件为 `/etc/sudoers`。如果该文件的格式错误，您可能会无法进入系统，因此强烈建议您使用 `visudo` 来进行编辑。`visudo` 可以防止发生编辑冲突，并会在保存修改内容之前检查语法错误。

您可以通过设置 `EDITOR` 环境变量来使用 `vi` 以外的另一种编辑器，例如：

```
sudo EDITOR=/usr/bin/nano visudo
```

请记住，/etc/sudoers 文件是由系统软件包提供的，直接在该文件中进行修改可能会使更新中断。因此，建议将自定义配置存放在 /etc/sudoers.d/ 目录下的文件中。使用以下命令创建或编辑文件：

```
sudo visudo -f /etc/sudoers.d/NAME
```

以下命令使用另一种编辑器（在本例中为 nano）打开文件：

```
sudo EDITOR=/usr/bin/nano visudo -f /etc/sudoers.d/NAME
```

## 注意：/etc/sudoers.d 中忽略的文件

/etc/sudoers 中的 #includedir 指令会忽略以~（波浪线）字符结尾或包含.（点）字符的文件。

有关 visudo 命令的详细信息，请运行 man 8 visudo。

## 2.2.2 sudoers 基本配置语法

sudoers 配置文件包含两种类型的选项：字符串和标志。字符串可以包含任何值，而标志则只能在“ON”或“OFF”之间切换。sudoers 配置文件最重要的语法构造如下：

```
# Everything on a line after # is ignored ①
Defaults !insults # Disable the insults flag ②
Defaults env_keep += "DISPLAY HOME" # Add DISPLAY and HOME to env_keep
tux ALL = NOPASSWD: /usr/bin/frobnicate, PASSWD: /usr/bin/journalctl ③
```

- ① 存在两个例外：#include 和 #includedir 是普通命令。
- ② 去除！字符会将所需标志设置为 ON。
- ③ 请参见第 2.2.3 节“基本 sudoers 规则”。

### 有用的标志和选项

#### targetpw

此标志控制调用用户是否需要输入目标用户（例如 root）的口令 (ON) 还是需要输入调用用户的口令 (OFF)。

```
Defaults targetpw # Turn targetpw flag ON
```

### rootpw

如果设置, **sudo** 将提示输入 root 口令。默认值为 OFF。

```
Defaults !rootpw # Turn rootpw flag OFF
```

### env\_reset

如果设置, **sudo** 会构造一个具有

TERM、PATH、HOME、MAIL、SHELL、LOGNAME、USER、USERNAME 和 SUDO\_\* 的极简环境。此外, 会从调用环境导入 env\_keep 中列出的变量。默认值为 “ON”。

```
Defaults env_reset # Turn env_reset flag ON
```

### env\_keep

env\_reset 标志设为 “ON” 时要保留的环境变量列表。

```
# Set env_keep to contain EDITOR and PROMPT
Defaults env_keep = "EDITOR PROMPT"
Defaults env_keep += "JRE_HOME" # Add JRE_HOME
Defaults env_keep -= "JRE_HOME" # Remove JRE_HOME
```

### env\_delete

env\_reset 标志设为 “OFF” 时要去除的环境变量列表。

```
# Set env_delete to contain EDITOR and PROMPT
Defaults env_delete = "EDITOR PROMPT"
Defaults env_delete += "JRE_HOME" # Add JRE_HOME
Defaults env_delete -= "JRE_HOME" # Remove JRE_HOME
```

还可以使用 Defaults 令牌为用户、主机和命令集合创建别名。并且, 可以仅将选项应用到特定用户集。

有关 /etc/sudoers 配置文件的详细信息, 请参见 man 5 sudoers。

## 2.2.3 基本 sudoers 规则

每条规则都遵循以下模式 ([ ] 标记的是可选部分) :

```
#Who      Where      As whom      Tag      What
User_List Host_List = [(User_List)] [NOPASSWD:|PASSWD:] Cmnd_List
```

## SUDOERS 规则语法

### User\_List

一个或多个（用逗号分隔）标识符：用户名、%GROUPNAME 格式的组或 #UID 格式的用户 ID。可以使用 ! 前缀指定求反。

### Host\_List

一个或多个（用逗号分隔）标识符：（完全限定的）主机名或 IP 地址。可以使用 ! 前缀指定求反。Host\_List 的常用选项为 ALL。

### NOPASSWD: | PASSWD:

如果用户在 NOPASSWD: 后面运行的命令与 Cmd\_List 匹配，系统不会提示用户输入口令。

PASSWD 为默认选项。仅当 PASSWD 和 NOPASSWD 位于同一行时，才需要指定此选项：

```
tux ALL = PASSWD: /usr/bin/foo, NOPASSWD: /usr/bin/bar
```

### Cmnd\_List

一个或多个（用逗号分隔）说明符：可执行文件的路径，后跟允许使用的可选参数。

```
/usr/bin/foo      # Anything allowed
/usr/bin/foo bar # Only "/usr/bin/foo bar" allowed
/usr/bin/foo ""   # No arguments allowed
```

ALL 可以用作 User\_List、Host\_List 和 Cmnd\_List。

允许 tux 在无需输入口令的情况下以 root 身份运行所有命令的规则：

```
tux ALL = NOPASSWD: ALL
```

允许 tux 运行 systemctl restart apache2 的规则：

```
tux ALL = /usr/bin/systemctl restart apache2
```

允许 tux 在不带参数的情况下以 admin 身份运行 wall 的规则：

```
tux ALL = (admin) /usr/bin/wall ""
```



## 警告：不安全的规则

使用 `ALL ALL = ALL` 之类的规则时切勿不带 `Defaults targetpw`。否则，任何人都能够以 `root` 身份运行命令。



## 重要：Winbind 和 sudo

在 `sudoers` 文件中指定组名时，请确保使用 NetBIOS 域名而不是领域，例如：

```
%DOMAIN\\GROUP_NAME ALL = (ALL) ALL
```

请注意，使用 `winbindd` 时，其格式还取决于 `smb.conf` 文件中的 `winbind separator` 选项。默认为 `\`。例如，如果将其更改为 `+`，则 `sudoers` 文件中的帐户格式必须为 `DOMAIN+GROUP_NAME`。

## 2.3 sudo 使用案例

虽然默认配置适用于标准的使用方案，您仍可以根据特定需求自定义默认配置。

### 2.3.1 在无需提供 root 口令的情况下使用 sudo

根据设计，组 `wheel` 的成员能以 `root` 身份运行所有带有 `sudo` 的命令。下面的过程介绍如何将用户帐户添加到 `wheel` 组。

1. 校验 `wheel` 组是否存在：

```
> getent group wheel
```

如果之前的命令未返回任何结果，请安装会创建 `wheel` 组的 `system-group-wheel` 软件包：

```
> sudo zypper install system-group-wheel
```

2. 将您的用户帐户添加到 `wheel` 组。

如果您的用户帐户还不是 `wheel` 组的成员, 请使用 `sudo usermod -a -G wheel USERNAME` 命令添加。注销然后重新登录即可启用更改。运行 `groups USERNAME` 命令以校验更改是否成功。

### 3. 使用用户帐户的普通口令进行身份验证。

使用 `visudo` 命令创建文件 `/etc/sudoers.d/userpw` (请参见[第 2.2.1 节 “编辑配置文件”](#)) 并添加以下内容:

```
Defaults !targetpw
```

### 4. 选择新默认规则。

根据是否希望用户重新输入口令, 在 `/etc/sudoers` 中取消注释相应的行, 并将默认规则注释掉。

```
## Uncomment to allow members of group wheel to execute any command
# %wheel  ALL=(ALL)  ALL

## Same thing without a password
# %wheel  ALL=(ALL)  NOPASSWD:  ALL
```

### 5. 提高默认规则的限制性。

将 `/etc/sudoers` 中允许一切操作的规则注释掉或去除:

```
ALL      ALL=(ALL)  ALL  # WARNING! Only use this together with 'Defaults
                           targetpw'!
```



### 警告: sudoers 中的危险规则

请勿跳过此步骤。否则任何用户都能以 `root` 身份执行任何命令!

### 6. 测试配置。

以 `wheel` 的成员和非成员身份运行 `sudo`。

```
tux:~ > groups
users wheel
tux:~ > sudo id -un
tux's password:
```

```
root
wilber:~ > groups
users
wilber:~ > sudo id -un
wilber is not in the sudoers file. This incident will be reported.
```

### 2.3.2 对 X.Org 应用程序使用 **sudo**

使用 **sudo** 启动图形应用程序通常会导致以下错误：

```
> sudo xterm
xterm: Xt error: Can't open display: %s
xterm: DISPLAY is not set
```

一种简单的解决方法是使用 **xhost** 来暂时允许 root 用户访问本地用户的 X 会话。可使用以下命令完成此操作：

```
xhost si:localuser:root
```

以下命令可去除授予的访问权限：

```
xhost -si:localuser:root
```



#### 警告：潜在安全问题

以 root 特权运行图形应用程序存在安全隐患。建议仅在例外的情况下，才启用对图形应用程序的 root 访问权限。另外，建议在关闭图形应用程序后，立即撤消授予的 root 访问权限。

## 2.4 更多信息

**sudo --help** 命令提供可用命令行选项的简要概述，而 **man sudoers** 命令则会提供有关 **sudoers** 及其配置的详细信息。

## 3 使用 YaST

YaST 是一个 SUSE Linux Enterprise Desktop 工具，它提供图形界面用于完成所有基本的安装和系统配置任务。无论您是要更新软件包、配置打印机、修改防火墙设置、设置 FTP 服务器还是为硬盘分区，都可以使用 YaST 来完成。YaST 使用 Ruby 编写，提供了一个可扩展的体系结构，让您能够通过模块添加新的功能。

项目官方网站 (<https://yast.opensuse.org/>) 上提供了有关 YaST 的更多信息。

### 3.1 YaST 界面概述

YaST 有两个图形界面：一个用于 KDE 和 GNOME 等图形桌面环境，另一个基于 ncurses 的伪图形界面用于不带 X 服务器的系统（请参见第 4 章 “文本模式下的 YaST”）。

在图形版 YaST 中，YaST 中的所有模块按类别分组，您可以使用导航侧栏快速访问所需类别中的模块。使用顶部的搜索字段可按名称查找模块。要查找特定的模块，请在搜索字段中输入其名称，然后，便会显示与所输入字符串匹配的模块。

### 3.2 有用的组合键

图形版 YaST 支持键盘快捷键

**Print Screen**

截图并保存。但在某些桌面环境中，可能无法正常工作。

**Shift + F4**

启用和禁用为视觉障碍用户专门优化的调色板。

**Shift + F7**

启用/禁用记录调试消息。

**Shift + F8**

打开一个文件对话框，以将日志文件保存到用户定义的位置。

**Ctrl** – **Shift** – **Alt** – **D**

发送一个调试事件。YaST 模块可执行特殊的调试操作来对此作出反应。结果取决于具体的 YaST 模块。

**Ctrl** – **Shift** – **Alt** – **M**

启动和停止宏记录器。

**Ctrl** – **Shift** – **Alt** – **P**

重新播放宏。

**Ctrl** – **Shift** – **Alt** – **S**

显示样式表编辑器。

**Ctrl** – **Shift** – **Alt** – **T**

将控件树转储到日志文件。

**Ctrl** – **Shift** – **Alt** – **X**

打开一个终端窗口 (xterm)。当通过 VNC 安装时很有用。

**Ctrl** – **Shift** – **Alt** – **Y**

显示控件树浏览器。

## 4 文本模式下的 YaST

基于 ncurses 的伪图形 YaST 界面主要用于帮助系统管理员管理不带 X 服务器的系统。与传统的 GUI 相比，该界面具备多项优势。您可以使用键盘在 ncurses 界面中导航，并且几乎所有界面元素都有对应的键盘快捷键。ncurses 界面的资源占用较小，即使是在最低配置的硬件上也能快速运行。您可以通过 SSH 连接运行基于 ncurses 的 YaST 版本，以便管理远程系统。请注意，用于运行 YaST 的终端仿真器支持的最小大小为 80x25 个字符。

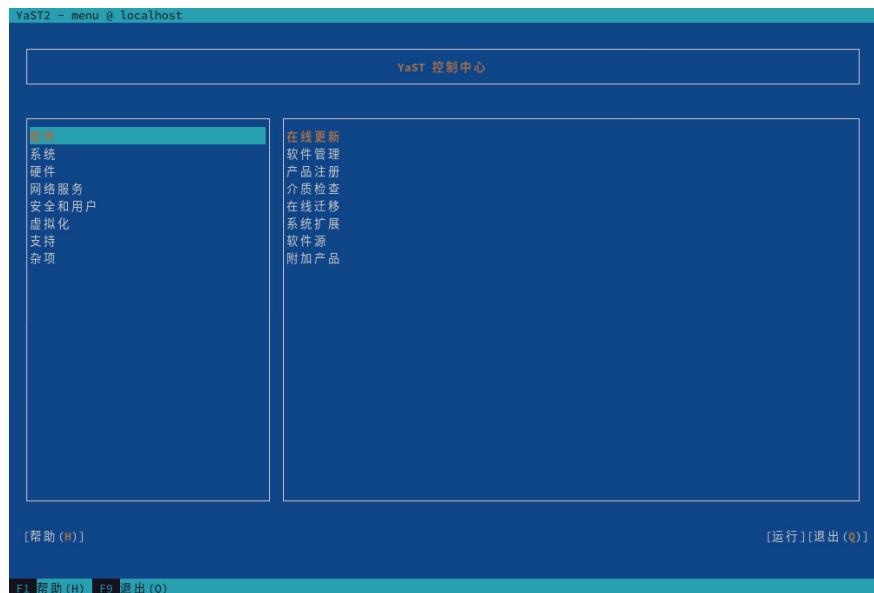


图 4.1：文本模式下的 YAST 主窗口

要启动基于 ncurses 的 YaST 版本，请打开终端并运行 `sudo yast2` 命令。使用 `→` 键或箭头键在菜单项、字段和按钮等界面元素之间导航。您可以使用相应功能键或键盘快捷键访问 YaST 中的所有菜单项和按钮。例如，您可以按 `F9` 键来取消当前操作，使用 `F10` 键可接受更改。在基于 ncurses 的 YaST 界面中，每个菜单项和按钮的标签中都有一个高亮显示的字母。此字母是指派给界面元素的键盘快捷键的一部分。例如，退出按钮中高亮显示了字母 `Q`。这表示您可以按 `Alt + Alt+Q` 来激活该按钮。



### 提示：刷新 YaST 对话框

如果 YaST 对话框损坏或变形（例如在调整窗口大小时），请按 `Ctrl + L` 来刷新并恢复其内容。

## 4.1 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 4.3 节 “组合键的限制”。

### 在按钮和选择列表之间移动

使用 **→|** 键可在按钮和包含选择列表的框架之间移动。要以相反方向导航，请使用 **Alt** **→|** 或 **Shift** **→|** 组合键。

### 在选择列表中导航

使用箭头键 (**↑** 和 **↓**) 可在包含选择列表的活动框架中的各个元素之间移动。如果单个项目超出了框架宽度，可使用 **Shift** **→** 或 **Shift** **←** 键水平滚动。如果使用箭头键会导致选择移到另一框架，请改用 **Ctrl** **→ E** 或 **Ctrl** **→ A** 键。

### 使用按钮、单选按钮和复选框

要选择带空方括号（复选框）或空圆括号（单选按钮）的项目，请按 **Space** 或 **Enter** 键。或者，可以使用 **Alt** **highlighted\_letter** 直接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 **→|** 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。

### 功能键

使用功能键（从 **F1** 到 **F12**）可快速访问特定按钮。YaST 屏幕底部的行中显示了可用的功能键组合 (FX)。哪个功能键实际对应哪个按钮取决于活动 YaST 模块，因为不同的模块提供的按钮不同（细节、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。

### 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。使用方向键 (**↑** 和 **↓**) 可在树中导航。使用 **Space** 可打开或关闭树中的项。在 ncurses 模式下，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。

### 在软件安装模块中选择软件

使用左侧的过滤器可以列出与指定的字符串匹配的软件包。已安装的软件包标有字母 i。要更改软件包的状态，请按 **Space** 或 **Enter**。或者，也可以使用操作菜单选择所需的状态更改（安装、删除、更新、禁止或锁定）。

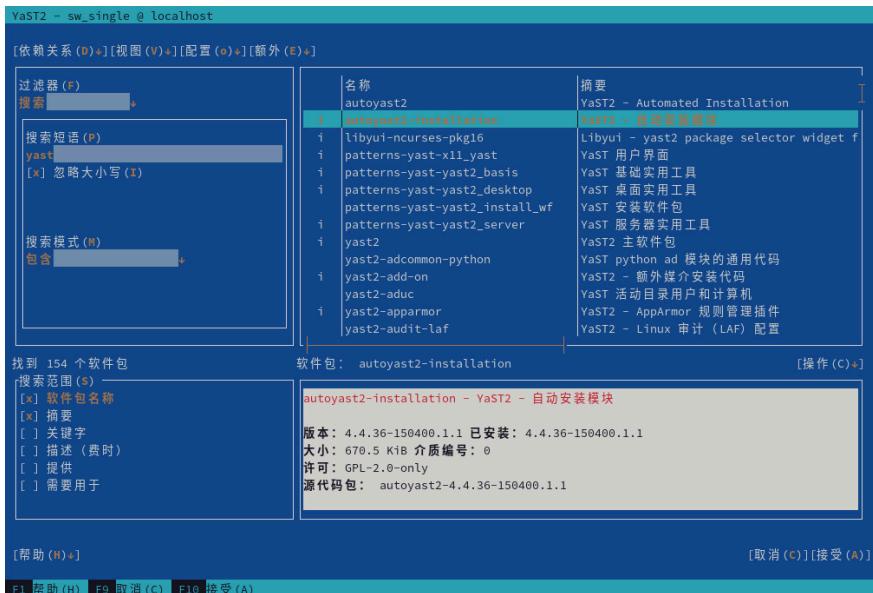


图 4.2：软件安装模块

## 4.2 高级组合键

基于 ncurses 的 YaST 版本提供有数个高级组合键。

**Shift – F1**

列出高级热键。

**Shift – F4**

更改颜色方案。

**Ctrl – Q**

退出应用程序。

**Ctrl – L**

刷新屏幕。

**Ctrl – D | F1**

列出高级热键。

**Ctrl – D | Shift – D**

以屏幕截图的形式将对话框转储到日志文件。

**Ctrl + D** | **Shift + Y**

打开 YDialogSpy 以查看控件层次结构。

## 4.3 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 **Alt** 而不是 **Esc**

可以代替 **Alt** 而使用 **Esc** **Alt** 快捷键。例如，**Esc + H** 可代替 **Alt + H**。（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** 组合键被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（**F1** … **F12**）也用于执行多种功能。某些功能键可能已被终端占用，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。

## 4.4 YaST 命令行选项

除了文本模式界面以外，YaST 还提供了一个命令行界面。要获取 YaST 命令行选项列表，请使用以下命令：

```
> sudo yast -h
```

### 4.4.1 从命令行安装软件包

如果您知道软件包名称，且该软件包是由某个活动安装储存库提供的，您便可以使用命令行选项 -i 安装该软件包：

```
> sudo yast -i package_name
```

或

```
> sudo yast --install -i package_name
```

`package_name` 可以是通过依赖项检查安装的单个软件包简称（例如 `gvim`），也可以是并非通过依赖项检查安装的 RPM 软件包的完整路径。

虽然 YaST 提供了通过命令行管理软件的基本功能，但对于更高级的软件包管理任务，请考虑使用 Zypper。在[第 9.1 节 “使用 Zypper”](#) 中查找有关使用 Zypper 的详细信息。

## 4.4.2 使用单个模块

为了节省时间，您可以使用以下命令启动单个 YaST 模块：

```
> sudo yast module_name
```

使用 `yast -l` 或 `yast --list` 可查看系统上所有可用模块的列表。

## 4.4.3 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。但并非所有模块都具有命令行支持。要显示模块的可用选项，请使用以下命令：

```
> sudo yast module_name help
```

如果某个模块未提供命令行支持，则会以文本模式启动该模块，并显示以下消息：

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

下列各节介绍具有命令行支持的所有 YaST 模块，并简要说明了其所有命令和可用选项。

### 4.4.3.1 常用 YaST 模块命令

所有 YaST 模块都支持以下命令：

#### **help**

列出所有模块的支持命令及其说明：

```
> sudo yast lan help
```

## longhelp

与 help 作用相同，但添加了所有命令选项及其说明的详细列表：

```
> sudo yast lan longhelp
```

## xmlhelp

与 longhelp 作用相同，但输出采用 XML 文本结构并重定向至某个文件：

```
> sudo yast lan xmlhelp xmlfile=/tmp/yast_lan.xml
```

## interactive

进入 交互 模式。这样，您就可以直接运行该模块的命令，而无需在命令前加上 sudo yast 前缀。使用 exit 可以退出交互模式。

### 4.4.3.2 yast add-on

从指定的路径添加新附加产品：

```
> sudo yast add-on http://server.name/directory/Lang-AddOn-CD1/
```

您可以使用以下协议来指定来源路径：http:// ftp:// nfs:// disk:// cd:// 或 dvd://。

### 4.4.3.3 yast audit-laf

显示和配置 Linux Audit Framework。有关更多细节，请参见《安全和强化指南》。yast audit-laf 接受下列命令：

#### set

设置选项：

```
> sudo yast audit-laf set log_file=/tmp/audit.log
```

要显示完整的选项列表，请运行 **yast audit-laf set help**。

## show

显示选项的设置：

```
> sudo yast audit-laf show diskspace
space_left: 75
space_left_action: SYSLOG
admin_space_left: 50
admin_space_left_action: SUSPEND
action_mail_acct: root
disk_full_action: SUSPEND
disk_error_action: SUSPEND
```

要显示完整的选项列表，请运行 [\*\*yast audit-laf show help\*\*](#)。

### 4.4.3.4 [\*\*yast dhcp-server\*\*](#)

管理 DHCP 服务器以及配置其设置。[\*\*yast dhcp-server\*\*](#) 接受下列命令：

#### **disable**

禁用 DHCP 服务器服务。

#### **enable**

启用 DHCP 服务器服务。

#### **host**

配置单个主机的设置。

#### **interface**

指定要侦听的网络接口：

```
> sudo yast dhcp-server interface current
Selected Interfaces: eth0
Other Interfaces: bond0, pbu, eth1
```

要显示完整的选项列表，请运行 [\*\*yast dhcp-server interface help\*\*](#)。

#### **options**

管理全局 DHCP 选项。要显示完整的选项列表，请运行 [\*\*yast dhcp-server options help\*\*](#)。

## status

打印 DHCP 服务的状态。

## subnet

管理 DHCP 子网选项。要显示完整的选项列表，请运行 [\*\*yast dhcp-server subnet help\*\*](#)。

### 4.4.3.5 [\*\*yast dns-server\*\*](#)

管理 DNS 服务器配置。[\*\*yast dns-server\*\*](#) 接受下列命令：

#### acls

显示访问控制列表设置：

```
> sudo yast dns-server acls show
ACLs:
-----
Name      Type      Value
-----
any      Predefined
localips  Predefined
localnets Predefined
none      Predefined
```

#### dnsrecord

配置区域资源记录：

```
> sudo yast dnsrecord add zone=example.org query=office.example.org type=NS
value=ns3
```

要显示完整的选项列表，请运行 [\*\*yast dns-server dnsrecord help\*\*](#)。

#### forwarders

配置 DNS 转发器：

```
> sudo yast dns-server forwarders add ip=10.0.0.100
> sudo yast dns-server forwarders show
[...]
```

#### Forwarder IP

-----  
10.0.0.100

要显示完整的选项列表，请运行 **yast dns-server forwarders help**。

#### host

立即处理“A”及其相关的“PTR”记录：

```
> sudo yast dns-server host show zone=example.org
```

要显示完整的选项列表，请运行 **yast dns-server host help**。

#### logging

配置日志记录设置：

```
> sudo yast dns-server logging set updates=no transfers=yes
```

要显示完整的选项列表，请运行 **yast dns-server logging help**。

#### mailserver

配置区域邮件服务器：

```
> sudo yast dns-server mailserver add zone=example.org mx=mx1 priority=100
```

要显示完整的选项列表，请运行 **yast dns-server mailserver help**。

#### nameserver

配置区域名称服务器：

```
> sudo yast dns-server nameserver add zone=example.com ns=ns1
```

要显示完整的选项列表，请运行 **yast dns-server nameserver help**。

#### soa

配置授权起始 (SOA) 记录：

```
> sudo yast dns-server soa set zone=example.org serial=2006081623  
ttl=2D3H20S
```

要显示完整的选项列表，请运行 **yast dns-server soa help**。

## startup

管理 DNS 服务器服务：

```
> sudo yast dns-server startup atboot
```

要显示完整的选项列表，请运行 [\*\*yast dns-server startup help\*\*](#)。

## transport

配置区域传输规则。要显示完整的选项列表，请运行 [\*\*yast dns-server transport help\*\*](#)。

## zones

管理 DNS 区域：

```
> sudo yast dns-server zones add name=example.org zonetype=master
```

要显示完整的选项列表，请运行 [\*\*yast dns-server zones help\*\*](#)。

### 4.4.3.6 yast disk

打印所有磁盘或分区的相关信息。唯一支持的命令是 [\*\*list\*\*](#)，后跟以下任一选项：

#### disks

列出系统中配置的所有磁盘：

```
> sudo yast disk list disks
Device | Size | FS Type | Mount Point | Label | Model
-----+-----+-----+-----+-----+-----+
/dev/sda | 119.24 GiB | | | | SSD 840
/dev/sdb | 60.84 GiB | | | | WD1003FBYX-0
```

#### partitions

列出系统中的所有分区：

```
> sudo yast disk list partitions
Device | Size | FS Type | Mount Point | Label | Model
-----+-----+-----+-----+-----+-----+
/dev/sda1 | 1.00 GiB | Ext2 | /boot | |
```

|                |            |      |            |  |  |
|----------------|------------|------|------------|--|--|
| /dev/sdb1      | 1.00 GiB   | Swap | swap       |  |  |
| /dev/sdc1      | 698.64 GiB | XFS  | /mnt/extra |  |  |
| /dev/vg00/home | 580.50 GiB | Ext3 | /home      |  |  |
| /dev/vg00/root | 100.00 GiB | Ext3 | /          |  |  |
| [...]          |            |      |            |  |  |

#### 4.4.3.7 **yast ftp-server**

配置 FTP 服务器设置。**yast ftp-server** 接受下列选项：

##### **SSL、TLS**

通过 SSL 和 TLS 控制安全连接。SSL 选项仅对 **vsftpd** 有效。

```
> sudo yast ftp-server SSL enable  
> sudo yast ftp-server TLS disable
```

##### **access**

配置访问权限：

```
> sudo yast ftp-server access authen_only
```

要显示完整的选项列表，请运行 **yast ftp-server access help**。

##### **anon\_access**

配置匿名用户的访问权限：

```
> sudo yast ftp-server anon_access can_upload
```

要显示完整的选项列表，请运行 **yast ftp-server anon\_access help**。

##### **anon\_dir**

指定匿名用户的目录。服务器上必须已存在该目录：

```
> sudo yast ftp-server anon_dir set_anon_dir=/srv/ftp
```

要显示完整的选项列表，请运行 **yast ftp-server anon\_dir help**。

##### **chroot**

控制 **change root** 环境 (chroot)：

```
> sudo yast ftp-server chroot enable  
> sudo yast ftp-server chroot disable
```

## idle-time

设置最大闲置时间（分钟），经过该时间后 FTP 服务器会终止当前的连接：

```
> sudo yast ftp-server idle-time set_idle_time=15
```

## logging

确定是否将日志消息保存到日志文件中：

```
> sudo yast ftp-server logging enable  
> sudo yast ftp-server logging disable
```

## max\_clients

指定并行连接的客户端的最大数目：

```
> sudo yast ftp-server max_clients set_max_clients=1500
```

## max\_clients\_ip

指定通过 IP 并行连接的客户端的最大数目：

```
> sudo yast ftp-server max_clients_ip set_max_clients=20
```

## max\_rate\_anon

指定匿名客户端允许使用的最大数据传输速度 (KB/s)：

```
> sudo yast ftp-server max_rate_anon set_max_rate=10000
```

## max\_rate\_authen

指定通过本地身份验证的用户允许使用的最大数据传输速度 (KB/s)：

```
> sudo yast ftp-server max_rate_authen set_max_rate=10000
```

## port\_range

指定被动连接答复的端口范围：

```
> sudo yast ftp-server port_range set_min_port=20000 set_max_port=30000
```

要显示完整的选项列表, 请运行 **yast ftp-server port\_range help**。

## show

显示 FTP 服务器设置。

## startup

控制 FTP 启动方法:

```
> sudo yast ftp-server startup atboot
```

要显示完整的选项列表, 请运行 **yast ftp-server startup help**。

## umask

指定 authenticated:anonymous 用户的文件权限掩码:

```
> sudo yast ftp-server umask set_umask=177:077
```

## welcome\_message

指定当用户连接到 FTP 服务器时显示的文本:

```
> sudo yast ftp-server welcome_message set_message="hello everybody"
```

## 4.4.3.8 yast http-server

配置 HTTP 服务器 (Apache2)。 **yast http-server** 接受下列命令:

### configure

配置 HTTP 服务器主机设置:

```
> sudo yast http-server configure host=main servername=www.example.com \
  serveradmin=admin@example.com
```

要显示完整的选项列表, 请运行 **yast http-server configure help**。

### hosts

配置虚拟主机:

```
> sudo yast http-server hosts create servername=www.example.com \
```

```
serveradmin@admin@example.com documentroot=/var/www
```

要显示完整的选项列表, 请运行 **yast http-server hosts help**。

## listen

指定 HTTP 服务器应该侦听的端口和网络地址:

```
> sudo yast http-server listen add=81
> sudo yast http-server listen list
Listen Statements:
=====
:80
:81
> sudo yast http-server delete=80
```

要显示完整的选项列表, 请运行 **yast http-server listen help**。

## mode

启用或禁用向导模式:

```
> sudo yast http-server mode wizard=on
```

## modules

控制 Apache2 服务器模块:

```
> sudo yast http-server modules enable=php5,rewrite
> sudo yast http-server modules disable=ssl
> sudo http-server modules list
[...]
Enabled rewrite
Disabled ssl
Enabled php5
[...]
```

### 4.4.3.9 **yast kdump**

配置 **kdump** 设置。有关 **kdump** 的详细信息, 请参见《系统分析和微调指南》, 第 18 章 “Kexec 和 Kdump”, 第 18.7 节 “Kdump 基本配置”。**yast kdump** 接受下列命令:

## copykernel

将内核复制到转储目录。

## customkernel

指定自定义内核名称的 `kernel_string` 部分。命名模式为 `/boot/vmlinu[zx]-kernel_string[.gz]`。

```
> sudo yast kdump customkernel kernel=kdump
```

要显示完整的选项列表，请运行 `yast kdump customkernel help`。

## dumpformat

指定转储内核映像的（压缩）格式。可用格式有“none”、“ELF”、“compressed”或“lzo”：

```
> sudo yast kdump dumpformat dump_format=ELF
```

## dumplevel

指定转储级别，范围从 0 到 31：

```
> sudo yast kdump dumplevel dump_level=24
```

## dumptarget

指定用于保存转储映像的目标：

```
> sudo kdump dumptarget target=ssh server=name_server port=22 \
  dir=/var/log/dump user=user_name
```

要显示完整的选项列表，请运行 `yast kdump dumptarget help`。

## immediatereboot

控制系统是否应在将核心保存到 Kdump 内核中后立即重引导：

```
> sudo yast kdump immediatereboot enable
> sudo yast kdump immediatereboot disable
```

## keepolddumps

指定保留旧转储映像的数量。可指定保留零个到全部保留：

```
> sudo yast kdump keepolddumps no=5
```

## kernelcommandline

指定需要传递到 kdump 内核的命令行：

```
> sudo yast kdump kernelcommandline command="ro root=LABEL=/"
```

## kernelcommandlineappend

指定需要追加到默认命令行字符串的命令行：

```
> sudo yast kdump kernelcommandlineappend command="ro root=LABEL=/"
```

## notificationcc

指定用于抄送通知邮件的电子邮件地址：

```
> sudo yast kdump notificationcc email="user1@example.com  
user2@example.com"
```

## notificationonto

指定用于发送通知邮件的电子邮件地址：

```
> sudo yast kdump notificationonto email="user1@example.com  
user2@example.com"
```

## show

显示 kdump 设置：

```
> sudo yast kdump show  
Kdump is disabled  
Dump Level: 31  
Dump Format: compressed  
Dump Target Settings  
target: file  
file directory: /var/crash  
Kdump immediate reboots: Enabled  
Numbers of old dumps: 5
```

## smtppass

指定含有用于发送通知邮件的纯文本 SMTP 口令的文件：

```
> sudo yast kdump smtppass pass=/path/to/file
```

## smtpserver

指定用于发送通知邮件的 SMTP 服务器主机名：

```
> sudo yast kdump smtpserver server=smtp.server.com
```

## smtpuser

指定用于发送通知邮件的 SMTP 用户名：

```
> sudo yast kdump smtpuser user=smtp_user
```

## startup

启用或禁用启动选项：

```
> sudo yast kdump startup enable alloc_mem=128,256  
> sudo yast kdump startup disable
```

## 4.4.3.10 yast keyboard

配置虚拟控制台的系统键盘。它不会影响图形桌面环境（例如 GNOME 或 KDE）中的键盘设置。**yast keyboard** 接受下列命令：

### list

列出所有可用的键盘布局。

### set

激活新的键盘布局设置：

```
> sudo yast keyboard set layout=czech
```

### summary

显示当前的键盘配置。

## 4.4.3.11 yast lan

配置网卡。**yast lan** 接受下列命令：

### add

配置新网卡：

```
> sudo yast lan add name=vlan50 ethdevice=eth0 bootproto=dhcp
```

要显示完整的选项列表, 请运行 **yast lan add help**。

## delete

删除某个现有网卡:

```
> sudo yast lan delete id=0
```

## edit

更改某个现有网卡的配置:

```
> sudo yast lan edit id=0 bootproto=dhcp
```

## list

显示网卡配置摘要:

```
> sudo yast lan list
id name,          bootproto
0 Ethernet Card 0, NONE
1 Network Bridge, DHCP
```

## 4.4.3.12 yast language

配置系统语言。 **yast language** 接受下列命令:

### list

列出所有可用的语言。

### set

指定系统的主要语言和次要语言:

```
> sudo yast language set lang=cs_CZ languages=en_US,es_ES no_packages
```

## 4.4.3.13 yast mail

显示邮件系统的配置:

```
> sudo yast mail summary
```

#### 4.4.3.14 yast nfs

控制 NFS 客户端。 **yast nfs** 接受下列命令：

##### add

添加新 NFS 载具：

```
> sudo yast nfs add spec=remote_host:/path/to/nfs/share file=/local/mount/point
```

要显示完整的选项列表，请运行 **yast nfs add help**。

##### delete

删除现有的 NFS 载具：

```
> sudo yast nfs delete spec=remote_host:/path/to/nfs/share file=/local/mount/point
```

要显示完整的选项列表，请运行 **yast nfs delete help**。

##### edit

更改现有的 NFS 载具：

```
> sudo yast nfs edit spec=remote_host:/path/to/nfs/share \
  file=/local/mount/point type=nfs4
```

要显示完整的选项列表，请运行 **yast nfs edit help**。

##### list

列出现有的 NFS 载具：

| Server          | Remote File System  | Mount Point | Options |
|-----------------|---------------------|-------------|---------|
| nfs.example.com | /mnt                | /nfs/mnt    | nfs     |
| nfs.example.com | /home/tux/nfs_share | /nfs/tux    | nfs     |

#### 4.4.3.15 `yast nfs-server`

配置 NFS 服务器。`yast nfs-server` 接受下列命令：

##### **add**

将目录添加到导出中：

```
> sudo yast nfs-server add mountpoint=/nfs/export hosts=*.allowed_hosts.com
```

要显示完整的选项列表，请运行 `yast nfs-server add help`。

##### **delete**

从 NFS 导出中删除目录：

```
> sudo yast nfs-server delete mountpoint=/nfs/export
```

##### **set**

指定 NFS 服务器的其他参数：

```
> sudo yast nfs-server set enablev4=yes security=yes
```

要显示完整的选项列表，请运行 `yast nfs-server set help`。

##### **start**

启动 NFS 服务器服务：

```
> sudo yast nfs-server start
```

##### **stop**

停止 NFS 服务器服务：

```
> sudo yast nfs-server stop
```

##### **summary**

显示 NFS 服务器配置摘要：

```
> sudo yast nfs-server summary
NFS server is enabled
NFS Exports
```

```
* /mnt
* /home

NFSv4 support is enabled.
The NFSv4 domain for idmapping is localdomain.
NFS Security using GSS is enabled.
```

#### 4.4.3.16 `yast nis`

配置 NIS 客户端。`yast nis` 接受下列命令：

##### **configure**

更改 NIS 客户端的全局设置：

```
> sudo yast nis configure server=nis.example.com broadcast=yes
```

要显示完整的选项列表，请运行 `yast nis configure help`。

##### **disable**

禁用 NIS 客户端：

```
> sudo yast nis disable
```

##### **enable**

允许您的计算机作为 NIS 客户端：

```
> sudo yast nis enable server=nis.example.com broadcast=yes automounter=yes
```

要显示完整的选项列表，请运行 `yast nis enable help`。

##### **find**

显示给定域的可用 NIS 服务器：

```
> sudo yast nis find domain=nisdomain.com
```

##### **summary**

显示 NIS 客户端的配置摘要。

## 4.4.3.17 yast nis-server

配置 NIS 服务器。yast nis-server 接受下列命令：

### master

配置 NIS 主服务器：

```
> sudo yast nis-server master domain=nisdomain.com yppasswd=yes
```

要显示完整的选项列表，请运行 yast nis-server master help。

### slave

配置 NIS 工作器服务器：

```
> sudo yast nis-server slave domain=nisdomain.com master_ip=10.100.51.65
```

要显示完整的选项列表，请运行 yast nis-server slave help。

### stop

停止 NIS 服务器：

```
> sudo yast nis-server stop
```

### summary

显示 NIS 服务器配置摘要：

```
> sudo yast nis-server summary
```

## 4.4.3.18 yast proxy

配置代理设置。yast proxy 接受下列命令：

### authentication

指定代理的身份验证选项：

```
> sudo yast proxy authentication username=tux password=secret
```

要显示完整的选项列表，请运行 yast proxy authentication help。

## enable、disable

启用或禁用代理设置。

### set

更改当前代理设置：

```
> sudo yast proxy set https=proxy.example.com
```

要显示完整的选项列表，请运行 **yast proxy set help**。

### summary

显示代理设置。

## 4.4.3.19 yast rdp

控制远程桌面设置。**yast rdp** 接受下列命令：

### allow

允许远程访问服务器的桌面：

```
> sudo yast rdp allow set=yes
```

### list

显示远程桌面配置摘要。

## 4.4.3.20 yast samba-client

配置 Samba 客户端设置。**yast samba-client** 接受下列命令：

### configure

更改 Samba 的全局设置：

```
> sudo yast samba-client configure workgroup=FAMILY
```

### isdomainmember

检查计算机是否为域成员：

```
> sudo yast samba-client isdomainmember domain=SMB_DOMAIN
```

## joindomain

让计算机成为域成员：

```
> sudo yast samba-client joindomain domain=SMB_DOMAIN user=username  
password=pwd
```

## winbind

启用或禁用 Winbind 服务 (winbindd 守护程序)：

```
> sudo yast samba-client winbind enable  
> sudo yast samba-client winbind disable
```

### 4.4.3.21 yast samba-server

配置 Samba 服务器设置。yast samba-server 接受下列命令：

#### backend

指定存储用户信息的后端：

```
> sudo yast samba-server backend smbpasswd
```

要显示完整的选项列表，请运行 yast samba-server backend help。

#### configure

配置 Samba 服务器的全局设置：

```
> sudo yast samba-server configure workgroup=FAMILY description='Home  
server'
```

要显示完整的选项列表，请运行 yast samba-server configure help。

#### list

显示可用共享列表：

```
> sudo yast samba-server list
```

| Status   | Type    | Name     |
|----------|---------|----------|
| <hr/>    |         |          |
| Disabled | Disk    | profiles |
| Enabled  | Disk    | print\$  |
| Enabled  | Disk    | homes    |
| Disabled | Disk    | groups   |
| Enabled  | Disk    | movies   |
| Enabled  | Printer | printers |

## role

指定 Samba 服务器的角色：

```
> sudo yast samba-server role standalone
```

要显示完整的选项列表，请运行 [yast samba-server role help](#)。

## service

启用或禁用 Samba 服务 (smb 和 nmb)：

```
> sudo yast samba-server service enable
> sudo yast samba-server service disable
```

## share

操作单个 Samba 共享：

```
> sudo yast samba-server share name=movies browseable=yes guest_ok=yes
```

要显示完整的选项列表，请运行 [yast samba-server share help](#)。

## 4.4.3.22 yast security

控制主机的安全级别。[yast security](#) 接受下列命令：

### level

指定主机的安全级别：

```
> sudo yast security level server
```

要显示完整的选项列表, 请运行 **yast security level help**。

## set

设置特定选项的值:

```
> sudo yast security set passwd=sha512 crack=yes
```

要显示完整的选项列表, 请运行 **yast security set help**。

## summary

显示当前安全配置的摘要:

```
sudoyast security summary
```

### 4.4.3.23 yast sound

配置声卡设置。**yast sound** 接受下列命令:

#### add

配置新声卡。不使用任何参数时, 该命令会添加检测到的第一个声卡。

```
> sudo yast sound add card=0 volume=75
```

要显示完整的选项列表, 请运行 **yast sound add help**。

#### channels

列出声卡的可用音量声道:

```
> sudo yast sound channels card=0
Master 75
PCM 100
```

#### modules

列出所有可用的声音内核模块:

```
> sudo yast sound modules
snd-atiixp ATI IXP AC97 controller (snd-atiixp)
snd-atiixp-modem ATI IXP MC97 controller (snd-atiixp-modem)
```

```
snd-virtuoso Asus Virtuoso driver (snd-virtuoso)
[...]
```

## playtest

在声卡上播放测试声音：

```
> sudo yast sound playtest card=0
```

## remove

去除配置的声卡：

```
> sudo yast sound remove card=0
> sudo yast sound remove all
```

## set

为声卡指定新值：

```
> sudo yast sound set card=0 volume=80
```

## show

显示有关声卡的详细信息：

```
> sudo yast sound show card=0
Parameters of card 'ThinkPad X240' (using module snd-hda-intel):

align_buffer_size
  Force buffer and period sizes to be multiple of 128 bytes.
bdl_pos_adj
  BDL position adjustment offset.
beep_mode
  Select HDA Beep registration mode (0=off, 1=on) (default=1).
  Default Value: 0
enable_msi
  Enable Message Signaled Interrupt (MSI)
[...]
```

## summary

打印系统上所有声卡的配置摘要：

```
> sudo yast sound summary
```

## volume

指定声卡的音量级别：

```
sudoyast sound volume card=0 play
```

### 4.4.3.24 yast sysconfig

控制 /etc/sysconfig 下的文件中的变量。**yast sysconfig** 接受下列命令：

#### clear

为变量设置空值：

```
> sudo yast sysconfig clear=POSTFIX_LISTEN
```



#### 提示：多个文件中的变量

如果变量可在多个文件中使用，请使用 VARIABLE\_NAME\$FILE\_NAME 语法：

```
> sudo yast sysconfig clear=CONFIG_TYPE$/etc/sysconfig/mail
```

#### details

显示有关变量的详细信息：

```
> sudo yast sysconfig details variable=POSTFIX_LISTEN
Description:
Value:
File: /etc/sysconfig/postfix
Possible Values: Any value
Default Value:
Configuration Script: postfix
Description:
Comma separated list of IP's
NOTE: If not set, LISTEN on all interfaces
```

## list

显示已修改变量的摘要。使用 all 可列出所有变量及相应的值：

```
> sudo yast sysconfig list all
AOU_AUTO_AGREE_WITH_LICENSES="false"
AOU_ENABLE_CRONJOB="true"
AOU_INCLUDE_RECOMMENDS="false"
[...]
```

## set

为变量设置值：

```
> sudo yast sysconfig set DISPLAYMANAGER=gdm
```



## 提示：多个文件中的变量

如果变量可在多个文件中使用，请使用 VARIABLE\_NAME\$FILE\_NAME 语法：

```
> sudo yast sysconfig set CONFIG_TYPE$/etc/sysconfig/mail=advanced
```

### 4.4.3.25 yast tftp-server

配置 TFTP 服务器。yast tftp-server 接受下列命令：

#### directory

指定 TFTP 服务器的目录：

```
> sudo yast tftp-server directory path=/srv/tftp
> sudo yast tftp-server directory list
Directory Path: /srv/tftp
```

#### status

控制 TFTP 服务器服务的状态：

```
> sudo yast tftp-server status disable
> sudo yast tftp-server status show
```

```
Service Status: false
> sudo yast tftp-server status enable
```

#### 4.4.3.26 **yast timezone**

配置时区。**yast timezone** 接受下列命令：

##### **list**

按地区分组列出所有可用的时区：

```
> sudo yast timezone list
Region: Africa
Africa/Abidjan (Abidjan)
Africa/Accra (Accra)
Africa/Addis_Ababa (Addis Ababa)
[...]
```

##### **set**

为时区配置指定新值：

```
> sudo yast timezone set timezone=Europe/Prague hwclock=local
```

##### **summary**

显示时区配置摘要：

```
> sudo yast timezone summary
Current Time Zone: Europe/Prague
Hardware Clock Set To: Local time
Current Time and Date: Mon 12. March 2018, 11:36:21 CET
```

#### 4.4.3.27 **yast users**

管理用户帐户。**yast users** 接受下列命令：

##### **add**

添加新用户：

```
> sudo yast users add username=user1 password=secret home=/home/user1
```

要显示完整的选项列表，请运行 **yast users add help**。

## delete

删除现有用户帐户：

```
> sudo yast users delete username=user1 delete_home
```

要显示完整的选项列表，请运行 **yast users delete help**。

## edit

更改现有用户帐户：

```
> sudo yast users edit username=user1 password=new_secret
```

要显示完整的选项列表，请运行 **yast users edit help**。

## list

列出按用户类型过滤的现有用户：

```
> sudo yast users list system
```

要显示完整的选项列表，请运行 **yast users list help**。

## show

显示有关某个用户的细节：

```
> sudo yast users show username=wwwrun
Full Name: WWW daemon apache
List of Groups: www
Default Group: wwwrun
Home Directory: /var/lib/wwwrun
Login Shell: /sbin/nologin
Login Name: wwwrun
UID: 456
```

要显示完整的选项列表，请运行 **yast users show help**。

# 5 使用 YaST 更改语言和国家/地区设置

本章介绍如何配置语言和国家/地区设置。您可以全局更改整个系统的语言，单独更改特定用户或桌面的语言，或暂时更改单个应用程序的语言。此外，您还可以配置次要语言，调整日期和国家/地区设置。

如果您在不同的国家/地区或多语言环境中工作，应相应地配置您的系统。SUSE® Linux Enterprise Desktop 可以同时处理不同的 `locales`。区域设置代表一组参数，这些参数定义在用户界面中反映的语言和国家/地区设置。

安装期间会选定主要系统语言，并相应地调整键盘和时区设置。不过，您可以安装更多语言并决定应将安装的哪种语言作为默认语言。

对于这些任务，请按[第 5.1 节 “更改系统语言”](#) 中所述使用 YaST 语言模块。如果需要以非主要语言启动应用程序或桌面，请安装次要语言以获取可选的本地化内容。

可以通过 YaST 时区模块相应地调整国家/地区和时区设置。通过它还可以将系统时钟与时间服务器进行同步。有关细节，请参见[第 5.2 节 “更改国家/地区和时间设置”](#)。

## 5.1 更改系统语言

根据您使用桌面的方式，以及您是希望将整个系统还是仅将桌面环境切换为另一种语言，可供选择的方式有以下几种：

### 全局更改系统语言

按[第 5.1.1 节 “使用 YaST 修改系统语言”](#) 和[第 5.1.2 节 “切换默认系统语言”](#) 中所述操作，以便使用 YaST 安装更多本地化软件包并设置默认语言。更改会在下次登录后生效。要确保整个系统都实施更改，请重引导系统或关闭，然后重启所有正在运行的服务、应用程序和程序。

### 仅更改桌面语言

如果先前已按照以下所述使用 YaST 为桌面环境安装了所需语言软件包，则可使用桌面的控制中心切换桌面语言。有关细节，请参见《GNOME 用户指南》，第 3 章“自定义您的设置”，第 3.2 节“配置语言设置”。重启 X 服务器之后，整个桌面将反映新选择的语言。不属于桌面框架的应用程序不受此更改的影响，仍然以 YaST 中设置的语言显示。

## 仅为一个应用程序临时切换语言

此外，还可以使用另一种语言来运行单个应用程序（该语言已使用 YaST 安装）。要实现该目的，可以按照第 5.1.3 节“切换标准 X 和 GNOME 应用程序的语言”中所述，通过指定语言代码从命令行中启动它。

### 5.1.1 使用 YaST 修改系统语言

YaST 支持两种不同的语言类别：

#### 主要语言

在 YaST 中设置的主要语言将应用于整个系统，包括 YaST 和桌面环境。除非指定其他语言，否则系统会尽可能使用这种语言。

#### 次要语言

安装次要语言，让系统支持多语种。需要时，可以手动选择作为次要语言安装的语言。例如，使用次要语言以特定语言启动应用程序，以便以这种语言进行文字处理。

在安装其他语言之前，请先确定要将其中的哪个语言用作默认的系统语言（主要语言）。

要访问 YaST 语言模块，可以启动 YaST 并单击系统 > 语言。或者，在命令行中运行 **sudo yast2 language &**，直接启动语言对话框。



## 过程 5.1：安装其他语言

安装其他语言时，YaST 允许为用户 `root` 设置不同的区域设置；请参见[步骤 4](#)。选项用户 `root` 的区域设置决定如何为 `root` 设置文件 `/etc/sysconfig/language` 中的区域设置变量 (`LC_*`)。可将它们设置为与普通用户相同的区域设置。或者，您可以不让其受任何语言变化的影响，或者仅将变量 `RC_LC_CTYPE` 设置为与普通用户相同的值。`RC_LC_CTYPE` 变量为语言特定的函数调用设置本地化配置。

- 要在 YaST 语言模块中添加语言，可以选择要安装的次要语言。
- 要将某种语言用作默认语言，可以将它设置为主要语言。
- 此外，根据新的主要语言调整键盘并调整时区（如果适用）。



### 提示：高级设置

对于高级键盘或时区设置，请在 YaST 中分别选择硬件 > 系统键盘布局，或系统 > 日期和时间。有关更多信息，请参考[第 32 章 “设置系统键盘布局”](#) 和[第 5.2 节 “更改国家/地区和时间设置”](#)。

- 要更改特定于用户 `root` 的语言设置，请单击细节。
  - 将用户 `root` 的区域设置设置为所需值。有关更多信息，单击帮助。
  - 确定是否要对 `root` 启用使用 UTF-8 编码。
- 如果可用主要语言列表中不包含您的区域设置，请尝试使用详细区域设置指定它。但这可能会导致某些区域设置不完整。
- 在对话框中单击确定确认更改。如果选择了次要语言，则 YaST 会安装对应于附加语言的软件包。

系统现在即可支持多种语言。但是，要使用主要语言之外的其他语言启动应用程序，您需要明确设置所需语言，如[第 5.1.3 节 “切换标准 X 和 GNOME 应用程序的语言”](#) 中所述。

## 5.1.2 切换默认系统语言

要全局更改系统的默认语言，请执行以下过程：

1. 启动 YaST 语言模块。
2. 选择所需的新的系统语言作为主要语言。



## 重要：删除原先的系统语言

如果切换为不同的主要语言，原主要语言的本地化软件包就会从系统中去除。要切换默认的系统语言，但要将原先的主要语言保留为附加语言，可以通过选择相应的复选框将它添加为次要语言。

3. 根据需要调整键盘和时区选项。
4. 单击确定确认更改。
5. 在 YaST 应用更改之后，重启动当前 X 会话（例如，注销并再次登录），使 YaST 和桌面应用程序反映新的语言设置。

### 5.1.3 切换标准 X 和 GNOME 应用程序的语言

使用 YaST 安装相应的语言之后，可以使用另一种语言运行单个应用程序。

在命令行中通过使用以下命令启动应用程序：

```
LANG=LANGUAGE application
```

例如，要使用德语启动 f-spot，可以运行 `LANG=de_DE f-spot`。对于其他语言，请使用相应语言代码。使用 `locale -av` 命令获取所有可用语言代码的列表。

## 5.2 更改国家/地区和时间设置

使用 YaST 日期和时间模块，将系统日期、时钟和时区信息调整为您的工作地区。要访问 YaST 模块，请启动 YaST 并单击系统 > 日期和时间。或者，在命令行中运行 `sudo yast2 timezone &`，直接启动时钟和时区对话框。



首先选择一般地区，如欧洲。选择与您工作的地区匹配的相应国家/地区，如德国。

根据工作站上运行的操作系统，相应地调整硬件时钟设置：

- 如果您的计算机上运行的是其他操作系统，例如 Microsoft Windows\*，系统可能不使用 UTC，而是使用本地时间。在这种情况下，请停用硬件时钟设置为 UTC。
- 如果在计算机上只运行 Linux，请将硬件时钟设置为 UTC，并自动将标准时间切换为夏令时。

## ! 重要：将硬件时钟设置为 UTC

如果要从标准时间自动切换到夏令时（反之亦然），则前提条件是硬件时钟（CMOS 时钟）设置为 UTC。这同样适用于借助 NTP 使用自动时间同步的情况，因为系统只有在硬件与系统时钟之间的时差少于 15 分钟时才会执行自动同步。

由于错误的系统时间可能会导致严重的问题（错过备份、丢弃邮件、在远程文件系统上挂载失败等），因此强烈建议您**始终**将硬件时钟设置为 UTC。

可以手动更改日期和时间或选项，以便将计算机与 NTP 服务器永久同步，或仅调整硬件时钟。

## 过程 5.2：手动调整时间和日期

1. 在 YaST 时区模块中，单击其他设置以设置日期和时间。
2. 选择手动并输入日期和时间值。
3. 确认更改。

## 过程 5.3：使用 NTP 服务器设置日期和时间

1. 单击其他设置以设置日期和时间。
2. 选择与 NTP 服务器同步。
3. 输入 NTP 服务器的地址（如果尚未填充）。



4. 使用配置按钮，可以打开高级 NTP 配置。有关详细信息，请参见[第 39.1 节 “使用 YaST 配置 NTP 客户端”](#)。
5. 确认更改。

# 6 使用 YaST 管理用户

在安装期间，您可能已为系统创建了本地用户。使用 YaST 模块用户和组管理可以添加用户，或编辑现有用户。它还可让您配置系统，以便在网络服务器上对用户进行身份验证。

## 6.1 用户和组管理对话框

要管理用户或组，请启动 YaST 并单击安全和用户 > 用户和组管理。或者，通过从命令行运行 `sudo yast2 users &` 来直接启动用户和组管理对话框。

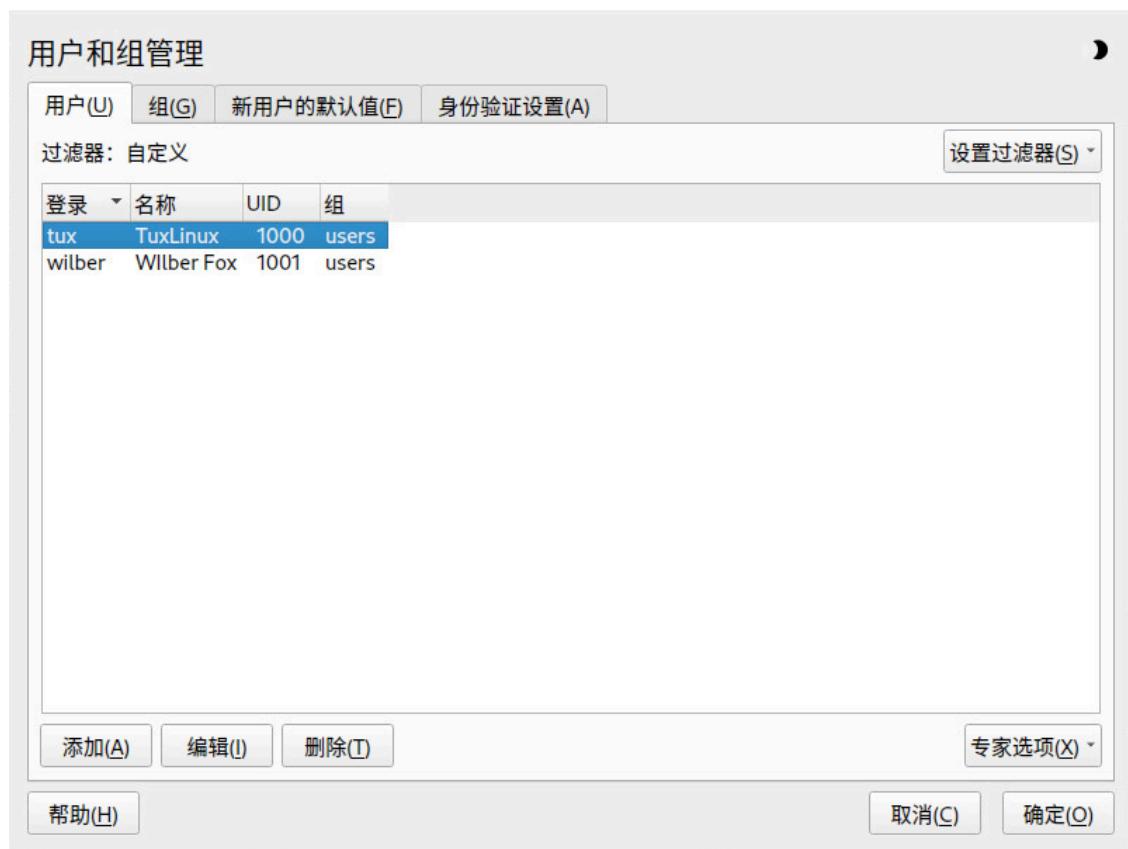


图 6.1：YaST 用户和组管理

每个用户都指派有一个系统范围的用户 ID (UID)。除了可以登录到计算机的用户之外，系统还提供了几个仅供内部使用的**系统用户**。每位用户都会被指派到一个或多个组中。与**系统用户**类似，还存在仅供内部使用的**系统组**。

根据您选择查看和修改的用户集（本地用户、网络用户、系统用户），主窗口会显示几个选项卡。这些选项卡可用于执行以下任务：

### 管理用户帐户

从用户选项卡中，创建、修改、删除或临时禁用用户帐户（如第 6.2 节“管理用户帐户”所述）。在第 6.3 节“其他用户帐户选项”中了解高级选项，例如强制实施口令策略、使用加密的主目录或管理磁盘配额。

### 更改默认设置

本地用户帐户是根据新用户的默认值选项卡中定义的设置创建的。通过第 6.4 节“更改本地用户的默认设置”可以了解到如何更改默认组指派或用户主目录的默认路径和访问权限。

### 将用户指派到组

通过第 6.5 节“将用户指派到组”可以了解到如何为单个用户更改组指派。

### 管理组

从组选项卡中，可以添加、修改或删除现有组。请参见第 6.6 节“管理组”以获取有关如何进行此操作的信息。

### 更改用户身份验证方法

如果您的计算机已连接到提供了 NIS 或 LDAP 之类的用户身份验证方法的网络上，您可以在身份验证选项卡上的若干身份验证方法中进行选择。有关详细信息，请参见第 6.7 节“更改用户身份验证方法”。

对于用户和组管理，此对话框提供了类似的功能。通过在此对话框顶部选择适当的选项卡可以轻松地在用户和组管理视图间切换。

使用过滤器选项可定义要修改的一组用户或组：在用户或组选项卡上，单击设置过滤器可查看和编辑用户或组。系统会根据本地用户或 LDAP 用户等特定类别（如果适用）列出用户或组。通过设置过滤器 > 自定义过滤器，您也可以设置和使用自定义过滤器。

此对话框可能不会提供以下所有选项和功能，具体取决于您选择的过滤器。

## 6.2 管理用户帐户

YaST 可让您创建、修改、删除或暂时禁用用户帐户。除非您是有经验的用户或管理员，否则不要修改用户帐户。



### 注意：更改现有用户的 ID

文件所有权与用户 ID，而非用户名绑定在一起。用户 ID 更改后，此用户的用户主目录中的文件会自动调整，以反映出此更改。但是，ID 更改后，此用户就不再拥有其在文件系统的其他位置创建的文件的所有权，除非手动更改这些文件的所有权。

下列说明演示了如何设置默认用户帐户。有关其他选项，请参见第 6.3 节 “其他用户帐户选项”。

#### 过程 6.1：添加或修改用户帐户

1. 打开 YaST 用户和组管理对话框并单击用户选项卡。
2. 使用设置过滤器定义要管理的用户集。对话框中会列出系统中的用户以及用户所属的组。
3. 要修改现有用户选项，请选择某一条目并单击编辑。  
要创建新的用户帐户，请单击添加。
4. 在第 1 个选项卡上输入合适的用户数据，如用户名（用于登录）和口令。这些数据足以创建新用户。如果此时单击确定，系统会自动指派用户 ID 并将所有其他值设置为默认值。
5. 若要将系统通知递送到该用户的邮箱，请激活接收系统邮件。这样，就会为 `root` 创建邮件别名，用户无需先以 `root` 身份登录便可阅读系统邮件。  
系统服务发送的邮件存储在本地邮箱 `/var/spool/mail/USERNAME` 中，其中 `USERNAME` 是所选用户的登录名。要阅读电子邮件，可以使用 `mail` 命令。
6. 要调整其他细节（如用户 ID 或用户主目录的路径），可在细节选项卡上进行。  
如果需要重新定位现有用户的用户主目录，请在该选项卡中输入新的用户主目录路径，并使用移至新位置移动当前用户主目录的内容。否则，不会使用任何的现有数据来创建新的用户主目录。
7. 要强制用户以常规方式更改口令或设置其他口令选项，请切换到口令设置并调整选项。有关详细信息，请参考 第 6.3.2 节 “强制实施口令策略”。

8. 如果已按照需要设置了所有选项, 请单击确定。
9. 单击确定以关闭管理对话框并保存更改。此时, 新添加的用户可以使用您创建的登录名和口令登录系统。  
或者, 要保存所有更改且不退出用户和组管理对话框, 请单击专家选项 > 立即写入更改。



## 提示: 匹配用户 ID

将 (本地) 用户 ID 与网络中的 ID 进行匹配会很有用。例如, 应该将便携式计算机上的新 (本地) 用户集成到包含相同用户 ID 的网络环境中。这样可确保用户“脱机”创建的文件的所有权和其直接在网络上创建的文件的所有权相同。

### 过程 6.2 : 禁用或删除用户帐户

1. 打开 YaST 用户和组管理对话框并单击用户选项卡。
2. 要在不删除用户帐户的情况下临时禁用该帐户, 请从列表中选择该用户并单击编辑。激活禁用用户登录。再次启用该帐户之前, 此用户不能登录您的计算机。
3. 要删除用户帐户, 请从列表中选择该用户并单击删除。选择您是要删除用户的主目录还是保留该数据。

## 6.3 其他用户帐户选项

除了默认用户帐户的设置外, SUSE® Linux Enterprise Desktop 还提供了其他选项。例如, 用于强制实施口令策略、使用加密的主目录, 或者为用户和组定义磁盘配额的选项。

### 6.3.1 自动登录和无口令登录

如果使用的是 GNOME 桌面环境, 则可为特定用户配置**自动登录**, 为所有用户配置**无口令登录**。自动登录使用户在引导时自动登录到桌面环境。一次只能为一位用户激活此功能。使用无口令登录可以让用户在登录管理器中输入其用户名后直接登录系统。



## 警告：安全风险

在多人可以访问的计算机上启用**自动登录**或**无口令登录**具有一定的安全性风险。无需身份验证，任何用户都能访问您的系统和数据。如果系统包含机密数据，请勿使用此功能。

要激活自动登录或无口令登录，请通过 YaST 用户和组管理中的专家选项 > 登录设置来访问这些功能。

### 6.3.2 强制实施口令策略

在有多个用户的系统上，最好至少强制实施基本的口令安全性策略。用户应该定期更改其口令并使用不能轻易识破的可靠口令。对于本地用户，请执行以下操作：

#### 过程 6.3：配置口令设置

1. 打开 YaST 用户和组管理对话框并选择用户选项卡。
2. 选择用户并单击编辑。
3. 切换至口令设置选项卡。用户的上次口令更改会显示在该选项卡上。
4. 要让用户在下次登录时更改其密码，请激活强制密码更改。
5. 要实施口令转换，请设置相同口令的最大有效天数和相同口令的最小有效天数。
6. 要在口令失效前提醒用户更改口令，请为口令失效前多少天发出警告设置数值。
7. 要限制密码失效后用户可以登录的时间周期，请更改密码失效后仍可登录的天数中的值。
8. 您也可为整个帐户指定一个特定的失效日期。输入格式为 YYYY-MM-DD 的失效日期。此设置与口令无关，而是应用于帐户本身。
9. 有关选项和默认值的更多信息，请单击帮助。
10. 单击确定应用您的更改。

### 6.3.3 管理配额

为了防止系统容量在没有通知的情况下耗尽，系统管理员可以为用户或组设置配额。可以为一个或多个文件系统定义配额，该配额限制可使用的磁盘空间量和可在该处创建的 inode（索引节点）数。Inode 是文件系统上的一种数据结构，用于存储普通文件、目录或其他文件系统对象的基本信息。该结构会存储文件系统对象的所有属性（如用户和组所有权、读取、写入或执行权限），但不会存储文件名和内容。

SUSE Linux Enterprise Desktop 允许使用 soft 和 hard 配额。另外，可以定义宽限间隔，允许用户或组暂时超出配额一定的数量。

#### 软配额

定义一个警告级别，用于在接近限制时告知用户。管理员可能会要求用户清理并减少分区上的数据。软配额限制通常低于硬配额限制。

#### 硬配额

定义一个限制，达到此限制时拒绝写入请求。如果达到了硬配额，则不再可以存储数据，并且应用程序可能会崩溃。

#### 宽限期

定义在超出软配额之后，经过多长时间再发出警告。通常设置为一个相当小的值，例如一小时或数小时。

#### 过程 6.4：为分区启用配额支持

要为特定用户和组配置配额，需要先在 YaST 专家分区程序中为相应的分区启用配额支持。

1. 在 YaST 中，选择系统 > 分区程序并单击是以继续。
2. 在专家分区程序中，选择要启用配额的分区并单击编辑。
3. 单击 Fstab 选项并激活启用配额支持。如果尚未安装 quota 软件包，当您单击是确认相应的消息时，系统就会安装该软件包。
4. 确认您的更改，然后离开专家分区程序。
5. 输入以下命令确保 quotaon 服务正在运行：

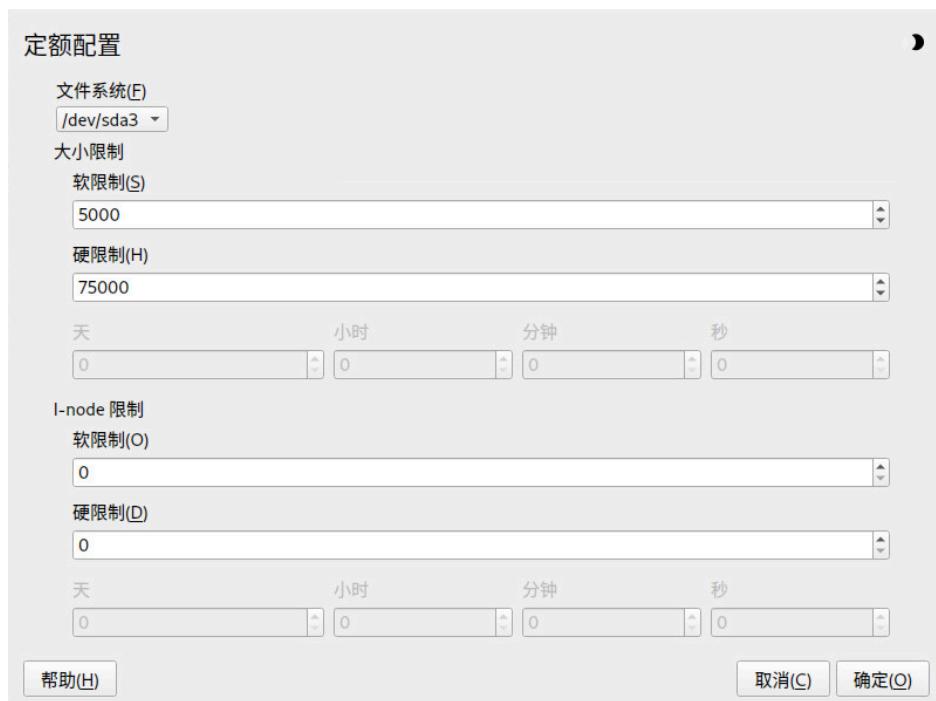
```
> sudo systemctl status quotaon.service
```

它应该标记为处于 `active` 状态。如果情况并非如此，请使用命令 `systemctl start quotaon.service` 将其启动。

#### 过程 6.5：为用户或组设置配额

现在，您可以为特定用户或组设置软配额或硬配额，并可设置时间周期作为宽限间隔。

1. 在 YaST 用户和组管理中，选择想要设置配额的用户或组并单击编辑。
2. 在插件选项卡上，选择管理用户配额项，并单击起动打开配额配置对话框。
3. 从文件系统中，选择应应用配额的分区。



4. 在大小限制下，限制磁盘空间量。请输入用户或组在此分区上可拥有的大小为 1 KB 的块数。指定软限制和硬限制值。
5. 另外，也可限制用户或组在分区上可拥有的 inode 数。在 Inode 限制中，输入软限制和硬限制
6. 仅当用户或组已超出指定的大小或 inode 软限制时，才能定义宽限间隔。否则，与时间有关的文本框不会处于激活状态。指定允许用户或组超出以上设置的限制的时间周期。
7. 单击确定确认您的设置。

8. 单击确定以关闭管理对话框并保存更改。

或者, 要保存所有更改且不退出用户和组管理对话框, 请单击专家选项, 立即写入更改。

SUSE Linux Enterprise Desktop 还随附了 `repquota` 或 `warnquota` 等命令行工具。系统管理员可以使用这些工具来控制磁盘用量, 或者向超出配额的用户发送电子邮件通知。管理员也可使用 `quota_nld` 向 D-BUS 转发与已超出配额有关的内核消息。有关更多信息, 请参见 `repquota`、`warnquota` 和 `quota_nld` 手册页。

## 6.4 更改本地用户的默认设置

创建新的本地用户时, YaST 将使用几个默认设置。例如, 这些设置包括用户所属的主组和次组或用户的用户主目录访问权限。您可以更改这些默认设置来满足要求:

1. 打开 YaST 用户和组管理对话框并选择新用户默认值选项卡。
2. 要更改新用户应自动归入的主组, 请从默认组中选择另一个组。
3. 要修改新用户的次组, 请在次要组中添加或更改组。组名必须用逗号隔开。
4. 如果不想使用 `/home/USERNAME` 作为新用户主目录的默认路径, 请修改主目录的路径前缀。
5. 要更改新建用户主目录的默认许可权限模式, 请调整用户主目录的权限掩码中的权限掩码值。有关权限掩码的更多信息, 请参见《安全和强化指南》, 第 19 章 “Linux 中的访问控制列表” 和 `umask` 手册页。
6. 有关各个选项的信息, 请单击帮助。
7. 单击确定应用您的更改。

## 6.5 将用户指派到组

系统会根据您可从用户和组管理对话框中新用户的默认值选项卡内访问的默认设置, 将本地用户指派给数个组。通过以下内容可以了解到修改单个用户的组指派的方法。如需更改新用户的默认组指派, 请参见第 6.4 节 “更改本地用户的默认设置”。

#### 过程 6.6：更改用户的组指派

1. 打开 YaST 用户和组管理对话框并单击用户选项卡。该选项卡会列出用户及用户所属的组。
2. 单击编辑并切换到细节选项卡。
3. 要更改用户所属的主组，请单击默认组并从列表中选择该组。
4. 要将用户指派给其他次组，请在其他组列表中激活对应的复选框。
5. 单击确定以应用您的更改。
6. 单击确定以关闭管理对话框并保存更改。

或者，要保存所有更改且不退出用户和组管理对话框，请单击专家选项，立即写入更改。

## 6.6 管理组

使用 YaST 还能轻松添加、修改或删除组。

#### 过程 6.7：创建和修改组

1. 打开 YaST 用户和组管理对话框并单击组选项卡。
2. 使用设置过滤器定义想要管理的组集。对话框会列出系统中的组。
3. 要创建新组，请单击添加。
4. 要修改现有组，请选择该组并单击编辑。
5. 在以下对话框中，输入或更改数据。右侧列表显示了可以成为该组成员的所有可用用户和系统用户的概述。



6. 要将现有用户添加到新组中，请通过选中对应的框来从可能的组成员列表中选择这些用户。要从组中去除这些用户，请停用对应的框。
7. 单击确定以应用您的更改。
8. 单击确定以关闭管理对话框并保存更改。

或者，要保存所有更改且不退出用户和组管理对话框，请单击专家选项 > 立即写入更改。

要删除一个组，该组中不得包含任何组成员。要删除某个组，请从列表中选择该组并单击删除。单击确定以关闭管理对话框并保存更改。或者，要保存所有更改且不退出用户和组管理对话框，请单击专家选项 > 立即写入更改。

## 6.7 更改用户身份验证方法

如果计算机已连接到网络，您可以更改身份验证方法。下列选项可用：

### NIS

在 NIS 服务器上对网络中的所有系统进行集中用户管理。有关详细信息，请参见《安全和强化指南》，第 3 章 “使用 NIS”。

## SSSD

**系统安全服务守护程序 (SSSD)** 可在本地缓存用户数据，并可让用户使用这些数据，即使实际目录服务（暂时）不可访问时也不例外。有关详细信息，请参见《安全和强化指南》，第 4 章 “使用 YaST 设置身份验证客户端”，第 4.2 节 “SSSD”。

## Samba

在 Linux 和 Windows 混用的网络中经常使用 SMB 身份验证。有关详细信息，请参见《安全和强化指南》，第 7 章 “Active Directory 支持”。

要更改身份验证方法，请执行以下操作：

1. 打开 YaST 中的用户和组管理对话框。
2. 单击身份验证设置选项卡以显示可用身份验证方法和当前设置的概述。
3. 要更改身份验证方法，请单击配置并选择想要修改的身份验证方法。随后您将直接转到 YaST 中的客户端配置模块。有关相应客户端配置的信息，请参见以下部分：

**NIS:** 《安全和强化指南》，第 3 章 “使用 NIS”，第 3.2 节 “配置 NIS 客户端”

**LDAP:** 《安全和强化指南》，第 4 章 “使用 YaST 设置身份验证客户端”，第 4.1 节 “使用 YaST 配置身份验证客户端”

**SSSD:** 《安全和强化指南》，第 4 章 “使用 YaST 设置身份验证客户端”，第 4.2 节 “SSSD”

4. 接受配置后，请返回到用户和组管理概述。
5. 单击确定以关闭管理对话框。

## 6.8 默认系统用户

SUSE Linux Enterprise Desktop 默认会创建一些不可删除的用户名。通常在 Linux Standard Base 中定义这些用户。下面的列表提供了常见用户名及其用途：

## 默认安装的常见用户名

bin,

daemon

旧式用户，为了与旧式应用程序兼容而提供。新式应用程序应该不再使用此用户名。

gdm

GNOME 显示管理器守护程序 (GDM) 使用此用户名来提供图形登录，以及管理本地和远程显示内容。

lp

由通用 Unix 打印系统 (CUPS) 的打印机守护程序使用。

mail

为 sendmail 或 postfix 等邮件程序保留的用户。

man

由 man 用来访问手册页。

messagebus

用于访问 D-Bus (桌面总线，用于进程间通讯的软件总线)。守护程序是 dbus-daemon。

nobody

不拥有任何文件且属于无权限组的用户。目前，其用途有限，因为 Linux Standard Base 建议为每个守护程序提供单独的用户帐户。

nscd

由名称服务缓存守护程序使用。此守护程序是用于改善 NIS 和 LDAP 性能的查找服务。守护程序是 nscd。

polkitd

由 PolicyKit 授权框架 (用于定义和处理非特权进程的授权请求) 使用。守护程序是 polkitd。

postfix

由 Postfix 邮件程序使用。

## pulse

由 Pulseaudio 声音服务器使用。

## root

由提供所有适当权限的系统管理员使用。

## rpc

由 **rpcbind** 命令 (RPC 端口映射器) 使用。

## rtkit

由提供 D-Bus 系统服务以实现实时调度模式的 **rtkit** 软件包使用。

## salt

Salt 提供的并行远程执行的用户。守护程序名为 **salt-master**。

## scard

用于与智能卡和读卡器通讯的用户。守护程序名为 **pcscd**。

## srvGeoClue

GeoClue D-Bus 服务使用它来提供位置信息。

## sshd

安全外壳守护程序 (SSH) 使用它来确保在非安全网络上进行安全的加密通讯。

## statd

**rpc.statd** 守护程序中实施的网络状态监视协议 (NSM) 使用它来侦听重引导通知。

## systemd-coredump

**/usr/lib/systemd/systemd-coredump** 命令使用它来获取、保存和处理核心转储。

## systemd-timesync

**/usr/lib/systemd/systemd-timesyncd** 命令使用它将本地系统时钟与远程网络时间协议 (NTP) 服务器同步。

## 7 YaST 联机更新

SUSE 持续为您的产品提供软件安全更新。默认使用更新小程序来确保系统处于最新状态。有关更新小程序的更多信息请参考第 8.5 节 “GNOME Package Updater” 。本章介绍用于更新软件包的备用工具：YaST 联机更新。

更新软件储存库提供了 SUSE® Linux Enterprise Desktop 的最新补丁。如果安装时已注册您的产品，则更新储存库已配置。如果您尚未注册 SUSE Linux Enterprise Desktop，可通过在 YaST 中启动产品注册来完成注册。或者，可以从信任的源中手动添加更新储存库。要添加或去除储存库，请使用 YaST 中的软件 > 软件储存库来启动储存库管理器。请在第 8.4 节 “管理软件储存库和服务” 中了解更多有关储存库管理器的内容。



### 注意：访问更新编目时出错

如果您无法访问更新编目，原因可能是订阅已过期。通常，SUSE Linux Enterprise Desktop 会附带一年或三年的订阅，在此期间您可以访问更新编目。订阅结束后，将拒绝您访问更新编目。

如果访问更新编目时遭到拒绝，您会看到一条警告消息，提示您访问 SUSE Customer Center 以查看您的订阅。可通过 <https://scc.suse.com/> 访问 SUSE Customer Center。



### 注意：用于接收更新的防火墙设置

SUSE Linux Enterprise Desktop 上的防火墙默认只阻止传入连接。如果系统受到会阻止传出流量的另一个防火墙的保护，请确保允许通过端口 80 和 443 连接至 <https://scc.suse.com/> 和 <https://updates.suse.com>，以便接收更新。

SUSE 提供了不同相关级别的更新：

#### 安全性更新

修复严重的安全性危害，请务必安装。

#### 推荐更新

修复可能危及计算机安全的问题。

## 可选更新

修复非安全性相关的问题或提供增强功能。

## 7.1 联机更新对话框

要打开 YaST 联机更新对话框, 请启动 YaST 并选择软件 > 联机更新。也可以从命令行输入 **yast2 online\_update** 来启动该模块。

联机更新窗口由四部分组成。

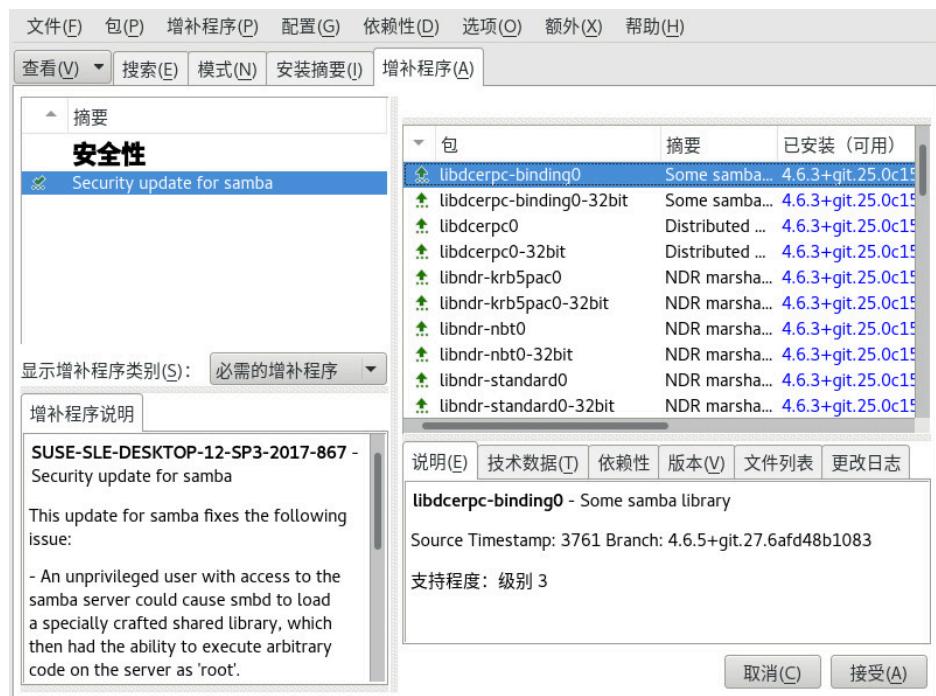


图 7.1：YaST 联机更新

左侧的摘要部分列出了 SUSE Linux Enterprise Desktop 可用的补丁。补丁按安全相关性 (security、recommended 和 optional) 排序。您可以从显示补丁类别中选择以下某个选项来更改摘要部分的视图：

### 必需的补丁 (默认视图)

当前未安装的适用于系统上已安装的软件包的补丁。

### 不必要的补丁

适用于系统上未安装的软件包的补丁, 或要求已满足的补丁 (因为已从另一源对相关软件包进行了更新)。

## 所有补丁

SUSE Linux Enterprise Desktop 的所有可用补丁。

摘要部分的每个列表项都由符号和补丁名称组成。如需了解可能符号及其含义的概述，请按 **Shift + F1**。系统会自动预设需要对 **Security** 和 **Recommended** 补丁执行的操作。这些操作有自动安装、自动更新和自动删除。

如果从非更新储存库的某个储存库安装最新软件包，此安装可能满足此软件包的某个补丁的要求。在这种情况下，在补丁摘要前会显示一个复选标记。该补丁会显示在列表中，直到将其标记为待安装。这不会安装补丁（因为该软件包已经是最新的），而是将该补丁标记为已安装。在摘要部分选择一个项，可在对话框左下角的补丁描述中查看简短描述。右上部分列出所选补丁中包含的软件包（一个补丁可以由多个软件包组成）。单击右上部分中的项可以查看有关补丁中包含的各个软件包的细节。

## 7.2 安装补丁

在 YaST 的“联机更新”对话框中，您可以一次性安装所有可用的补丁，也可以手动选择所需的补丁。还可以还原已应用于系统的补丁。

默认情况下，您的系统当前可用的所有新补丁（**optional** 补丁除外）都已标记为待安装。一旦您单击接受或应用，将自动应用它们。如果一个或多个补丁需要重引导系统，在开始安装补丁之前，系统会发出相关通知。此时，您可以选择继续安装所选补丁、跳过需要重引导的所有补丁的安装并安装剩余的补丁，或者返回补丁手动选择屏幕。

### 过程 7.1：使用 YaST 联机更新应用补丁

1. 启动 YaST 并选择软件 > 联机更新。
2. 要自动应用您的系统当前可用的所有新补丁（**optional** 补丁除外），请单击应用或接受。
3. 首先请修改要应用的补丁选择：
  - a. 使用界面中提供的相应过滤器和视图。有关细节，请参见第 7.1 节“联机更新对话框”。
  - b. 根据您的需要和喜好选择或取消选择补丁，方法是右键单击补丁并从上下文菜单中选择相应操作。



## 重要：始终应用安全更新

除非有必要，否则请不要取消选择任何与 `security` 相关的补丁。因为这些补丁可修复严重的安全性危害，防止系统遭受攻击。

- c. 多数补丁包含几个软件包的更新。要更改对单个软件包执行的操作，请在包视图中右键单击某个软件包，并选择一项操作。
  - d. 要确认您的选择并应用所选补丁，请单击 **应用** 或 **接受** 以继续。
4. 安装完成后，单击完成退出 YaST 联机更新。您的系统现在已经是最新的。

## 7.3 查看已撤回的补丁

维护更新会经过仔细测试，以最大限度地降低引入 Bug 的风险。如果发现某个补丁包含 Bug，将会自动撤回该补丁。一个新的更新（版本号更高）将会发布以还原有 Bug 的补丁，并会阻止再次安装该补丁。可以在软件包分类选项卡上查看已撤回的补丁及其历史记录。

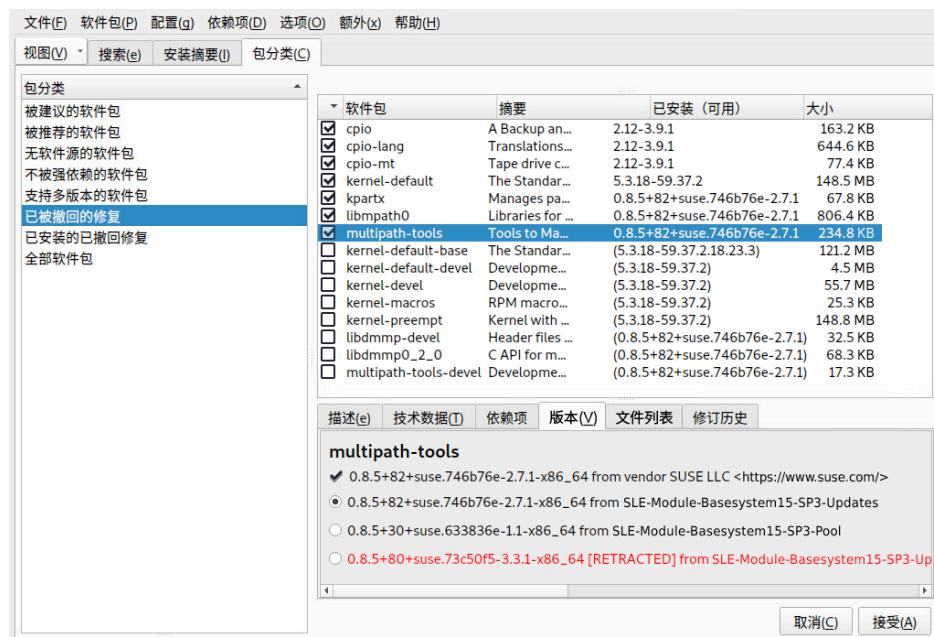


图 7.2：查看已撤回的补丁和历史记录

## 7.4 自动联机更新

您可以使用 YaST 配置每日、每周或每月执行的自动更新。安装 [yast2-online-update-configuration](#) 软件包。

默认情况下，更新将以增量 RPM 的形式供您下载。由于基于增量 RPM 重构建 RPM 软件包需要占用大量内存和处理器资源，出于性能考虑，某些设置或硬件配置可能要求您禁用增量 RPM。某些补丁（如需要许可协议的内核更新或软件包）需要用户交互，这可能会导致自动更新过程停止。您可以配置以跳过需要用户交互的补丁。

使用 YaST 软件模块中的补丁选项卡可以查看可用的及已安装的补丁，包括 Bug 报告和 CVE 公告的参考内容。

### 过程 7.2：配置自动联机更新

1. 安装后，启动 YaST 并选择软件 > 联机更新。选择配置 > 联机更新。如果未安装 [yast2-online-update-configuration](#)，系统将提示您加以安装。



图 7.3：YaST 联机更新配置

也可以从命令行输入 [yast2 online\\_update\\_configuration](#) 来启动该模块。

2. 选择更新间隔：每日、每周或每月。

**3.** 有时，补丁可能需要引起管理员的注意，例如，在重启动关键服务时。而此补丁可能是适用于 Docker 开放源代码引擎的更新，运行此更新需要重启动所有容器。在安装补丁之前，系统会向用户告知此操作的后果，并要求他们确认是否安装该补丁。此类补丁称为“交互式补丁”。

自动安装补丁时，系统会假设您已确认要安装交互式补丁。如果您希望在安装之前先检查这些补丁，请选中跳过交互式补丁。在这种情况下，自动修补期间会跳过交互式补丁。请确保定期运行手动联机更新，以检查是否有交互式补丁正在等待安装。

**4.** 要自动接受任何许可协议，请激活同意许可证。

**5.** 要自动安装更新包推荐的所有软件包，请激活包含推荐的软件包。

**6.** 要禁用增量 RPM（出于性能方面的考虑），请取消选中使用增量 RPM。

**7.** 要按照类别（例如安全性或推荐）过滤补丁，请选中按类别过滤，并从列表中添加适当的补丁类别。只会安装选中类别的补丁，合理做法是仅启用自动安全更新，并手动检查所有其他更新。修补通常是可靠的，但您可能想要测试非安全性补丁，并在遇到任何问题时进行回滚。

- Packagemanager 和 YaST 提供的补丁用于软件包管理以及 YaST 功能和模块。
- 安全性补丁提供至关重要的更新和 Bug 修复。
- 推荐补丁提供可选的 Bug 修复和增强功能。
- 可选表示新软件包。
- 其他相当于杂项。
- 文档不可用。

**8.** 单击确定确认您的配置。

之后，自动联机更新将不会自动重启动系统。如果有的软件包更新需要重引导系统，您需要手动重引导。

# 8 安装或去除软件

使用 YaST 的软件管理模块，您可以搜索、安装和去除软件包。安装软件包时，YaST 将自动解析所有依赖项。要安装为保存在安装媒体中的软件包，可以添加软件储存库和 YaST 来管理它们。您还可以使用更新小程序管理软件更新，让系统保持最新状态。

YaST 软件管理器让您可以管理系统上的储存库。此 YaST 模块有两个版本：一个是 X Window 的图形版本，另一个是搭配命令行使用的基于文本的版本。下文会介绍图形变体 — 有关基于文本的 YaST 的细节，请参见[第 4 章 “文本模式下的 YaST”](#)。



## 注意：确认和检查更改

安装、更新或去除软件包时，只有在单击接受或应用之后，才会应用软件管理器中的所有更改。YaST 有一个包含所有操作的列表，允许您在将更改应用于系统前对更改进行检查和修改。

## 8.1 术语定义

要了解如何在 SUSE Linux Enterprise Desktop 中安装和去除软件，必须理解以下术语。

### 储存库

包含软件包和有关这些软件包的更多信息（软件包元数据）的本地或远程目录。

### （储存库）别名/储存库名称

储存库的简短名称（在 Zypper 中称为 Alias，在 YaST 中称为储存库名称）。该别名可以由用户在添加储存库时选择且必须唯一。

### 储存库描述文件

每个储存库都提供描述储存库内容（软件包名称、版本等）的文件。这些储存库描述文件将下载到 YaST 使用的本地缓存中。

### 产品

表示整个产品，例如 SUSE® Linux Enterprise Desktop。

## 软件集

软件集是一组专用于某种用途的可安装软件包。例如，Laptop 软件集包含移动计算环境中所需的所有软件包。软件集定义软件包依赖项（比如必需的软件包或推荐的软件包），并预选择了标记为安装的软件包。这可确保在安装此软件集后某种用途所需的最重要的软件包在系统上可用。如有必要，您可以手动选择或取消选择某个软件集中的软件包。

## 软件包

软件包是 rpm 格式的压缩文件，包含特定程序的文件。

## 补丁

补丁由一个或多个软件包组成，可通过增量 RPM 应用。它也可能带来尚未安装的软件包的依赖项。

## 可解决

用于产品、模式、软件包或补丁的一个常用术语。最常见的可解决类型为软件包或补丁。

## 增量 RPM

增量 RPM 仅包含某个软件包的两个已定义版本之间的有区别二进制文件，因此其下载大小最小。安装前，需要在本地计算机上重构完整 rpm 软件包。

## 软件包依赖项

一些软件包依赖于其他软件包，例如共享库。换句话说，一个软件包可能会 require 其他软件包，如果必需的软件包不可用，便无法安装软件包。除了必须满足的依赖关系（软件包要求）之外，某些软件包还会 recommend 其他软件包。这些推荐的软件包仅在实际可用时才安装，否则将忽略它们，但是会安装推荐它们的软件包。

## 8.2 注册已安装的系统

如果您在安装期间跳过了注册，或者想重新注册系统，随时都可以注册系统。使用 YaST 模块产品注册或命令行工具 SUSEConnect。

### 8.2.1 使用 YaST 注册

要注册系统，请启动 YaST，并依次切换到软件和产品注册。

系统默认会在 SUSE Customer Center 中注册。如果您的组织提供了本地注册服务器，您可以从自动检测到的服务器列表中选择一个服务器，或手动提供 URL。

## 8.2.2 使用 SUSEConnect 注册

要从命令行注册，请使用命令

```
> sudo SUSEConnect -r REGISTRATION_CODE -e EMAIL_ADDRESS
```

将 REGISTRATION\_CODE 替换为与 SUSE Linux Enterprise Desktop 副本一同收到的注册码。

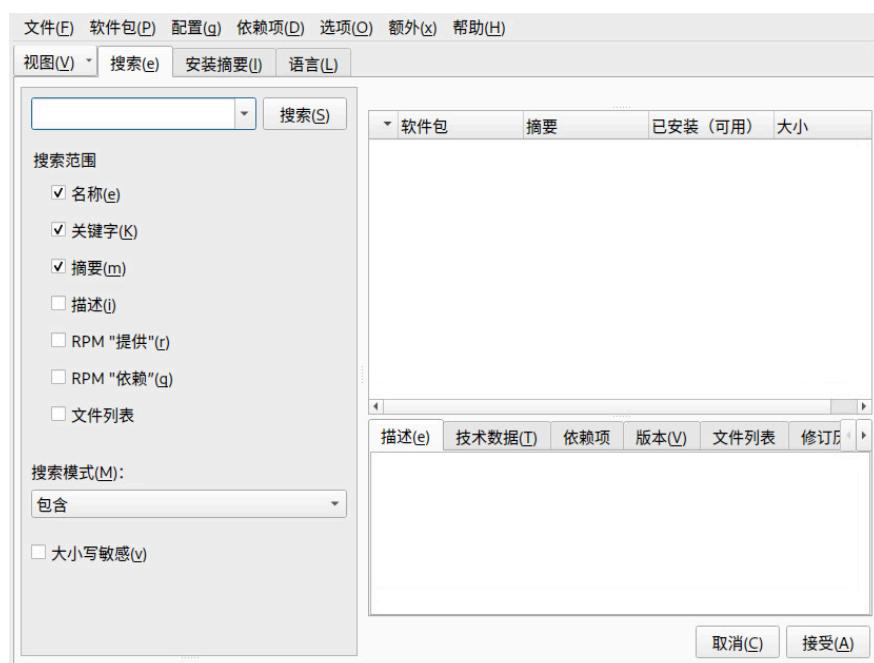
将 EMAIL\_ADDRESS 替换为与您或贵组织管理订阅时所用 SUSE 帐户关联的电子邮件地址。

要使用本地注册服务器注册，还需提供该服务器的 URL：

```
> sudo SUSEConnect -r REGISTRATION_CODE -e EMAIL_ADDRESS --url "URL"
```

## 8.3 使用 YaST 软件管理器

从 YaST 控制中心选择软件 > 软件管理可以启动软件管理器。



## 8.3.1 搜索软件

YaST 软件管理器可以从所有当前启用的储存库中安装软件包或软件集。它提供了不同的视图和过滤器，帮助更方便地查找要搜索的软件。搜索视图是窗口的默认视图。要更改视图，请单击视图并从下拉框中选择以下项之一。选定视图会在新的选项卡中打开。

### 搜索软件包或软件集的视图

#### 软件集

列出系统上所有可用于安装的软件集。

#### 软件包组

列出所有软件包，按组排序，例如图形、编程或安全性。

#### 语言

一个过滤器，用于列出添加某个新系统语言所需的所有软件包。

#### 安装源

一个过滤器，用于按储存库列出软件包。要选择多个储存库，请按住 `Ctrl` 键的同时单击储存库名称。“伪储存库” `@System` 会列出当前安装的所有软件包。

#### 服务

显示哪些软件包属于特定的模块或扩展。选项一个项（例如 `Basesystem` 或 `High Availability`）可以显示属于此模块或扩展的软件包列表。

#### 搜索

可让您按照特定准则搜索软件包。输入搜索条件并按下 `Enter`。通过指定搜索范围和更改搜索模式来优化搜索。例如，如果您不知道软件包名称，只知道要搜索的应用程序名称，请尝试在搜索过程中包括软件包描述。

#### 安装摘要

如果已选择要安装、更新或去除的软件包，此视图会显示单击接受后将应用于系统的更改。要过滤此视图中具有特定状态的软件包，请激活或停用相应的复选框。要了解状态标志的细节，请按 `Shift + F1`。



## 提示：查找不属于活动储存库的软件包

要列出所有不属于活动安装源的软件包，可以选择视图 > 安装源 > @System，然后选择次过滤器 > 未维护的软件包。例如，如果您删除了某个储存库，并想要确保系统上不再安装该储存库中的任何软件包，则此操作就非常有用。



## 提示：联机搜索软件

联机搜索功能可以搜索所有已注册和未注册模块和扩展的软件包。

### 过程 8.1：联机搜索软件

要联机搜索软件包，请执行以下步骤：

1. 单击额外 > 联机搜索打开联机搜索窗口。
2. 输入软件包名称，然后按 **Enter** 键或单击搜索。YaST 会连接 SUSE Customer Center 并在表格中显示结果，其中包括每个软件包的模块或扩展。选择某个软件包可查看更多细节。
3. 单击表格中相应的行和切换选择，选择一个或多个要安装的软件包。或者，您也可以双击某一行。如果软件包属于未注册的模块或扩展，YaST 会向您确认是否注册该模块或扩展。
4. 单击下一步，检查更改，然后安装软件包。

### 8.3.2 安装和去除软件包或软件集

一些软件包依赖于其他软件包，例如共享库。一些软件包不能与其他软件包在系统中共存。如果可行，YaST 会自动解决这些依赖项或冲突。如果您的选择导致无法自动解决的依赖项冲突，则需要按照第 8.3.4 节 “软件包依赖项” 中所述手动解决冲突。



## 注意：软件包的去除

去除软件包时，默认情况下 YaST 仅去除所选的软件包。如果希望 YaST 还去除在去除指定软件包后变为不需要的所有其他软件包，请从主菜单中选择选项 > 删除软件包时清理。

1. 按照第 8.3.1 节 “搜索软件” 中所述搜索软件包。
2. 右窗格中会列出找到的软件包。要安装或删除某个软件包，可以右键单击它并选择安装或删除。如果相关选项不可用，可以检查软件包名称前的符号指示的软件包状态 — 按 **Shift + F1** 可获取帮助。



### 提示：对列出的所有软件包应用操作

要对右窗格中列出的所有软件包应用某项操作，请转到主菜单，并从软件包 > 此列表中的全部内容中选择操作。

3. 要安装某个软件集，可以右键单击软件集名称并选择安装。
  4. 无法去除软件集。但可以选择软件集中要去除的软件包，并将它们标记为待去除。
  5. 要选择更多的软件包，请重复上述步骤。
  6. 在应用您的更改前，可以通过单击视图 > 安装摘要对它们进行检查或修改。默认会列出所有将更改状态的软件包。
  7. 要还原某个软件包的状态，可以右键单击该软件包并选择以下任一项：保留（如果已安排要删除或更新软件包），或不安装（如果已安排要安装软件包）。要放弃所有更改并退出软件管理器，请单击取消和放弃。
  8. 完成后，单击接受应用您的更改。
  9. 如果 YaST 找到其他依赖项，会显示要安装、更新或去除的相关软件包的列表。单击继续接受它们。
- 安装、更新或去除所有选定软件包后，YaST 软件管理器会自动关闭。



### 注意：安装源软件包

目前无法通过 YaST 软件管理器安装源软件包。为了实现该目的，可以使用命令行工具 **zypper**。有关详细信息，请参见第 9.1.3.5 节 “安装或下载源软件包”。

### 8.3.3 更新软件包

除了更新单个软件包外，还可以从某个储存库更改所有安装的软件包或所有软件包。大批量更新软件包时，通常需要注意以下几个方面：

- 提供软件包的储存库的优先级，
- 软件包的体系结构（例如 AMD64/Intel 64），
- 软件包的版本号，
- 软件包的供应商。

哪个方面对选择更新候选软件包最重要，取决于您选择的相应更新选项。

1. 要将所有安装的软件包更新到最新版本，请从主菜单中选择软件包 > 所有软件包，如有较新版本可用则更新。

系统会根据以下策略在所有储存库中检查有无可用的更新候选软件包：YaST 会先尝试将搜索范围限制在与已安装软件包的体系结构和供应商相同的软件包。如果找到结果，将根据以下过程选择“最佳”更新候选软件包。但是，如果找不到具有相同供应商的可比较软件包，则将搜索范围扩大到具有相同体系结构的所有软件包。如果还是找不到可比较软件包，将考虑所有软件包，并根据以下准则选择“最佳”更新候选软件包：

1. 储存库优先级：使用优先级最高的储存库中的软件包。
2. 如果这种选择的结果得出多个软件包，请选择体系结构“最佳”的软件包（最佳选择：与已安装软件包的体系结构匹配）。

如果得到的软件包的版本号比已安装软件包高，将用所选的更新候选软件包更新并替换已安装软件包。

此选项会试图避免更改已安装软件包的体系结构和供应商，但在某些情况下，会容许此类更改。



#### 注意：无条件更新

如果选择的是软件包 > 所有软件包 > 无条件更新，则会应用相同的准则，但会无条件安装找到的任何候选软件包。因此，选择此选项可能会使某些软件包降级。

2. 确保用于大批量更新的软件包来自特定安装源：

- a. 请按照第 8.3.1 节 “搜索软件” 中所述选择要从中更新的安装源。
  - b. 在窗口的右侧，单击将系统软件包切换为此储存库中的版本。此操作明确允许 YaST 在替换软件包时更改软件包供应商。  
单击接受后，所有已安装的软件包都将替换为来自此储存库的软件包（如果可用）。这样会导致供应商和体系结构变更，甚至会使某些软件包降级。
  - c. 要避免这种情况，请单击取消将系统软件包切换为此储存库中的版本。只有在单击接受按钮后才能取消。
3. 在应用您的更改前，可以通过单击视图 > 安装摘要对它们进行检查或修改。默认情况下，会列出所有将更改状态的软件包。
4. 如果所有选项都按照您的喜好设置完毕，请单击接受确认您的更改以启动大批量更新。

#### 8.3.4 软件包依赖项

大多数软件包都依赖于其他软件包。例如，如果软件包使用了某个共享库，则它依赖于提供该库的软件包。一些软件包若共存则会导致冲突（例如，邮件传送代理 sendmail 与 postfix 之中，您只能安装一个）。安装或删除软件时，软件管理器会确保不留下任何未解决的依赖项或冲突，以保证系统完整性。

当依赖项或冲突只有一种解决方法时，将会自动解决依赖项或冲突。存在多种解决方法总是会导致需要手动解决的冲突。如果解决冲突涉及到更改供应商或体系结构，则它也需要手动解决。在软件管理器中单击接受应用更改时，会显示由自动解析程序触发的所有操作的概述，需要您确认。

默认情况下，会自动检查依赖项。每次更改软件包的状态时（例如，将某个软件包标记为供安装或删除），都会执行检查。通常它很有用，但在手动解决依赖项冲突时会令人厌烦。要禁用此功能，请转到主菜单，然后停用依赖项 > 自动检查。通过依赖项 > 立即检查手动执行依赖项检查。在单击接受来确认您的选择时，总是会执行一致性检查。

要查看软件包的依赖项，可以单击右键并选择显示解析程序信息。此时会打开显示依赖项的图。已安装的软件包显示在绿框中。



## 注意：手动解决软件包冲突

除非有经验，否则在处理软件包冲突时请遵循 YaST 所做的建议，不然可能无法解决它们。请注意，您所做的每个更改都可能会触发其他冲突，因此最终很容易导致冲突数量不断上升。发生这种情况时，请取消软件管理器，放弃所有更改并重新开始。



图 8.1：软件管理器的冲突管理

### 8.3.5 处理软件包建议

除了具有运行程序所需的硬依赖项（例如特定的库）以外，软件包还可能具有弱依赖项（例如，增加额外的功能或翻译）。这些弱依赖项称为软件包推荐。

安装新软件包时，默认情况下仍会安装推荐的软件包。更新现有软件包时，不会自动安装缺少的推荐软件包。要更改此设置，请在 `/etc/sysconfig/yast2` 中设置 `PKGMR_RECOMMENDED="yes"`。要安装已安装软件包的所有缺失推荐组件，请启动 YaST > 软件管理器，然后选择额外 > 安装所有匹配的推荐软件包。

要在安装新软件包时不安装推荐的软件包，请在 YaST 软件管理器中停用依赖项 > 安装推荐的软件包。如果使用命令行工具 Zypper 来安装软件包，请使用选项 `--no-recommends`。

## 8.4 管理软件储存库和服务

要安装第三方软件，请向系统添加软件储存库。默认情况下，系统注册时会自动配置产品储存库（例如 SUSE Linux Enterprise Desktop-DVD 15 SP5）和匹配的更新储存库。有关注册的更多信息，请参见《Deployment Guide》，第 4 章“Installation steps”，第 4.6 节“Registration”或《Upgrade Guide》，第 4 章“Upgrading offline”，第 4.7 节“Registering your system”。根据最初所选的产品，可能还会配置包含翻译、词典等的附加储存库。

要管理储存库，请启动 YaST，并选择软件 > 软件储存库。将打开已配置软件储存库对话框。在此，还可以通过将对话框右角的视图更改为所有服务来管理服务订阅。此环境中的服务是一种储存库索引服务 (RIS)，它能提供一个或多个软件储存库。此类服务可通过其管理员或供应商手动更改。

每个储存库都提供描述储存库内容（软件包名称、版本等）的文件。YaST 将这些储存库描述文件下载到本地缓存中。为了确保完整性，可使用储存库维护人员的 GPG 密钥来为软件储存库签名。每当您添加新的储存库时，YaST 都可让您导入其密钥。



### 警告：信任外部软件源

将外部软件储存库添加到您的储存库列表前，请先确保此储存库可信。对于因安装的来自第三方软件储存库中的软件引起的任何问题，SUSE 不承担任何责任。

#### 8.4.1 添加软件储存库

可以从 DVD/CD、USB 闪存盘、本地目录、ISO 映像或网络源添加储存库。

要通过 YaST 的已配置的软件储存库对话框添加储存库，请执行以下步骤：

1. 单击添加。
2. 选择对话框中列出的选项之一：



图 8.2：添加软件储存库

- 要通过 SLP 在网络中扫描宣告了其服务的安装服务器，请选择使用 SLP 扫描，然后单击下一步。
- 要从可卸媒体添加安装源，请选择相关选项，并在计算机中插入媒体或连接 USB 设备。单击下一步开始安装。
- 对于大多数储存库，在选择相应的选项并单击下一步后，系统会提示您指定媒体的路径（或 URL）。可以选择性地指定储存库名称。如果不指定任何储存库名称，YaST 将使用产品的名称或 URL 作为储存库名称。

默认情况下已激活下载储存库说明文件选项。如果停用该选项，YaST 稍后会根据需要自动下载这些文件。

- 根据您添加的储存库，系统可能会提示您导入储存库的 GPG 密钥，或者要求您同意许可条款。  
您确认后，YaST 即会下载并分析元数据。它会将储存库添加到已配置储存库列表。
- 如果需要，请根据第 8.4.2 节 “管理储存库属性” 中所述调整储存库的属性。
- 单击确定确认所做的更改，并关闭配置对话框。

6. 成功添加储存库之后，会启动软件管理器，您可以从此储存库安装软件包。有关细节，请参见[第 8 章 “安装或去除软件”](#)。

## 8.4.2 管理储存库属性

通过软件储存库的已配置的软件储存库概述，可以更改以下储存库属性：

### 状态

储存库状态可以为已启用或已禁用。您只能从已启用的储存库安装软件包。要暂时关闭某个储存库，请将它选中并停用启用。您还可以双击某个储存库名称来切换其状态。要彻底去除某个储存库，请单击删除。

### 刷新

刷新储存库时，会将其内容描述（软件包名称、版本等）下载到 YaST 使用的本地缓存中。对于诸如 CD 或 DVD 之类的静态储存库，该操作执行一次就已足够，而对于内容经常改变的储存库，应经常进行刷新。要使储存库的缓存保持最新，最简单的方式就是选择自动刷新。要执行手动刷新，请单击刷新并选择一个选项。

### 保留已下载的软件包

安装前从远程储存库下载软件包。默认情况下，这些软件包安装成功后即会删除。激活保留已下载的软件包可防止删除已下载的软件包。下载位置在 `/etc/zypp/zypp.conf` 中配置，默認為 `/var/cache/zypp/packages`。

### 优先级

储存库优先级是一个介于 `1` 和 `200` 之间的值，`1` 是最高优先级，`200` 是最低优先级。默认情况下，通过 YaST 添加的任何新储存库的优先级都是 `99`。如果您不在意某个储存库的优先级值，还可以将值设置为 `0`，表示对该储存库应用默认优先级 (`99`)。如果有多个储存库都提供了某个软件包，那么将优先使用优先级最高的储存库。因此，要避免从互联网下载不需要的软件包，可以为本地储存库（如 DVD）指定较高优先级。



### 重要：优先级与版本的对比

在任何情况下，优先级最高的储存库都是优先的。因此，请确保更新储存库总是具有最高优先级，否则您可能会安装过时的版本，直到下一次联机更新时才会更新该版本。

## 名称和 URL

要更改储存库名称或其 URL，可以从列表中单击选择它，然后单击编辑。

### 8.4.3 管理储存库密钥

为了确保完整性，可使用储存库维护人员的 GPG 密钥来为软件储存库签名。每当您要添加新的储存库时，YaST 都可让您导入其密钥。像校验任何其他 GPG 密钥一样对它进行校验，并确保它未更改。如果检测到密钥更改，则说明储存库可能存在错误。禁用它作为储存库，直到知道密钥更改原因为止。

要管理所有导入的密钥，请在已配置的软件储存库对话框中单击 GPG 密钥。用鼠标选择一个项以在窗口底部显示密钥属性。单击相应按钮可添加、编辑或删除密钥。

## 8.5 GNOME Package Updater

SUSE 持续为您的产品提供软件安全补丁和更新。可以使用连同桌面一起提供的工具或者运行 YaST 联机更新模块来安装这些补丁和更新。本节介绍如何使用 Package Updater 通过 GNOME 桌面更新系统。

与 YaST Online Update 模块相比，GNOME Package Updater 不仅可以从更新储存库安装补丁，而且还能安装已安装的软件包的新版本。（补丁可以修复安全问题或功能异常；功能和版本号通常不会更改。新版软件包的版本号会增大，并且软件包中通常会增加功能，或者引入重大更改。）

每当有新的补丁或软件包更新可用时，GNOME 就会在通知区域或锁定屏幕中显示通知。

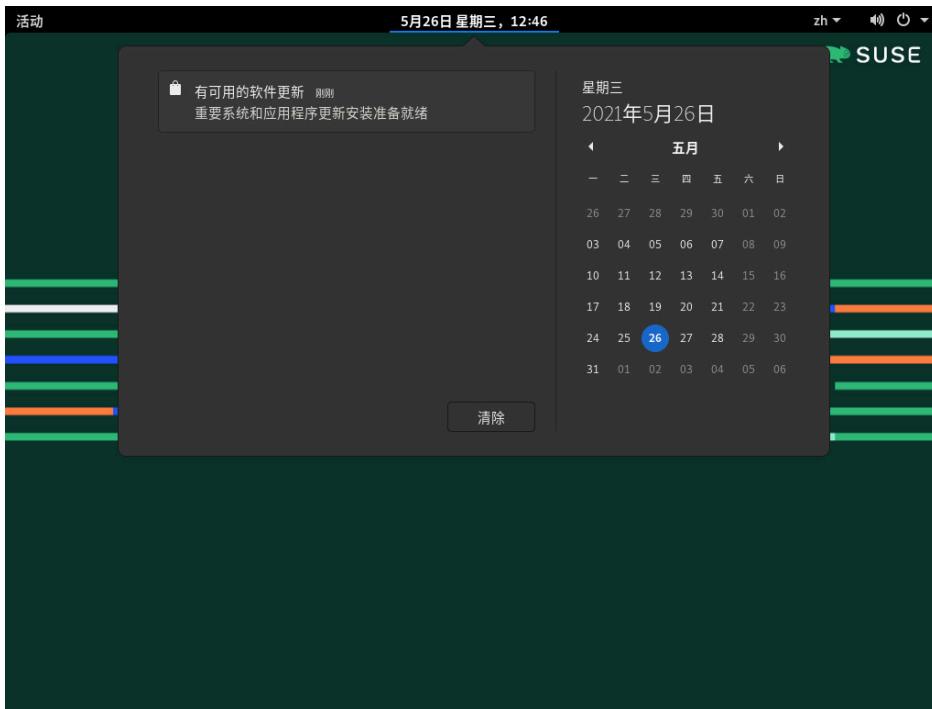
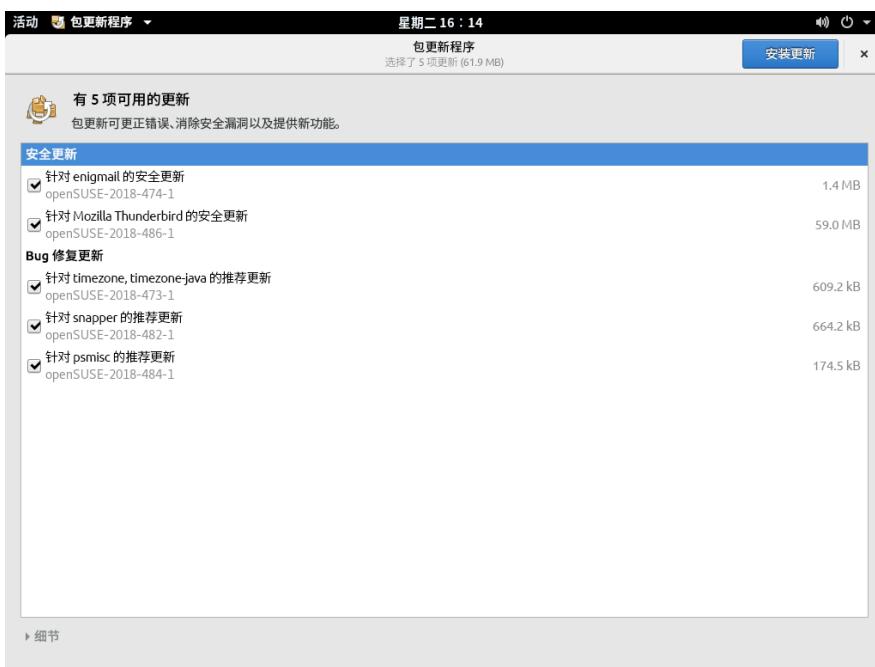


图 8.3：GNOME 桌面上的更新通知

要配置 Package Updater 的通知设置，请启动 GNOME 设置，并选择通知 > Package Updater。

#### 过程 8.2：使用 GNOME PACKAGE UPDATER 安装补丁和更新

1. 要安装补丁和更新，请单击通知消息。此时会打开 GNOME Package Updater。或者，在活动中键入 package 并选择 Package Updater，以打开更新程序。



## 2. 更新已按四个类别排序：

### 安全性更新（补丁）

修复严重的安全性危害，请务必安装。

### 建议的更新（补丁）

修复可能危及计算机安全的问题。强烈建议安装此类更新。

### 可选更新（补丁）

修复非安全性相关的问题或提供增强功能。

### 其他更新

已安装的包的新版本。

系统已预先选择安装所有可用更新。如果您不想要安装所有更新，请先取消选择不需要的更新。强烈建议始终安装所有安全更新和建议的更新。

要查看某个更新的详细信息，请单击其标题，然后单击 **细节**。信息将显示在软件包列表下面的框中。

## 3. 单击安装更新以开始安装。

## 4. 某些更新可能需要重启动计算机或注销。检查安装后显示的消息，以获取相关说明。

## 8.6 使用 GNOME Software 更新软件包

除 GNOME Package Updater 以外，GNOME 还提供具有以下功能的 GNOME Software：

- 安装、更新和去除通过 PackageKit 以 RPM 形式交付的软件
- 安装、更新和去除以 Flatpak 形式交付的软件
- 安装、更新和去除 GNOME 外壳扩展 (<https://extensions.gnome.org>)
- 使用 **Linux 供应商固件服务** (LVFS，[请参见 https://fwupd.org](https://fwupd.org)) 更新硬件设备的固件

除此之外，GNOME Software 还提供软件的屏幕快照、评级和评论。

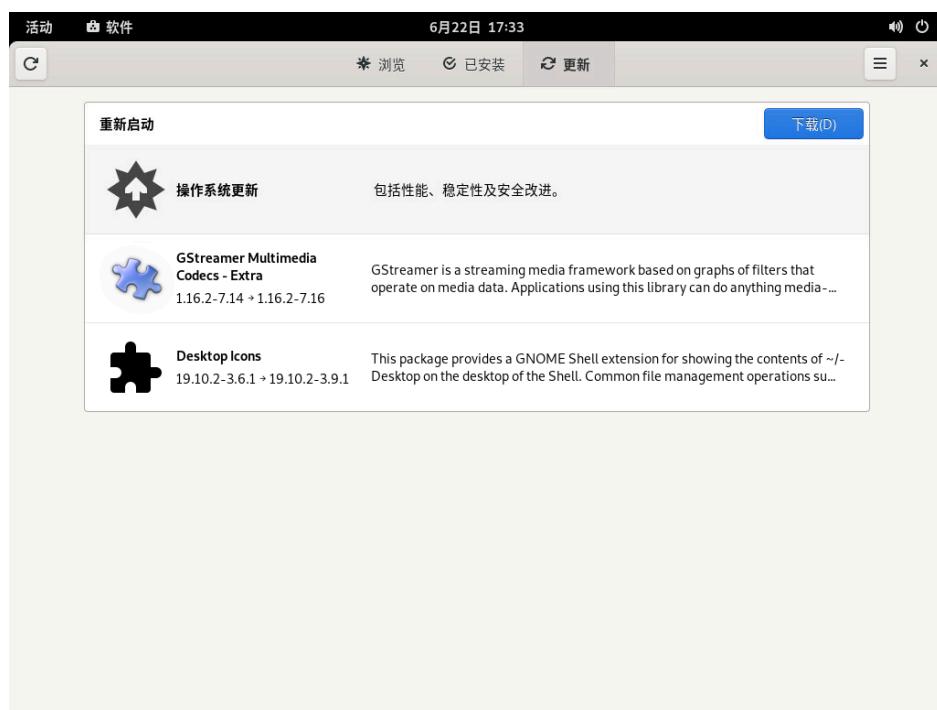


图 8.4：GNOME SOFTWARE — 更新视图

与 SUSE Linux Enterprise Desktop 中提供的其他工具相比，GNOME Software 存在以下差别：

- YaST 或 Zypper 可以安装打包为 RPM 的软件，而 GNOME Software 则与此不同，它仅限于安装可提供 AppStream 元数据的软件。这包括大部分桌面应用程序。
- GNOME Package Updater 会更新正在运行的系统中的软件包（这会强制您重启动相关的应用程序），而 GNOME Software 则是下载更新并在重引导后应用这些更新。

# 9 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。有关此环境中所用术语的定义（例如，repository、patch 或 update），请参见第 8.1 节“术语定义”。

## 9.1 使用 Zypper

Zypper 是用于安装、更新和去除软件包的命令行软件包管理器。它还可管理储存库。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。

### 9.1.1 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。有关常规选项和所有命令的列表，请参见 zypper help。要获取有关特定命令的帮助，请键入 zypper help COMMAND。

#### Zypper 命令

执行 Zypper 最简单的方式是，键入其名称后跟一个命令。例如，要将所有需要的补丁应用于系统，请使用：

```
> sudo zypper patch
```

#### 全局选项

此外，您还可以选择使用一个或多个全局选项，只需在命令前面键入它们即可：

```
> sudo zypper --non-interactive patch
```

在上面的示例中，选项 --non-interactive 表示在不询问任何问题的情况下运行命令（自动应用默认答复）。

#### 命令特定的选项

要使用特定于某个命令的选项，请紧接在该命令后面键入这些选项：

```
> sudo zypper patch --auto-agree-with-licenses
```

在上面的示例中，--auto-agree-with-licenses 用于将所有需要的补丁应用于系统，不要求您确认任何许可条款，而是自动接受许可条款。

## 参数

某些命令需要一个或多个参数。例如，使用 install 命令时，需要指定您要安装的一个或多个软件包：

```
> sudo zypper install mplayer
```

某些选项还需要单个参数。用以下命令可列出所有已知软件集：

```
> zypper search -t pattern
```

您可以组合上述所有软件集。例如，下面的命令将在详细模式下安装 factory 储存库中的 mc 和 vim 软件包：

```
> sudo zypper -v install --from factory mc vim
```

--from 选项可在从指定储存库请求软件包时，使所有储存库都处于启用状态（以解决任何依赖性问题）。--repo 是 --from 的别名，您可以使用两者中的任何一个。

多数 Zypper 命令都有 dry-run 选项，它模拟给定的命令。它可用于测试。

```
> sudo zypper remove --dry-run MozillaFirefox
```

Zypper 支持 --userdata STRING 全局选项。您可以使用此选项指定一个将会写入 Zypper 的日志文件和插件（例如 Btrfs 插件）的字符串。它可用于标记和标识日志文件中的事务。

```
> sudo zypper --userdata STRING patch
```

## 9.1.2 使用 Zypper 子命令

Zypper 子命令是存储在 zypper\_execdir 配置选项所指定的目录中的可执行文件。默认目录为 /usr/lib/zypper/commands。如果在该位置找不到某个子命令，Zypper 会自动在其余 \$PATH 位置搜索该子命令。这样，您便可以创建自己的本地扩展并将其存储在用户空间中。

不支持在 Zypper 外壳中执行子命令和使用全局 Zypper 选项。

列出可用的子命令：

```
> zypper help subcommand
[...]
Available zypper subcommands in '/usr/lib/zypper/commands'

appstream-cache
lifecycle
migration
search-packages

Zypper subcommands available from elsewhere on your $PATH

log           Zypper logfile reader
              (/usr/sbin/zypper-log)
```

查看子命令的帮助屏幕：

```
> zypper help appstream-cache
```

### 9.1.3 使用 Zypper 安装和去除软件

要安装或去除软件包，请使用以下命令：

```
> sudo zypper install PACKAGE_NAME
> sudo zypper remove PACKAGE_NAME
```



#### 警告：不要去除必需的系统软件包

不要去除必需的系统软件包，例如 `glibc`、`zypper`、`kernel`。如果去除这些包，系统可能会变得不稳定，或完全停止工作。

#### 9.1.3.1 选择要安装或去除的软件包

使用 `zypper install` 和 `zypper remove` 命令时，可以通过多种方法来找到软件包。

## 按确切的软件包名称

```
> sudo zypper install MozillaFirefox
```

## 按确切的软件包名称和版本号

```
> sudo zypper install MozillaFirefox-52.2
```

## 按储存库别名和软件包名称

```
> sudo zypper install mozilla:MozillaFirefox
```

其中 mozilla 是用于安装的储存库别名。

## 按使用通配符的软件包名称

您可以选择名称以特定字符串开头或结尾的所有软件包。使用通配符要小心，特别是去除软件包的时候。以下命令将安装名称以“Moz”开头的所有软件包：

```
> sudo zypper install 'Moz*'
```



## 提示：去除所有 -debuginfo 软件包

在调试问题时，您有时需要临时安装大量的 -debuginfo 软件包，以获取有关正在运行的进程的详细信息。在调试会话完成后，如果您需要清理环境，请运行以下命令：

```
> sudo zypper remove '*-debuginfo'
```

## 按功能

例如，要安装不知道名称的软件包，这些功能就很有用。下面的命令将安装软件包 MozillaFirefox：

```
> sudo zypper install firefox
```

## 按功能、硬件体系结构或版本

可以结合功能指定硬件体系结构和版本：

- 所需硬件体系结构的名称需要追加在功能的后面，两者以句点分隔。例如，要指定 AMD/Intel 64 体系结构（在 Zypper 中命名为 `x86_64_64`），请使用：

```
> sudo zypper install 'firefox.x86_64'
```

- 版本必须追加到字符串的末尾，并且前面必须带有一个运算符：`<`（小于）、`<=`（小于等于）、`=`（等于）、`>=`（大于等于）或`>`（大于）。

```
> sudo zypper install 'firefox>=74.2'
```

- 还可以指定硬件体系结构与版本组合要求：

```
> sudo zypper install 'firefox.x86_64>=74.2'
```

## 按 RPM 文件的路径

您还可以指定软件包的本地或远程路径：

```
> sudo zypper install /tmp/install/MozillaFirefox.rpm  
> sudo zypper install http://download.example.com/MozillaFirefox.rpm
```

### 9.1.3.2 同时安装和去除软件包

要同时安装和去除软件包，请使用`+-`修饰符。要安装`emacs`并同时去除`vim`，请使用：

```
> sudo zypper install emacs -vim
```

要去除`emacs`并同时安装`vim`，请使用：

```
> sudo zypper remove emacs +vim
```

为避免以`-`开头的软件包名称被解释为命令行选项，应始终将此类名称用作第二个参数。如果做不到这一点，请在名称前面加上`--`：

```
> sudo zypper install -emacs +vim      # Wrong  
> sudo zypper install vim -emacs        # Correct  
> sudo zypper install -- -emacs +vim   # Correct
```

```
> sudo zypper remove emacs +vim           # Correct
```

### 9.1.3.3 清理已去除软件包的依赖项

如果想让系统在去除指定的软件包后自动去除不再需要的所有软件包，请使用 --clean-deps 选项：

```
> sudo zypper rm --clean-deps PACKAGE_NAME
```

### 9.1.3.4 在脚本中使用 Zypper

默认情况下，在安装或删除选定软件包之前或发生问题时，Zypper 会要求确认。您可以使用 --non-interactive 选项覆盖此行为。必须在实际命令（install、remove 和 patch）的前面指定此选项，如下所示：

```
> sudo zypper --non-interactive install PACKAGE_NAME
```

该选项允许在脚本和 cron 任务中使用 Zypper。

### 9.1.3.5 安装或下载源软件包

要安装某个软件包的对应源软件包，请使用：

```
> zypper source-install PACKAGE_NAME
```

以 root 身份执行时，源软件包的默认安装位置为 /usr/src/packages/；以用户身份运行时，则为 ~/rpmbuild。可以在本地 rpm 配置中更改这些值。

使用此命令还会安装指定软件包的版本依赖项。如果不想执行此操作，请添加开关 -D：

```
> sudo zypper source-install -D PACKAGE_NAME
```

要只安装版本依赖项，请使用 -d。

```
> sudo zypper source-install -d PACKAGE_NAME
```

当然，只有当储存库列表中启用了含有源软件包的储存库时，才能这样做（默认添加但不启用它）。请参见第 9.1.6 节 “使用 Zypper 管理储存库” 了解有关储存库管理的细节。

可使用以下方法来获取储存库中所有源软件包的列表：

```
> zypper search -t srcpackage
```

您也可以将所有已安装软件包的源软件包下载到本地目录。要下载源软件包，请使用：

```
> zypper source-download
```

默认的下载目录是 `/var/cache/zypper/source-download`。您可以使用 `--directory` 选项更改下载目录。若只想显示缺失或多余的软件包，而不下载或删除任何内容，请使用 `--status` 选项。要删除多余的源软件包，请使用 `--delete` 选项。要禁止删除，请使用 `--no-delete` 选项。

### 9.1.3.6 从禁用的储存库安装软件包

通常，您只能安装或刷新来自启用的储存库的软件包。`--plus-content TAG` 选项可帮助您指定要刷新的、要在当前 Zypper 会话期间暂时启用的，以及要在会话完成后禁用的储存库。例如，要启用可以提供其他 `-debuginfo` 或 `-debugsource` 软件包的储存库，请使用 `--plus-content debug`。可以多次指定此选项。

要暂时启用此类“调试”储存库以安装特定的 `-debuginfo` 软件包，请按如下所示使用该选项：

```
> sudo zypper --plus-content debug \
  install "debuginfo(build-id)=eb844a5c20c70a59fc693cd1061f851fb7d046f4"
```

对于缺少的 `debuginfo` 软件包，`gdb` 将会报告 `build-id` 字符串。



#### 注意：已禁用安装媒体

仍会配置 SUSE Linux Enterprise Desktop 安装媒体中的储存库，但成功安装后会禁用这些储存库。您可以使用 `--plus-content` 选项从安装媒体而不是联机储存库安装软件包。在调用 `zypper` 之前，请确保该媒体可用，例如，将 DVD 插入计算机的光驱中。

### 9.1.3.7 实用程序

要校验所有依赖项是否仍然满足，并修复缺少的依赖项，请使用：

```
> zypper verify
```

除了依赖项必须满足外，某些软件包还“推荐”其他软件包。只有在实际可用并可安装时才会安装这些推荐软件包。如果推荐的软件包在推荐它们的软件包已安装（通过添加其他软件包或硬件）之后才可用，请使用以下命令：

```
> sudo zypper install-new-recommends
```

此命令在插入网络摄像头或 Wi-Fi 设备后非常有用。如果可用，它将安装设备驱动程序和相关软件。只有在满足特定硬件依赖项后，才可安装驱动程序和相关软件。

## 9.1.4 使用 Zypper 更新软件

用 Zypper 更新软件有三种方式：安装软件包、安装软件包的新版本或更新整个分发包。最后一种方式可通过 **zypper dist-upgrade** 来实现。中介绍了如何升级 SUSE Linux Enterprise Desktop 《Upgrade Guide》，第 2 章“Upgrade paths and methods”。

### 9.1.4.1 安装全部所需的补丁

要安装已安装软件包的新版本，最可靠的方式是**修补** SUSE Linux Enterprise。这种方式可保证安装版本正确的所有必需软件包，并确保忽略被视为**冲突**的软件包版本。

要安装所有适用于您系统的正式发布的补丁，请运行：

```
> sudo zypper patch
```

系统将会检查您计算机上配置的储存库中提供的所有补丁是否与您的安装相关。如果相关（未分为 **optional** 或 **feature** 类别），则会立即安装这些补丁。如果 **zypper patch** 成功，便可确保不会安装带漏洞版本的软件包，除非您确认该例外。请注意，正式的更新储存库仅在注册 SUSE Linux Enterprise Desktop 安装后才可用。

如果即将安装的补丁所包含的更改要求重引导系统，您会在重引导前收到警告。

单纯使用 **zypper patch** 命令不会应用来自第三方储存库的软件包。要同时更新第三方储存库，请使用 with-update 命令选项，如下所示：

```
> sudo zypper patch --with-update
```

要额外安装可选补丁，请使用：

```
> sudo zypper patch --with-optional
```

要安装与特定 Bugzilla 问题相关的所有补丁，请使用：

```
> sudo zypper patch --bugzilla=NUMBER
```

要安装与特定 CVE 数据库项相关的所有补丁，请使用：

```
> sudo zypper patch --cve=NUMBER
```

例如，要安装 CVE 编号为 CVE-2010-2713 的安全补丁，请执行：

```
> sudo zypper patch --cve=CVE-2010-2713
```

如果只想安装影响 Zypper 和软件包管理本身的补丁，请使用：

```
> sudo zypper patch --updatestack-only
```

请记住，如果您使用了 updatestack-only 命令选项，将会丢弃原本还会更新其他储存库的其他命令选项。

### 9.1.4.2 列出补丁

为了让您确定补丁是否可用，Zypper 允许您查看以下信息：

#### 所需补丁的数目

要列出所需补丁（适用于您的系统但尚未安装的补丁）的数目，请使用 patch-check：

```
> zypper patch-check
Loading repository data...
Reading installed packages...
5 patches needed (1 security patch)
```

可以结合 `--updatestack-only` 选项使用此命令，以便仅列出影响 Zypper 和软件包管理本身的补丁。

## 所需补丁的列表

要列出全部所需的补丁（适用于您的系统但尚未安装的补丁），请使用 `zypper list-patches`。

## 所有补丁的列表

要列出 SUSE Linux Enterprise Desktop 可用的所有补丁，而不管它们是否已安装或适用于您的安装，请使用 `zypper patches`。

还可以列出并安装与特定问题相关的补丁。要列出特定的补丁，请使用带以下选项的 `zypper list-patches` 命令：

### 按 Bugzilla 问题

要列出与 Bugzilla 问题相关的全部所需补丁，请使用 `--bugzilla` 选项。

要列出针对特定 Bug 的补丁，您也可以指定 Bug 编号：`--bugzilla=NUMBER`。要搜索与多个 Bugzilla 问题相关的补丁，请在 bug 编号之间添加逗号，例如：

```
> zypper list-patches --bugzilla=972197,956917
```

### 按 CVE 编号

要列出与 CVE（公共漏洞和披露）数据库中某一项相关的全部所需补丁，请使用 `--cve` 选项。

要列出针对特定 CVE 数据库项的补丁，您也可以指定 CVE 编号：`--cve=NUMBER`。要搜索与多个 CVE 数据库项相关的补丁，请在 CVE 编号之间添加逗号，例如：

```
> zypper list-patches --cve=CVE-2016-2315,CVE-2016-2324
```

## 列出已撤回的补丁

在 SUSE Linux Enterprise 15 代码流中，一些补丁可自动撤回。维护更新会经过仔细测试，因为更新有可能包含新的 Bug。如果发现某个补丁包含 Bug，将会发布一个新的更新（版本号更高）以还原有 Bug 的更新，并会阻止再次安装有 Bug 的更新。您可以使用 `zypper` 列出已撤回的补丁：

```
> zypper lp --all |grep retracted
SLE-Module-Basesystem15-SP3-Updates | SUSE-SLE-Module-Basesystem-15-
SP3-2021-1965
```

```
| recommended | important | --- | retracted | Recommended update for
multipath-tools
SLE-Module-Basesystem15-SP3-Updates | SUSE-SLE-Module-Basesystem-15-
SP3-2021-2689
| security | important | --- | retracted | Security update for cpio
SLE-Module-Basesystem15-SP3-Updates | SUSE-SLE-Module-Basesystem-15-
SP3-2021-3655
| security | important | reboot | retracted | Security update for the
Linux Kernel
```

查看有关已撤回（或任何）补丁的完整信息：

```
> zypper patch-info SUSE-SLE-Product-SLES-15-2021-2689
Loading repository data...
Reading installed packages...

Information for patch SUSE-SLE-Product-SLES-15-2021-2689:
-----
Repository : SLE-Product-SLES15-LTSS-Updates
Name       : SUSE-SLE-Product-SLES-15-2021-2689
Version    : 1
Arch       : noarch
Vendor     : maint-coord@suse.de
Status     : retracted
Category   : security
Severity   : important
Created On : Mon 16 Aug 2021 03:44:00 AM PDT
Interactive : ---
Summary    : Security update for cpio
Description :
    This update for cpio fixes the following issues:

    It was possible to trigger Remote code execution due to a integer
    overflow
    (CVE-2021-38185, bsc#1189206)

    UPDATE:
    This update was buggy and could lead to hangs, so it has been
    retracted.
```

```
There will be a follow up update.
```

```
[...]
```

## 具有冲突软件包的补丁

```
Information for patch openSUSE-SLE-15.3-2022-333:  
-----  
Repository : Update repository with updates from SUSE Linux Enterprise 15  
Name       : openSUSE-SLE-15.3-2022-333  
Version    : 1  
Arch       : noarch  
Vendor     : maint-coord@suse.de  
Status     : needed  
Category   : security  
Severity   : important  
Created On : Fri Feb  4 09:30:32 2022  
Interactive: reboot  
Summary    : Security update for xen  
Description :  
    This update for xen fixes the following issues:  
  
    - CVE-2022-23033: Fixed guest_physmap_remove_page not removing the p2m  
    mappings. (XSA-393) (bsc#1194576)  
    - CVE-2022-23034: Fixed possible DoS by a PV guest Xen while unmapping  
    a grant. (XSA-394) (bsc#1194581)  
    - CVE-2022-23035: Fixed insufficient cleanup of passed-through device  
    IRQs. (XSA-395) (bsc#1194588)  
Provides   : patch:openSUSE-SLE-15.3-2022-333 = 1  
Conflicts  : [22]  
    xen.src < 4.14.3_06-150300.3.18.2  
    xen.noarch < 4.14.3_06-150300.3.18.2  
    xen.x86_64 < 4.14.3_06-150300.3.18.2  
    xen-devel.x86_64 < 4.14.3_06-150300.3.18.2  
    xen-devel.noarch < 4.14.3_06-150300.3.18.2  
[...]
```

上面的补丁与 22 个软件包的受影响或带漏洞版本冲突。如果安装了这些受影响或带漏洞软件包中的任何一个，将会触发冲突，并且补丁会划分为**需要**类别。**zypper patch** 会尝试安装所有可用的补丁。如果遇到问题，它会报告这些问题，以告知您并非所有更新都已

安装。可以通过更新受影响或带漏洞软件包或者去除这些软件包来解决冲突。由于 SUSE 更新储存库也提供固定的软件包，因此更新是解决冲突的标准方法。例如，如果由于依赖关系问题或软件包锁定而无法更新软件包，则会在用户批准后将其删除。

要列出所有补丁而不考虑是否需要安装它们，另外还需使用 `--all` 选项。例如，要列出指派有 CVE 编号的所有补丁，请使用：

```
> zypper list-patches --all --cve
Issue | No.           | Patch           | Category | Severity | Status
-----+-----+-----+-----+-----+-----+
cve   | CVE-2019-0287 | SUSE-SLE-Module.. | recommended | moderate | needed
cve   | CVE-2019-3566 | SUSE-SLE-SERVER.. | recommended | moderate | not needed
[...]
```

### 9.1.4.3 安装新的软件包版本

如果某个安装源只包含新软件包，但未提供补丁，则 `zypper patch` 不会产生任何作用。要将所有已安装的软件包更新为较新的可用版本，请使用以下命令：

```
> sudo zypper update
```

## ! 重要

`zypper update` 会忽略有问题的软件包。例如，如果某个软件包被锁定，即使该软件包有更高的版本可用，`zypper update` 也会忽略该软件包。相反，如果软件包被视为带漏洞，`zypper patch` 会报告冲突。

要更新个别软件包，请用更新或安装命令指定软件包：

```
> sudo zypper update PACKAGE_NAME
> sudo zypper install PACKAGE_NAME
```

可使用此命令来获取所有新的可安装软件包的列表：

```
> zypper list-updates
```

请注意，此命令只会列出符合以下准则的软件包：

- 与已安装的软件包拥有相同的供应商，
- 由至少与已安装软件包拥有相同优先级的储存库提供，
- 可安装（满足所有依赖项）。

所有新的可用软件包（无论是否可安装）的列表可通过以下方式获取：

```
> sudo zypper list-updates --all
```

要确定无法安装新软件包的原因，请使用上面所述的 [zypper install](#) 或 [zypper update](#) 命令。

#### 9.1.4.4 识别孤立的软件包

每当您从 Zypper 中去除某个储存库或者升级系统时，某些软件包可能会进入“孤立”状态。这些孤立的软件包不再属于任何活动储存库。以下命令可以列出这些软件包：

```
> sudo zypper packages --orphaned
```

借助此列表，您可以确定是否仍然需要某个软件包，或者是否可以安全去除某个软件包。

#### 9.1.5 识别使用已删除文件的进程和服务

在修补、更新或去除软件包时，系统上可能有一些正在运行的进程仍在使用已被更新或去除操作删除的文件。运行 [zypper ps](#) 可以列出使用已删除文件的进程。如果此类进程属于某个已知的服务，则会列出服务名称，方便您重启该服务。默认情况下，[zypper ps](#) 会显示一个表：

```
> zypper ps
PID  | PPID | UID | User | Command      | Service      | Files
-----+-----+-----+-----+-----+-----+-----+
814  | 1   | 481 | avahi | avahi-daemon | avahi-daemon | /lib64/ld-2.19.s->
     |     |     |     |           |           | /lib64/libdl-2.1->
     |     |     |     |           |           | /lib64/libpthread->
     |     |     |     |           |           | /lib64/libc-2.19->
[...]
```

**PID**：进程 ID

**PPID**: 父进程 ID

**UID**: 运行进程的用户的 ID

**登录**: 运行进程的用户的登录名

**命令**: 用于执行进程的命令

**服务**: 服务名称 (仅当命令与系统服务关联时)

**文件**: 已删除文件的列表

通过如下方式可控制 **zypper ps** 的输出格式:

### **zypper ps -s**

创建一份简短表格，其中不会显示已删除的文件。

```
> zypper ps -s
PID | PPID | UID | User | Command | Service
-----+-----+-----+-----+-----+-----+
814 | 1 | 481 | avahi | avahi-daemon | avahi-daemon
817 | 1 | 0 | root | irqbalance | irqbalance
1567 | 1 | 0 | root | sshd | sshd
1761 | 1 | 0 | root | master | postfix
1764 | 1761 | 51 | postfix | pickup | postfix
1765 | 1761 | 51 | postfix | qmgr | postfix
2031 | 2027 | 1000 | tux | bash |
```

### **zypper ps -ss**

仅显示与系统服务关联的进程。

```
PID | PPID | UID | User | Command | Service
-----+-----+-----+-----+-----+-----+
814 | 1 | 481 | avahi | avahi-daemon | avahi-daemon
817 | 1 | 0 | root | irqbalance | irqbalance
1567 | 1 | 0 | root | sshd | sshd
1761 | 1 | 0 | root | master | postfix
1764 | 1761 | 51 | postfix | pickup | postfix
1765 | 1761 | 51 | postfix | qmgr | postfix
```

### **zypper ps -sss**

仅显示使用已删除文件的系统服务。

```
avahi-daemon
irqbalance
```

```
postfix
sshd
```

**zypper ps--print "systemctl status %s"**

显示用于检索可能需要重启动的服务状态信息的命令。

```
systemctl status avahi-daemon
systemctl status irqbalance
systemctl status postfix
systemctl status sshd
```

有关服务处理的详细信息，请参见[第 19 章 “systemd 守护程序”](#)。

### 9.1.6 使用 Zypper 管理储存库

Zypper 的所有安装或补丁命令均基于已知储存库列表。要列出系统已知的所有储存库，请使用命令：

```
> zypper repos
```

结果将类似于与以下输出：

#### 例 9.1：ZYPPER — 已知储存库的列表

```
> zypper repos
# | Alias          | Name          | Enabled | Refresh
--+
1 | SLEHA-15-GE0 | SLEHA-15-GE0 | Yes     | No
2 | SLEHA-15     | SLEHA-15     | Yes     | No
3 | SLES15       | SLES15       | Yes     | No
```

当在各个命令中指定储存库时，可以使用别名、URI 或 **zypper repos** 命令输出中的储存库编号。储存库别名是用于储存库处理命令中的储存库名称的简短版本。请注意，在修改储存库列表后，储存库编号可能会更改。别名本身不会更改。

默认情况下不显示储存库的 URI 或优先级之类的细节。用以下命令可以列出所有细节：

```
> zypper repos -d
```

### 9.1.6.1 添加储存库

要添加储存库，请运行

```
> sudo zypper addrepo URI ALIAS
```

URI 可以是互联网储存库、网络资源、目录、CD 或 DVD（有关细节请参见 [https://en.opensuse.org/openSUSE:Libzypp\\_URLs](https://en.opensuse.org/openSUSE:Libzypp_URLs)）。ALIAS 是储存库的唯一简写标识符。您可以随意选择别名，前提是它必须唯一。如果指定的别名已在使用，Zypper 将发出警告。

### 9.1.6.2 刷新储存库

zypper 可让您从配置的储存库中提取软件包的更改。要提取更改，请运行：

```
> sudo zypper refresh
```



#### 注意：zypper 的默认行为

有些命令默认会自动执行 refresh，因此您不需要明确运行该命令。

将 refresh 命令与 --plus-content 选项搭配使用时，还可查看已禁用储存库中的更改：

```
> sudo zypper --plus-content refresh
```

该选项虽然会提取储存库中的更改，但会使禁用储存库的状态保持不变，即仍为禁用。

### 9.1.6.3 去除储存库

要从列表中去除某个储存库，请将命令 zypper removerepo 与要删除的储存库的别名或编号结合使用。例如，要从 SLEHA-12-GE0 中去除储存库 [例 9.1 “Zypper — 已知储存库的列表”](#)，请使用下列命令之一：

```
> sudo zypper removerepo 1
```

```
> sudo zypper removerepo "SLEHA-12-GEO"
```

#### 9.1.6.4 修改储存库

使用 **zypper modifyrepo** 可以启用或禁用储存库。您还可以用该命令更改储存库的属性（例如刷新行为、名称或优先级）。以下命令将会启用名为 updates 的储存库、打开自动刷新并将其优先级设置为 20：

```
> sudo zypper modifyrepo -er -p 20 'updates'
```

修改储存库并不局限于单个储存库 — 您也可以对组执行该操作：

-a: 所有储存库

-l: 本地储存库

-t: 远程储存库

-m TYPE: 特定类型的储存库（其中 TYPE 可以是以下之

-: http、https、ftp、cd、dvd、dir、file、cifs、smb、nfs、hd、iso)

要重命名储存库别名，请使用 renamerepo 命令。以下示例将别名从 Mozilla Firefox 更改为 firefox：

```
> sudo zypper renamerepo 'Mozilla Firefox' firefox
```

#### 9.1.7 使用 Zypper 查询储存库和软件包

Zypper 提供各种查询储存库或软件包的方式。要获取所有可用的产品、软件集、软件包或补丁的列表，请使用以下命令：

```
> zypper products
> zypper patterns
> zypper packages
> zypper patches
```

要查询特定软件包的所有储存库，请使用 search。要获得有关特定软件包的信息，请使用 info 命令。

### 9.1.7.1 搜索软件

**zypper search** 命令可对软件包名或（视情况）对软件包摘要和说明执行搜索。用 / 括住的字符串会解译为正则表达式。默认情况下搜索不区分大小写。

执行简单搜索来查找包含 fire 的软件包名称

```
> zypper search "fire"
```

执行简单搜索来查找确切的软件包 MozillaFirefox

```
> zypper search --match-exact "MozillaFirefox"
```

同时在软件包描述和摘要中搜索

```
> zypper search -d fire
```

仅显示尚未安装的软件包

```
> zypper search -u fire
```

显示包含字符串 fir 且该字符串后面不是 e 的软件包

```
> zypper se "/fir[^e]/"
```

### 9.1.7.2 在所有 SLE 模块中搜索软件包

要搜索当前已启用的 SLE 模块内部和外部的软件包，请使用 search-packages 子命令。此命令会联系 SUSE Customer Center，并在所有模块中搜索匹配的软件包，例如：

```
> zypper search-packages package1 package2
```

**zypper search-packages** 提供以下选项：

- 搜索与您的搜索字符串完全匹配的软件包：-x、--match-exact
- 按模块对结果进行分组（默认：按软件包分组）：-g、--group-by-module

- 显示有关软件包的更多详细信息: [-d](#)、[--details](#)
- 以 XML 格式输出搜索结果: [--xmlout](#)

### 9.1.7.3 搜索特定功能

要搜索提供特殊功能的软件包, 请使用命令 [what-provides](#)。例如, 如果您想知道哪个软件包提供 Perl 模块 [SVN::Core](#), 请使用以下命令:

```
> zypper what-provides 'perl(SVN::Core)'
```

[what-provides PACKAGE\\_NAME](#) 与 [rpm -q --whatprovides PACKAGE\\_NAME](#) 类似, 不过 RPM 只能查询 RPM 数据库 (即所有已安装的包的数据库)。另一方面, Zypper 将告诉您任意储存库的功能的提供商, 而非仅已安装的储存库功能的提供商。

### 9.1.7.4 显示软件包信息

要查询个别软件包, 请使用 [info](#) 命令, 并用完整软件包名称作为参数。这会显示有关某个软件包的详细信息。如果软件包名与储存库中的所有软件包名都不匹配, 该命令会输出非软件包匹配项的详细信息。如果您请求特定类型 (通过使用 [-t](#) 选项), 但该类型不存在, 该命令会输出其他可用的匹配项, 但不提供详细信息。

如果您指定源软件包, 该命令会显示基于该源软件包构建的二进制软件包。如果您指定二进制软件包, 该命令会输出用来构建该二进制软件包的源软件包。

如果还想显示该软件包所需/推荐的软件包, 则使用选项 [--requires](#) 和 [--recommends](#):

```
> zypper info --requires MozillaFirefox
```

## 9.1.8 显示生命周期信息

SUSE 产品的支持周期一般为 10 年。通常, 您可以使用 SUSE 的延长支持产品将标准生命周期延长三年。根据您的产品, 在 <https://www.suse.com/lifecycle> 中找到具体的支持生命周期。

要检查您的产品和所支持软件包的生命周期，请如下所示使用 **zypper lifecycle** 命令：

```
# zypper lifecycle
  Product end of support
  Codestream: SUSE Linux Enterprise Server 15           2028-07-31
  Product: SUSE Linux Enterprise Server 15 SP3          n/a*
  Module end of support
  Basesystem Module                                     n/a*
  Desktop Applications Module                         n/a*
  Server Applications Module                         n/a*
  Package end of support if different from product:
  autoofs                                         Now, installed 5.1.3-7.3.1, update
  available 5.1.3-7.6.1
```

## 9.1.9 配置 Zypper

Zypper 现在随附配置文件，允许您永久更改 Zypper 的行为（系统范围或用户特定）。要进行系统范围更改，请编辑 `/etc/zypp/zypper.conf`。要进行用户相关的更改，请编辑 `~/.zypper.conf`。如果 `~/.zypper.conf` 尚不存在，您可以使用 `/etc/zypp/zypper.conf` 作为模板：将其复制到 `~/.zypper.conf` 并根据您的喜好进行调整。请参见文件中的注释，获取有关可用选项的帮助。

## 9.1.10 查错

如果您在访问配置的储存库中的软件包时遇到问题（例如，尽管您知道某个软件包在某个储存库中，但 Zypper 找不到该软件包），刷新储存库或许可以解决问题：

```
> sudo zypper refresh
```

如果不起作用，则尝试

```
> sudo zypper refresh -fdb
```

这会强制完全刷新和重构数据库，包括强制下载原始元数据。

## 9.1.11 Btrfs 文件系统上的 Zypper 回滚功能

如果根分区上使用的是 Btrfs 文件系统，且系统中安装了 **snapper**，当 Zypper 提交对文件系统所做的更改以创建相应的文件系统快照时，会自动调用 **snapper**。这些快照可用于还原 Zypper 进行的任何更改。有关更多信息，请参见第 10 章 “通过 Snapper 进行系统恢复和快照管理”。

## 9.1.12 更多信息

有关从命令行管理软件的详细信息，请输入 **zypper help**、**zypper help COMMAND**，或参见 **zypper(8)** 手册页。有关详尽的命令参考、最重要的命令的 **cheat sheets**，以及如何在脚本和应用程序中使用 Zypper 的信息，请参见 [https://en.opensuse.org/SDB:Zypper\\_usage](https://en.opensuse.org/SDB:Zypper_usage)。SUSE Linux Enterprise Desktop 最新版本的软件更改列表可在 [https://en.opensuse.org/openSUSE:Zypper\\_versions](https://en.opensuse.org/openSUSE:Zypper_versions) 中找到。

## 9.2 RPM - 软件包管理器

RPM (RPM 程序包管理器) 用于管理软件包。主要命令为 **rpm** 和 **rpmbuild**。用户、系统管理员和软件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。

**rpm** 有五种模式：安装、卸装（或更新）软件包、重构建 RPM 数据库、查询 RPM 库或独立 RPM 存档、对软件包执行完整性检查以及对软件包签名。**rpmbuild** 可用于从原始源构建可安装的软件包。

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存储在 RPM 数据库中进行存档。RPM 存档的扩展名通常为 **.rpm**。



### 提示：软件开发包

对于数个软件包，软件开发所需的组件（库、标题、包含文件等）已分别封装在不同的软件包中。只有当您要自己编译软件时才需要这些开发包（例如最新的 GNOME 软件包）。通过扩展名 **-devel** 即可识别这些开发包，例如软件包 **alsa-devel** 和 **gimp-devel**。

## 9.2.1 校验软件包真实性

RPM 软件包具有 GPG 签名。要校验 RPM 软件包的签名，请使用 `rpm --checksig PACKAGE-1.2.3.rpm` 命令确定该软件包是来自 SUSE 还是另一个可信机构。特别建议对来自互联网的更新软件包使用此命令。

修复操作系统中的问题时，您可能需要将问题临时修复 (PTF) 安装到生产系统中。SUSE 提供的软件包已使用特殊的 PTF 密钥签名。但是，与 SUSE Linux Enterprise 11 不同，SUSE Linux Enterprise 12 系统上默认不会导入此密钥。要手动导入该密钥，请使用以下命令：

```
> sudo rpm --import \
/usr/share/doc/packages/suse-build-key/suse_ptf_key.asc
```

导入该密钥后，您可以在系统上安装 PTF 软件包。

## 9.2.2 管理软件包：安装、更新和卸装

通常，安装 RPM 存档的过程非常简单，只需使用 `rpm -i PACKAGE.rpm`。使用此命令可以安装软件包，但前提是满足其依赖项并且不与其他软件包冲突。如果出现错误消息，`rpm` 将请求那些需要安装的软件包以满足依赖项要求。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软件包。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` 或 `--upgrade` 以及 `-F` 或 `--freshen` 可用于更新软件包（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软件包，而 `-F` 只更新以前安装的软件包。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。
- 如果更新前系统管理员曾更改过配置文件，则 `rpm` 会将更改的文件另存为扩展名为 `.rpmodig` 或 `.rpmsave` 的文件（备份文件），并安装新软件包中的版本。仅当原先安装的文件和较新的版本不同时，才执行此操作。如果是这种情况，则将备份文件（`.rpmodig` 或 `.rpmsave`）与新安装的文件进行比较，然后在新文件中再次进行更改。之后，请删除所有 `.rpmodig` 和 `.rpmsave` 文件，以免日后的更新出现问题。
- 如果配置文件已存在并且 `.spec` 文件中指定了 `noreplace` 标签，便会出现 `.rpmnew` 文件。

更新后，在对 `.rpmsave` 和 `.rpmnew` 文件进行比较后应将它们删除，以防其阻碍日后的更新。如果 RPM 数据库以前未能识别文件，则会为其指派扩展名 `.rpmodig`。

否则，将使用 `.rpmsave`。换句话说，`.rpmodig` 是从异系统格式更新为 RPM 的结果。`.rpmsave` 而 `.rpmnew` 是从旧版 RPM 更新到新版 RPM 产生。`.rpmnew` 不提供任何有关系统管理员是否对配置文件进行过任何更改的信息。`/var/adm/rpmconfigcheck` 中提供了这些文件的列表。更新时不会覆盖某些配置文件（如 `/etc/httpd/httpd.conf`），以便系统能够持续运行。

`-U` 开关的作用并不完全等同于使用 `-e` 选项进行卸装，以及使用 `-i` 选项进行安装，它还有其他作用。只要可能，就应使用 `-U`。

要去除软件包，请输入 `rpm -e PACKAGE`。仅当不存在未解决的依赖项问题时，此命令才会删除软件包。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。如果出于某种原因无法执行此类删除操作（即使不存在其他依赖性问题），或许可以使用选项 `--rebuilddb` 来重构 RPM 数据库。

### 9.2.3 增量 RPM 软件包

增量 RPM 软件包包含旧版本和新版本的 RPM 软件包之间的差别。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 也可以与已安装的 RPM 一起工作。增量 RPM 软件包的大小甚至比补丁 RPM 小，这有利于通过互联网传送更新软件包。缺点是，涉及增量 RPM 的更新操作与使用纯粹 RPM 或补丁 RPM 进行更新的情况相比，占用的 CPU 周期要长得多。

**makedeltarpm** 和 **applydeltarpm** 二进制文件是增量 RPM 套件（软件包 `deltarpm`）的一部分，可帮助您创建和应用增量 RPM 软件包。使用以下命令可以创建名为 `new.delta.rpm` 的增量 RPM。以下命令假设 `old.rpm` 和 `new.rpm` 都存在：

```
> sudo makedeltarpm old.rpm new.rpm new.delta.rpm
```

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

```
> sudo applydeltarpm new.delta.rpm new.rpm
```

要从旧 RPM 产生新 RPM 而不访问文件系统，请使用 `-r` 选项：

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

有关技术细节，请参见 `/usr/share/doc/packages/deltarpm/README`。

## 9.2.4 RPM查询

带 `-q` 选项的 **rpm** 会启动查询，如此用户便可检查 RPM 存档（通过添加选项 `-p`）并查询已安装软件包的 RPM 数据库。可以使用多个开关指定所需信息的类型。请参见表 9.1 “基本 RPM 查询选项”。

表 9.1：基本 RPM 查询选项

|                      |   |
|----------------------|---|
| <code>-i</code>      | 软件包信息   |
| <code>-l</code>      | 文件列表  |
| <code>-f FILE</code> | 查询包含文件 <code>FILE</code> 的软件包（必须使用 <code>FILE</code> 指定完整路径）              |
| <code>-s</code>      | 包含状态信息的文件列表（隐含 <code>-l</code> ）  |
| <code>-d</code>      | 仅列出文档文件（隐含 <code>-l</code> ）  |
| <code>-c</code>      | 仅列出配置文件（隐含 <code>-l</code> ）  |
| <code>--dump</code>  | 包含完整详细信息的文件列表（与 <code>-l</code> 、 <code>-c</code> 或 <code>-d</code> 结合使用） |

|                |  |
|----------------|--|
| --provides     | 列出另一个软件包可通过 <u>--requires</u> 请求的该软件包的功能 |
| --requires, -R | 软件包需要的功能                                 |
| --scripts      | 安装脚本（预安装、后安装、卸载）                         |

例如，命令 **rpm -q -i wget** 可显示例 9.2 “**rpm -q -i wget**” 中所示的信息。

#### 例 9.2：**rpm -q -i wget**

```
Name      : wget
Version   : 1.14
Release   : 17.1
Architecture: x86_64
Install Date: Mon 30 Jan 2017 14:01:29 CET
Group     : Productivity/Networking/Web/Utilities
Size      : 2046483
License   : GPL-3.0+
Signature : RSA/SHA256, Thu 08 Dec 2016 07:48:44 CET, Key ID 70af9e8139db7c82
Source RPM : wget-1.14-17.1.src.rpm
Build Date : Thu 08 Dec 2016 07:48:34 CET
Build Host : sheep09
Relocations : (not relocatable)
Packager   : https://www.suse.com/
Vendor     : SUSE LLC <https://www.suse.com/>
URL       : http://www.gnu.org/software/wget/
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.
Distribution: SUSE Linux Enterprise 15
```

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：

```
> rpm -q -f /bin/rpm /usr/bin/wget
rpm-4.14.1-1p151.13.10.x86_64
wget-1.19.5-1p151.4.1.x86_64
```

如果只知道部分文件名，则可以使用外壳脚本，如例 9.3 “搜索软件包的脚本” 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脚本。

### 例 9.3：搜索软件包的脚本

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

**rpm -q --changelog PACKAGE** 命令会按日期顺序显示有关特定软件包的详细更改信息列表。

借助已安装的 RPM 数据库，可以进行校验检查。使用 **-V** 或 **--verify** 可启动这些检查。将 **rpm** 与此选项搭配使用，将显示软件包中自安装后已更改的所有文件。**rpm** 使用八位字符符号来提供有关以下更改的提示：

表 9.2：RPM 校验选项

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

对于配置文件，将输出字母 **c**。例如，要检查 **/etc/wgetrc** (**wget** 软件包) 的更改，请使用以下命令：

```
> rpm -V wget
```

RPM 数据库的文件放置在 `/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 的空间，特别是在全面更新之后。如果数据库比预期大得多，建议使用选项 `--rebuilddb` 来重构建数据库。在执行此操作之前，制作旧数据库的备份。`cron` 脚本 `cron.daily` 每天都会创建数据库副本（打包成 gzip 格式）并将这些副本存储在 `/var/adm/backup/rpmdb` 中。副本数量由 `/etc/sysconfig/backup` 中的变量 `MAX_RPMDB_BACKUPS`（默认值为 5）控制。对于 1 GB 的 `/usr`，单个备份的大小大约为 1 MB。

## 9.2.5 安装和编译源软件包

所有源软件包的扩展名都是 `.src.rpm`（源 RPM）。



### 注意：安装的源软件包

源软件包可以从安装媒体复制到硬盘并使用 YaST 解压缩。但在软件包管理器中，它们不会被标记为已安装 ([i])。这是因为源软件包不是在 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”源软件包时，只将源代码添加到系统中。

在 `/usr/src/packages` 中，必须存在用于 `rpm` 和 `rpmbuild` 的以下目录（除非在 `/etc/rpmrc` 这样的文件中指定了自定义设置）：

#### SOURCES

存放原始源（`.tar.bz2` 或 `.tar.gz` 文件等）和特定于发行套件的调整（多为 `.diff` 或 `.patch` 文件）

#### SPECS

代表 `.spec` 文件，类似于元 `Makefile`，可控制构建进程

#### BUILD

在此目录中解压缩、修补和编译所有源

#### RPMS

存储完整的二进制软件包的位置

这里是源 RPM

使用 YaST 安装源软件包时，所有必需的组件都安装在 `/usr/src/packages` 中：源和调整项存放于 `SOURCES` 中，相关的 `.spec` 文件存放于 `SPECS` 中。



### 警告：系统完整性

不要对系统组件（`glibc`、`rpm` 等）进行试验，因为这样会影响系统的稳定性。

下面的示例使用 `wget.src.rpm` 软件包。安装源软件包后，应具有类似以下列表中的文件：

```
/usr/src/packages/SOURCES/wget-1.19.5.tar.bz2  
/usr/src/packages/SOURCES/wgetrc.patch  
/usr/src/packages/SPECS/wget.spec
```

`rpmbuild -bX /usr/src/packages/SPECS/wget.spec` 会启动编译。`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：解压和修补。

-bc

执行与 -bp 相同的操作，但还会进行编译。

-bi

执行与 -bp 相同的操作，但还会安装构建的软件。注意：如果软件包不支持 BuildRoot 功能，则可能会重写配置文件。

-bb

执行与 -bi 相同的操作，但还会创建二进制软件包。如果编译成功，二进制软件包应该存放于 `/usr/src/packages/RPMS` 中。

-ba

执行与 -bb 相同的操作，但还会创建源 RPM。如果编译成功，二进制软件包应该存放于 `/usr/src/packages/SRPMS` 中。

## --short-circuit

跳过某些步骤。

现在可以使用 rpm -i (最好使用 rpm -U) 创建二进制 RPM。使用 rpm 进行安装使它显示在 RPM 数据库中。

请记住，规范文件中的 BuildRoot 指令已弃用。如果您仍然需要此功能，请使用 --buildroot 选项作为替代方案。

## 9.2.6 使用 build 编译 RPM 软件包

许多软件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，它将创建构建软件包的已定义环境。要建立这一 chroot 环境，build 脚本必须和完整的软件包树结构一起提供。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 命令会在源目录中查找 .spec 文件。要使用系统中 /media/dvd 下挂载的 DVD 构建 wget (如上例所示)，请以 root 用户身份运行以下命令：

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

之后，系统便会在 /var/tmp/build-root 中建立一个极简环境。在此环境中构建软件包。完成后，生成的软件包将位于 /var/tmp/build-root/usr/src/packages/RPMS 中。

build 脚本提供多个其他选项。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。有关更多信息，请使用 build --help 以及参见 build 手册页。

## 9.2.7 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 键可显示 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。

拥有全部功能的软件包管理器将作为 YaST 模块提供。有关详细信息，请参见[第 8 章 “安装或去除软件”](#)。

# 10 通过 Snapper 进行系统恢复和快照管理

Snapper 可用于创建和管理文件系统快照。文件系统快照可用于保留文件系统在某个时间点的状态副本。Snapper 的标准设置旨在实现系统更改的回滚。但是，您也可以使用它来创建用户数据的磁盘备份。Snapper 基于 Btrfs 文件系统或者采用 XFS（或 Ext4）文件系统的精简配置 LVM 卷实现此功能。

Snapper 具有命令行界面和 YaST 界面。Snapper 可让您在以下类型的文件系统上创建和管理文件系统快照：

- Btrfs：适用于 Linux 的“写入时复制”文件系统，其本身支持子卷的文件系统快照。（子卷是物理分区中可单独挂载的文件系统。）  
您也可以从 Btrfs 快照引导。有关详细信息，请参见[第 10.3 节 “通过从快照引导来执行系统回滚”](#)。
- 采用 XFS 或 Ext4 格式化的精简配置 LVM 卷。

您可以使用 Snapper 执行以下任务：

- 撤销 zypper 和 YaST 所做的系统更改。有关详细信息，请参见[第 10.2 节 “使用 Snapper 撤销更改”](#)。
- 通过先前的快照恢复文件。有关详细信息，请参见[第 10.2.2 节 “使用 Snapper 恢复文件”](#)。
- 从快照引导以执行系统回滚。有关详细信息，请参见[第 10.3 节 “通过从快照引导来执行系统回滚”](#)。
- 在正在运行的系统中手动创建和管理快照。有关详细信息，请参见[第 10.6 节 “手动创建和管理快照”](#)。

## 10.1 默认设置

SUSE Linux Enterprise Desktop 上的 Snapper 设置为系统更改的撤销和恢复工具。默认情况下，SUSE Linux Enterprise Desktop 的根分区 (/) 使用 Btrfs 格式。如果根分区 (/) 足够大（约为 16 GB 以上），创建快照功能便会自动启用。默认情况下，在除 / 以外的分区上会禁用快照。



### 提示：已在安装系统中启用 Snapper

如果您在安装期间禁用了 Snapper，以后随时都可启用它。要执行此操作，请运行以下命令创建根文件系统的默认 Snapper 配置：

```
> sudo snapper -c root create-config /
```

之后，按[第 10.1.4.1 节 “禁用/启用快照”](#) 中所述启用不同的快照类型。

请注意，要在 Btrfs 根文件系统上使用快照，需按照安装程序的建议配置包含子卷的文件系统，并提供一个大小至少为 16 GB 的分区。

创建快照时，快照与原始点都会指向文件系统中的同一个块。因此一开始时快照并不占用额外的磁盘空间。但如果修改了原始文件系统中的数据，则会复制已更改的数据块，同时将旧的数据块作为快照保留。因此，快照就将占用与已修改数据相同的空间。所以久而久之，分配给快照的空间便会不断增长。其结果是，即便从包含快照的 Btrfs 文件系统删除文件可能也**不会** 释放磁盘空间！



### 注意：快照位置

快照始终存放在创建快照的那个分区或子卷中，而无法将快照存储到其他分区或子卷。

因此，包含快照的分区需大于不包含快照的分区。确切的空间大小在很大程度上取决于要保留的快照数量以及数据修改量。按照以往经验，应为分区分配两倍于一般情况下使用的空间。为了防止磁盘上的空间耗尽，系统会自动清理旧快照。有关细节，请参考[第 10.1.4.4 节 “控制快照存档”](#)。

## 10.1.1 默认设置

### 大于 16 GB 的磁盘

- 配置文件: /etc/snapper/configs/root
- USE\_SNAPPER=yes
- TIMELINE\_CREATE=no

### 小于 16 GB 的磁盘

- 配置文件: 未创建
- USE\_SNAPPER=no
- TIMELINE\_CREATE=yes

## 10.1.2 快照类型

尽管快照本身在技术方面并无区别，但我们根据触发它们的事件将其分成三类：

### 时间线快照

每小时创建一个快照，且旧的快照会自动删除。默认情况下，会保留最近十天、最近十个月以及最近十年的首张快照。使用 YaST OS 安装方法（默认）时，将启用时间线快照（根文件系统除外）。

### 安装快照

每次使用 YaST 或 Zypper 安装一个或多个软件包时，会创建一对快照：一个是在安装开始前（“前”），另一个是在安装完成后（“后”）。如果安装了内核等重要的系统组件，快照将被标记为重要 (important=yes)。旧的快照会自动删除。默认情况下，会保留最近十个重要快照和最近十个“普通”快照（包括管理快照）。安装快照默认为启用状态。

### 管理快照

每次使用 YaST 管理系统时都会创建一对快照：一个是在 YaST 模块启动之前（“前”），另一个是在该模块关闭之后（“后”）。旧的快照会自动删除。默认情况下，会保留最近十个重要快照和最近十个“普通”快照（包括安装快照）。管理快照默认为启用状态。

### 10.1.3 从快照中排除的目录

出于多种不同的理由，有些目录需要排除在快照之外。以下列表显示了排除的所有目录：

/boot/grub2/i386-pc、/boot/grub2/x86\_64-efi、/boot/grub2/powerpc-ieee1275、/boot/grub2/s390x-emu

不能回滚引导加载程序配置。上面列出的目录是架构专属目录。前两个目录位于 AMD64/Intel 64 计算机上，后两个目录分别位于 IBM POWER 和 IBM Z 上。

/home

如果独立的分区中没有 /home，便会将该目录排除以免在回滚时发生数据丢失。

/opt

第三方产品通常安装到 /opt 下。排除此目录是为了防止在回滚时卸装这些应用程序。

/srv

包含 Web 和 FTP 服务器的数据。排除此目录是为了防止在回滚时发生数据丢失。

/tmp

包含临时文件和缓存的所有目录都会排除在快照范围之外。

/usr/local

在手动安装软件时会用到此目录。系统会将该目录排除以免在回滚时卸载这些安装的软件。

/var

此目录包含许多变量文件（包括日志、暂时缓存、/var/opt 中的第三方产品），是虚拟机映像和数据库的默认位置。因此，创建此子卷是为了从快照中排除所有这些变量数据，且已禁用“写入时复制”。

### 10.1.4 自定义设置

SUSE Linux Enterprise Desktop 自带的默认设置设计合理，适合大多数使用情况。不过，您可以根据自己的需要对创建自动快照以及保留快照的各个方面进行配置。

### 10.1.4.1 禁用/启用快照

这三种快照类型（时间线、安装、管理）都可以单独启用或禁用。

#### 禁用/启用时间线快照

**启用:** snapper -c root set-config "TIMELINE\_CREATE=yes"

**禁用:** snapper -c root set-config "TIMELINE\_CREATE=no"

使用 YaST OS 安装方法（默认）时，将启用时间线快照（根文件系统除外）。

#### 禁用/启用安装快照

**启用:** 安装软件包 snapper-zypp-plugin

**禁用:** 卸装 snapper-zypp-plugin 软件包

安装快照默认为启用状态。

#### 禁用/启用管理快照

**启用:** 在 /etc/sysconfig/yast2 中将 USE\_SNAPPER 设置为 yes。

**禁用:** 在 /etc/sysconfig/yast2 中将 USE\_SNAPPER 设置为 no。

管理快照默认为启用状态。

### 10.1.4.2 控制安装快照

使用 YaST 或 Zypper 安装软件包时所创建的快照会由 snapper-zypp-plugin 进行处理。何时创建快照由 XML 配置文件 /etc/snapper/zypp-plugin.conf 定义。默认情况下，该文件如下所示：

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <snapper-zypp-plugin-conf>
3   <solvables>
4     <solvable match="w" ①  important="true" ②>kernel-* ③</solvable>
5     <solvable match="w"  important="true">dracut</solvable>
6     <solvable match="w"  important="true">glibc</solvable>
7     <solvable match="w"  important="true">systemd*</solvable>
8     <solvable match="w"  important="true">udev</solvable>
9     <solvable match="w">*</solvable> ④
```

```
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

- ① `match` 属性定义软件集是 Unix 外壳风格的通配符 (`w`) 还是 Python 正则表达式 (`re`)。
- ② 如果匹配指定软件集而且对应的软件包标记为 `important` (例如内核软件包) , 则快照也会标记为 `important`。
- ③ 用于匹配软件包名称的软件集。根据 `match` 属性的设置, 特殊字符可能会被解析为外壳通配符或是正则表达式。此软件集匹配名称以 `kernel-` 开头的所有软件包。
- ④ 此行无条件匹配所有软件包。

在这样的快照配置下, 只要安装了软件包, 就会创建快照对 (第 9 行) 。如果标记为 `important` 的内核、`dracut`、`glibc`、`systemd` 或 `udev` 软件包已安装, 快照对也会标记为 `important` (第 4 行到第 8 行) 。所有规则都会进行评估。

要禁用某规则, 可以删除该规则或通过 XML 注释的方式将其停用。举例来说, 如果不希望系统在每次安装软件包时创建快照对, 可以将第 9 行注释掉:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <snapper-zypp-plugin-conf>
3  <solvables>
4    <solvable match="w" important="true">kernel-*</solvable>
5    <solvable match="w" important="true">dracut</solvable>
6    <solvable match="w" important="true">glibc</solvable>
7    <solvable match="w" important="true">systemd*</solvable>
8    <solvable match="w" important="true">udev</solvable>
9    <!-- <solvable match="w">*</solvable> -->
10   </solvables>
11 </snapper-zypp-plugin-conf>
```

#### 10.1.4.3 创建和挂载新子卷

系统支持在 `/` 层次下创建新的子卷, 并永久挂载该卷。此类子卷将从快照中排除。切勿在现有快照中创建此类子卷, 因为在回滚之后, 您将无法再删除快照。

SUSE Linux Enterprise Desktop 中配置了 `/@/` 子卷, 该子卷充当永久子卷 (例如 `/opt`、`/srv`、`/home` 等) 的独立根目录。您创建和永久挂载的任何新子卷都需要在这个初始根文件系统中创建。

为此，请运行以下命令。在此示例中，从 /dev/sda2 创建了一个新子卷 /usr/important。

```
> sudo mount /dev/sda2 -o subvol=@ /mnt
> sudo btrfs subvolume create /mnt/usr/important
> sudo umount /mnt
```

/etc/fstab 中的相应项需类似于：

```
/dev/sda2 /usr/important btrfs subvol=@/usr/important 0 0
```



### 提示：禁用写入时复制 (cow)

子卷可能包含经常更改的文件，例如虚拟化的磁盘映像、数据库文件或日志文件。如果是这样，可考虑对此卷禁用写入时复制功能，以免复制磁盘块。可在 /etc/fstab 中使用 nodatacow 挂载选项来实现此目的：

```
/dev/sda2 /usr/important btrfs nodatacow,subvol=@/usr/important 0 0
```

或者，要为单个文件或目录禁用写入时复制功能，请使用命令 chattr +C PATH。

#### 10.1.4.4 控制快照存档

快照会占用磁盘空间。为了防止磁盘空间耗尽而导致系统中断，旧的快照会自动删除。默认情况下，将保留最多 10 个重要的安装快照与管理快照，以及最多 10 个普通的安装快照与管理快照。如果这些快照占用的空间超过根文件系统大小的 50%，将会删除其他快照。系统始终会至少保留 4 个重要快照和 2 个普通快照。

有关如何更改这些值的指导，请参考[第 10.5.1 节 “管理现有配置”](#)。

#### 10.1.4.5 在精简配置的 LVM 卷上使用快照

除了在 Btrfs 文件系统上创建快照之外，Snapper 还支持在 XFS、Ext4 或 Ext3 格式化的精简配置 LVM 卷（不支持在常规 LVM 卷上创建快照）上创建快照。有关 LVM 卷的详细信息和设置指导，请参考《Deployment Guide》，第 6 章 “Expert Partitioner”，第 6.2 节 “LVM configuration”。

要在精简配置的 LVM 卷上使用 Snapper，需要为其创建 Snapper 配置。在 LVM 上，需要使用 `--fstype=lvm(FILESYSTEM)` 指定文件系统。`FILESYSTEM` 的有效值为 `ext3`、`ext4` 或 `xfs`。示例：

```
> sudo snapper -c lvm create-config --fstype="lvm(xfs)" /thin_lvm
```

您可以按照[第 10.5.1 节 “管理现有配置”](#) 中的说明根据需要调整此配置。

## 10.2 使用 Snapper 撤销更改

SUSE Linux Enterprise Desktop 上的 Snapper 经过预配置，可以用来撤消 `zypper` 和 YaST 所做的更改。要实现此功能，请对 Snapper 进行配置，让其在每次运行 `zypper` 和 YaST 前后创建一个快照对。您也可以使用 Snapper 来恢复被意外删除或修改的系统文件。要实现此目的，需要对根分区启用时间线快照 — 有关细节，请参见[第 10.1.4.1 节 “禁用/启用快照”](#)。

默认情况下，上述的自动快照针对根分区及其子卷所配置。若希望针对 `/home` 等其他分区生成快照，您可以创建自定义配置。

### ! 重要：撤销更改与回滚的比较

通过快照来恢复数据时，必须知道，Snapper 可以处理两种完全不同的情形：

#### 撤销更改

在如下文中所述撤销更改时，系统会对两个快照进行比较，并撤销两个快照之间的更改。借助这种方式可以明确地选择要恢复的文件。

#### 回滚

在如[第 10.3 节 “通过从快照引导来执行系统回滚”](#) 中所述进行回滚时，系统会重置为创建快照时的状态。

撤销更改时，可以将快照与现有系统进行比较。如果比较后恢复所有发生变化的文件，那么结果会和回滚完全相同。但是，还是建议使用[第 10.3 节 “通过从快照引导来执行系统回滚”](#) 中介绍的方法进行回滚，因为回滚操作的速度更快，而且您可以在进行回滚前查看系统。



## 警告：数据一致性

在创建快照时并没有能确保数据一致性的机制。如果在创建快照的同时写入某个文件（例如数据库），将导致文件损坏或写入不完整。恢复此类文件会产生问题。此外，有些系统文件（例如 `/etc/mtab`）是绝对不能恢复的。因此强烈建议您要始终仔细查看已更改文件及其差异的列表。只恢复您要还原的操作真正包含的文件。

### 10.2.1 撤消 YaST 和 Zypper 更改

如果在安装时使用 `Btrfs` 设置根分区，则将自动安装 Snapper（经过预配置，可以回滚 YaST 或 Zypper 所做的更改）。每次启动 YaST 模块或 Zypper 事务时，会创建两个快照：即截获模块启动之前文件系统状态的“前快照”以及截获模块完成之后状态的“后快照”。

您可以使用 YaST Snapper 模块或 `snapper` 命令行工具，通过从“前快照”恢复文件来撤销 YaST 或 Zypper 所做的更改。您也可以使用该工具比较这两张快照，以查看更改了哪些文件。您还可以显示文件的两个版本之间的差异 (`diff`)。

#### 过程 10.1：使用 YaST 快照程序模块撤销更改

1. 从 YaST 中的其他部分或通过输入 `yast2 snapper` 来启动 Snapper 模块。
2. 务必将当前配置设置为根。除非手动添加自己的 Snapper 配置，否则请始终做此设置。
3. 从列表中选择前后快照对。YaST 和 Zypper 快照对都属于前后类型。在说明列中，YaST 快照标为 `zypp(y2base)`；Zypper 快照标为 `zypp(zypper)`。

快照

当前配置 root

| ID      | 类型      | 开始日期                | 结束日期                | 描述                              | 用户数据          |
|---------|---------|---------------------|---------------------|---------------------------------|---------------|
| 1       | 单一      | 2022-06-22 03:50:18 |                     | first root filesystem           |               |
| 2       | 单一      | 2022-06-22 04:16:40 |                     | after installation              | important=yes |
| 4 & 5   | 前快照和后快照 | 2022-06-22 04:25:50 | 2022-06-22 04:29:20 | yast repositories               |               |
| 3 & 6   | 前快照和后快照 | 2022-06-22 04:25:48 | 2022-06-22 04:29:37 | yast sw_single                  |               |
| 7 & 8   | 前快照和后快照 | 2022-06-22 04:29:39 | 2022-06-22 04:30:34 | yast repositories               |               |
| 10 & 11 | 前快照和后快照 | 2022-06-22 04:31:26 | 2022-06-22 04:31:30 | zypp(ruby.ruby2.5) important=no |               |
| 9 & 12  | 前快照和后快照 | 2022-06-22 04:30:36 | 2022-06-22 04:33:00 | yast sw_single                  |               |
| 14 & 15 | 前快照和后快照 | 2022-06-22 04:33:29 | 2022-06-22 04:33:35 | zypp(ruby.ruby2.5) important=no |               |
| 13 & 16 | 前快照和后快照 | 2022-06-22 04:33:02 | 2022-06-22 04:33:40 | yast sw_single                  |               |
| 18 & 19 | 前快照和后快照 | 2022-06-22 04:34:05 | 2022-06-22 04:34:13 | zypp(ruby.ruby2.5) important=no |               |
| 17 & 20 | 前快照和后快照 | 2022-06-22 04:33:42 | 2022-06-22 04:34:42 | yast sw_single                  |               |
| 21 & 22 | 前快照和后快照 | 2022-06-22 04:34:44 | 2022-06-22 04:42:34 | yast lan                        |               |
| 23 & 24 | 前快照和后快照 | 2022-06-22 04:42:40 | 2022-06-22 04:49:16 | yast ntp-client                 |               |
| 25      | 前快照     | 2022-06-22 04:49:23 |                     | yast snapper                    |               |

显示更改 | 创建(I) | 修改(M) | 删除(T) | 刷新(R)

帮助(H) | 关闭(L)

4. 单击显示更改，会打开一个文件列表显示两张快照之间的差异。

选定快照概述

/ zypp(y2base)

11 & 14

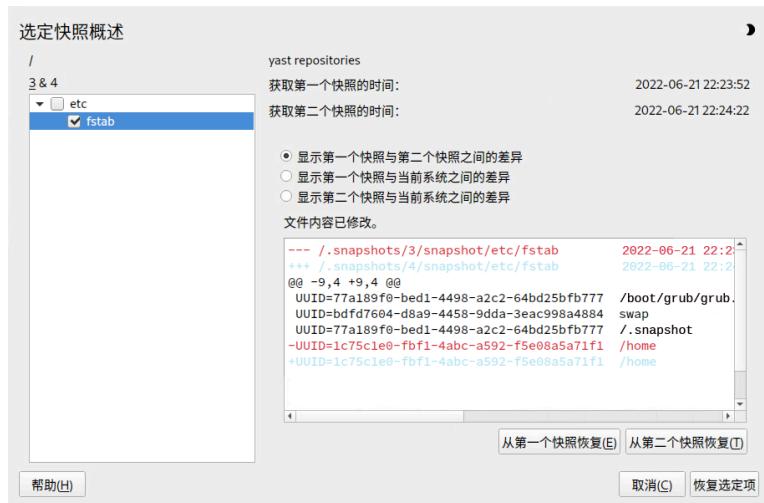
- var
- lib
- rpm
  - Basenames
  - Conflictname
  - Dirnames
  - Group
  - Installtid
  - Name
  - Packages
  - Providename
  - Requirename
  - Sha1header
  - Sigmd5
- zypp
  - Autoinstalled

获取第一个快照的时间: 2016-08-08 17:20:59

获取第二个快照的时间: 2016-08-08 17:21:37

帮助(H) | 取消(C) | 恢复选定

5. 查看文件列表。要显示文件的前后版本之间的“差异”，请从列表中选中该文件。



6. 要恢复一个或多个文件，请通过勾选相应的复选框选择相关的文件或目录。单击恢复选定，然后单击是以确认该操作。



要恢复单一文件，请单击其名称以激活该文件的差异视图。单击从第一个快照恢复，然后单击是予以确认。

## 过程 10.2：使用 **snapper** 命令撤销更改

1. 运行 **snapper list -t pre-post** 获得 YaST 与 Zypper 快照的列表。在说明列中, YaST 快照标为 yast MODULE\_NAME; Zypper 快照标为 zypp(zypper)。

```
> sudo snapper list -t pre-post
Pre # | Post # | Pre Date           | Post Date          | Description
-----+-----+-----+-----+-----+-----+
311  | 312   | Tue 06 May 2018 14:05:46 CEST | Tue 06 May 2018 14:05:52 CEST | zypp(y2base)
340  | 341   | Wed 07 May 2018 16:15:10 CEST | Wed 07 May 2018 16:15:16 CEST | zypp(zypper)
342  | 343   | Wed 07 May 2018 16:20:38 CEST | Wed 07 May 2018 16:20:42 CEST | zypp(y2base)
344  | 345   | Wed 07 May 2018 16:21:23 CEST | Wed 07 May 2018 16:21:24 CEST | zypp(zypper)
346  | 347   | Wed 07 May 2018 16:41:06 CEST | Wed 07 May 2018 16:41:10 CEST | zypp(y2base)
348  | 349   | Wed 07 May 2018 16:44:50 CEST | Wed 07 May 2018 16:44:53 CEST | zypp(y2base)
350  | 351   | Wed 07 May 2018 16:46:27 CEST | Wed 07 May 2018 16:46:38 CEST | zypp(y2base)
```

2. 使用以下命令获得快照对的已更改文件列表: **snapper status PRE..POST**。文件内容发生了更改会以 **c** 标记、添加了文件会以 **+** 标记、删除了文件会以 **-** 标记。

```
> sudo snapper status 350..351
+..... /usr/share/doc/packages/mikachan-fonts
+..... /usr/share/doc/packages/mikachan-fonts/COPYING
+..... /usr/share/doc/packages/mikachan-fonts/dl.html
c..... /usr/share/fonts/truetype/fonts.dir
c..... /usr/share/fonts/truetype/fonts.scale
+..... /usr/share/fonts/truetype/みかちゃん-p.ttf
+..... /usr/share/fonts/truetype/みかちゃん-pb.ttf
+..... /usr/share/fonts/truetype/みかちゃん-ps.ttf
+..... /usr/share/fonts/truetype/みかちゃん.ttf
c..... /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-
x86_64.cache-4
c..... /var/lib/rpm/Basenames
c..... /var/lib/rpm/Dirnames
c..... /var/lib/rpm/Group
c..... /var/lib/rpm/Installtid
c..... /var/lib/rpm/Name
c..... /var/lib/rpm/Packages
c..... /var/lib/rpm/Providename
c..... /var/lib/rpm/Requirename
c..... /var/lib/rpm/Shalheader
c..... /var/lib/rpm/Sigmd5
```

3. 要显示特定文件的差异, 请运行 **snapper diff PRE..POST FILENAME**。如果不指定 FILENAME, 将会显示所有文件的差异。

```
> sudo snapper diff 350..351 /usr/share/fonts/truetype/fonts.scale
--- /.snapshots/350/snapshot/usr/share/fonts/truetype/fonts.scale
2014-04-23 15:58:57.000000000 +0200
+++ /.snapshots/351/snapshot/usr/share/fonts/truetype/fonts.scale
2014-05-07 16:46:31.000000000 +0200
@@ -1,4 +1,4 @@
-1174
+1486
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-
iso10646-1
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso8859-1
[...]
```

4. 要恢复一或多个文件, 请运行 **snapper -v undochange PRE..POST FILENAMES**。如果不指定 FILENAMES, 将会恢复所有已更改的文件。

```
> sudo snapper -v undochange 350..351
create:0 modify:13 delete:7
undoing change...
deleting /usr/share/doc/packages/mikachan-fonts
deleting /usr/share/doc/packages/mikachan-fonts/COPYING
deleting /usr/share/doc/packages/mikachan-fonts/dl.html
deleting /usr/share/fonts/truetype/みかちゃん-p.ttf
deleting /usr/share/fonts/truetype/みかちゃん-pb.ttf
deleting /usr/share/fonts/truetype/みかちゃん-ps.ttf
deleting /usr/share/fonts/truetype/みかちゃん.ttf
modifying /usr/share/fonts/truetype/fonts.dir
modifying /usr/share/fonts/truetype/fonts.scale
modifying /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-
x86_64.cache-4
modifying /var/lib/rpm/Basenames
modifying /var/lib/rpm/Dirnames
modifying /var/lib/rpm/Group
modifying /var/lib/rpm/Installtid
modifying /var/lib/rpm/Name
```

```
modifying /var/lib/rpm/Packages
modifying /var/lib/rpm/Providename
modifying /var/lib/rpm/Requirename
modifying /var/lib/rpm/Shalheader
modifying /var/lib/rpm/Sigmd5
undoing change done
```



## 警告：还原用户添加操作

建议不要使用 Snapper 通过撤销更改的方式来还原用户添加。因为快照中排除了一些目录，文件系统中将保留属于用户的文件。如果使用已删除用户的 ID 创建新用户，该用户将继承原用户的文件。因此，强烈建议您使用 YaST 的用户和组管理工具来删除用户。

### 10.2.2 使用 Snapper 恢复文件

除了安装和管理快照之外，Snapper 还会创建时间线快照。您可以使用这些备份快照还原意外删除的文件或文件的前一个版本。您也可以使用 Snapper 的差异功能来确定在某个时间点所做的修改。

还原文件功能对于默认情况下不会生成快照的那些子卷或分区上的数据特别有用。例如，要从主目录还原文件，可以为自动生成时间线快照的 `/home` 创建单独的 Snapper 配置。有关指导，请参见 [第 10.5 节 “创建并修改 Snapper 配置”](#)。



## 警告：恢复文件与回滚的比较

从根文件系统（由 Snapper 的根配置定义）生成的快照可用于执行系统回滚。进行此类回滚时，建议先从快照引导，然后再执行回滚。有关详细信息，请参见 [第 10.3 节 “通过从快照引导来执行系统回滚”](#)。

从根文件系统快照还原所有文件（如下文中所述）也可以达到回滚的目的，但不建议这样做。您可以恢复单个文件，例如恢复 `/etc` 目录中的某个配置文件，但不能恢复快照中所列的全部文件。

此限制仅针对从根文件系统建立的快照。

### 过程 10.3：使用 YaST Snapper 模块恢复文件

1. 从 YaST 中的其他部分或通过输入 **yast2 snapper** 来启动 Snapper 模块。
2. 选择要从中选择快照的当前配置。
3. 选择要用于恢复文件的时间线快照，并选择显示更改。时间线快照的类型为单一，描述为时间线。
4. 单击文件名从文本框中选择文件。系统会显示快照版本和当前系统之间的差异。勾选复选框来选中要恢复的文件。请对要恢复的所有文件执行此操作。
5. 单击恢复选定，然后单击是以确认该操作。

### 过程 10.4：使用 snapper 命令恢复文件

1. 运行以下命令来显示特定配置的一系列时间线快照：

```
> sudo snapper -c CONFIG list -t single | grep timeline
```

CONFIG 需要替换为现有的 Snapper 配置。使用 **snapper list-configs** 可显示列表。

2. 运行以下命令来显示指定快照中发生更改的一系列文件：

```
> sudo snapper -c CONFIG status SNAPSHOT_ID..0
```

将 SNAPSHOT\_ID 替换为您要用于恢复文件的快照的 ID。

3. (可选) 运行以下命令列出当前文件版本与快照中文件版本的差异：

```
> sudo snapper -c CONFIG diff SNAPSHOT_ID..0 FILE NAME
```

如果不指定 FILE NAME，将会显示所有文件的差异。

4. 要恢复一个或多个文件，请运行

```
> sudo snapper -c CONFIG -v undochange SNAPSHOT_ID..0 FILENAME1 FILENAME2
```

如果没有指定文件名，则会恢复所有已更改的文件。

## 10.3 通过从快照引导来执行系统回滚

SUSE Linux Enterprise Desktop 上包含的 GRUB 2 版本可以从 Btrfs 快照进行引导。与 Snapper 的回滚功能相结合，就能恢复配置错误的系统。只有针对默认 Snapper 配置 (`root`) 创建的快照才可引导。

### ! 重要：支持的配置

从 SUSE Linux Enterprise Desktop 15 SP5 开始，只有在根分区的默认子卷配置未更改过的情况下，才支持系统回滚。

引导快照时，快照中包含的文件系统部分会装载为只读状态；从快照中排除的所有其他文件系统和部分会加载为读写状态，并且可以修改。

### ! 重要：撤消更改与回滚的比较

通过快照来恢复数据时，必须知道，Snapper 可以处理两种完全不同的情形：

#### 撤消更改

在如第 10.2 节 “使用 Snapper 撤销更改” 中所述撤消更改时，系统会对两个快照进行比较，并还原两个快照之间的更改。通过这种方式可以明确指定不还原所选的文件。

#### 回滚

在如下文所述进行回滚时，系统会重置为生成快照时的状态。

要从可引导快照进行回滚，必须满足以下要求。执行默认安装时，系统会进行相应的设置。

#### 从可引导快照执行回滚的要求

- 根文件系统必须是 Btrfs。不支持从 LVM 卷快照引导。
- 根文件系统必须在单个设备、单个分区和单个子卷上。`/srv` 等快照中排除的目录（参见第 10.1.3 节 “从快照中排除的目录” 获取完整列表）可以位于单独的分区。
- 系统应能够借助安装的引导加载程序进行引导。

要从可引导快照进行回滚，请执行以下操作：

1. 引导系统。在引导菜单中选择可引导快照，然后选择要引导的快照。快照会按日期顺序从近到远一一列出。
2. 登录系统。仔细检查是否一切运行正常。请注意，您无法对快照包含的任何目录进行写操作。写入其他目录的数据无论您下一步选择什么操作都**不会丢失**。
3. 根据您是否要执行回滚，选择下一步操作：
  - a. 如果您不想对当前状态的系统执行回滚，请重引导进入当前的系统状态。然后，您便可选择另一个快照，或是启动救援系统。
  - b. 要执行回滚，请运行

```
> sudo snapper rollback
```

然后重引导。在引导屏幕上，选择默认的引导项以重引导至恢复后的系统。系统即会创建回滚前文件系统状态的快照。根的默认子卷将替换为全新的读写快照。有关详细信息，请参见第 10.3.1 节 “回滚后的快照”。

通过 -d 选项添加快照的说明非常实用。例如：

```
New file system root since rollback on DATE TIME
```



### 提示：回滚到特定的安装状态

如果安装期间未禁用快照，将在初始系统安装结束时创建初始可引导快照。您随时可以通过引导此快照返回到该状态。该快照可通过 after installation 说明识别。

开始对服务包或新的主要版本进行系统升级时，也会创建可引导快照（前提是未禁用快照）。

## 10.3.1 回滚后的快照

在执行回滚之前，将会创建正在运行的文件系统的快照。快照说明会引用在回滚中恢复的快照的 ID。

对于通过回滚创建的快照，其 `Cleanup` 属性的值会设为 `number`。因此，回滚快照会在达到设置的快照数后自动删除。有关细节，请参考第 10.7 节 “快照的自动清理”。如果快照包含重要数据，请在系统去除快照之前从快照中提取数据。

### 10.3.1.1 回滚快照示例

例如，在全新安装之后，系统上存在以下可用的快照：

```
# snapper --iso list
Type | # | Cleanup | Description | Userdata
-----+---+ +-----+-----+-----+
single | 0 | | current | |
single | 1 | | first root filesystem | |
single | 2 | | number | after installation | important=yes
```

运行 `sudo snapper rollback` 之后，将会创建快照 3，它包含执行回滚前系统的状态。快照 4 是新的默认 Btrfs 子卷，因此是重引导之后的系统。

```
# snapper --iso list
Type | # | Cleanup | Description | Userdata
-----+---+ +-----+-----+-----+
single | 0 | | current | |
single | 1 | | number | first root filesystem | |
single | 2 | | number | after installation | important=yes
single | 3 | | number | rollback backup of #1 | important=yes
single | 4 | | | |
```

### 10.3.2 访问和识别快照引导项

要从快照引导，请重引导计算机并选择从只读的快照启动引导加载程序。一个屏幕即会打开，列出所有可引导的快照。最近的快照列在最前面，最旧的快照列在最后面。使用 `↓` 和 `↑` 导航，然后按 `Enter` 激活选定的快照。从引导菜单激活快照不会立即重引导计算机，而是打开选定快照的引导加载程序。

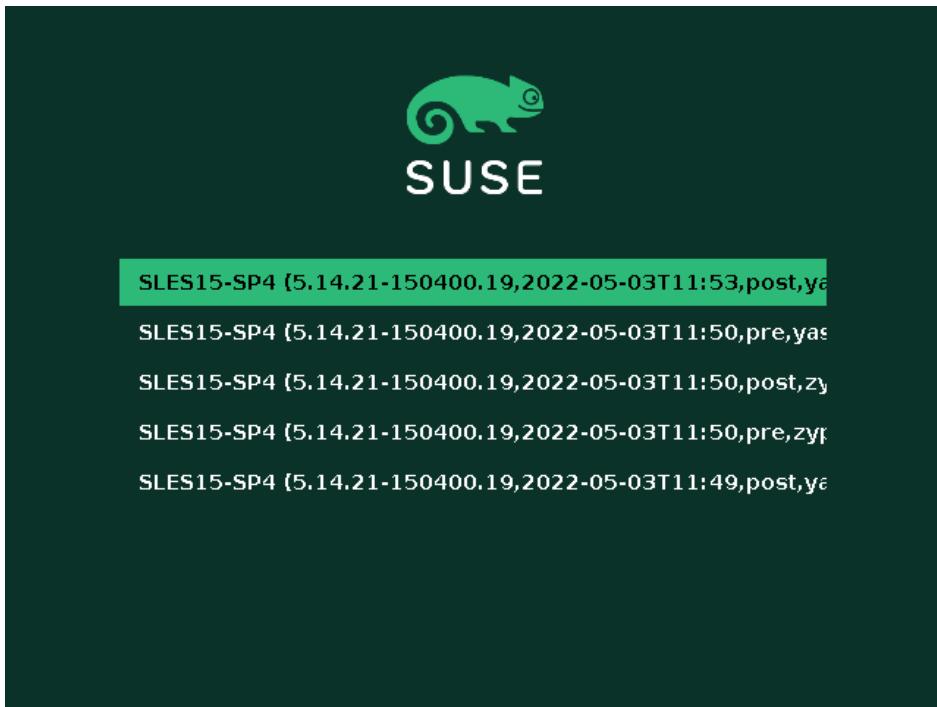


图 10.1：引导加载程序：快照



**警告：**目前无法使用 UEFI 从 Btrfs 快照引导 Xen

有关更多详细信息，请参见<https://www.suse.com/support/kb/doc/?id=000020602>。

引导加载程序中的每个快照项遵循一种可方便您识别快照的命名模式：

```
[*] ① OS ② (KERNEL ③ ,DATE ④ TTIME ⑤ ,DESCRIPTION ⑥ )
```

- ① 如果快照标记为 important，该项将标有 \*。
- ② 操作系统标签。
- ④ 采用 YYYY-MM-DD 格式的日期。
- ⑤ 采用 HH:MM 格式的时间。
- ⑥ 此字段包含快照的说明。对于手动创建的快照，这是使用选项 --description 创建的字符串，或自定义字符串（请参见 [提示：为引导加载程序快照项设置自定义说明](#)）。对于自动创建的快照，这是调用的工具，例如 zypp(zypper) 或 yast\_sw\_single。较长的说明可能会被截断，具体视引导屏幕的大小而定。



## 提示：为引导加载程序快照项设置自定义说明

可以将快照说明字段中的默认字符串替换为自定义字符串。例如，如果自动创建的说明不能提供充分的描述，或者用户提供的说明太长，这种做法将十分有用。要为快照 NUMBER 设置自定义字符串 STRING，请使用以下命令：

```
> sudo snapper modify --userdata "bootloader=STRING" NUMBER
```

说明的长度不应超过 25 个字符，超过此大小的任何内容都无法在引导屏幕上正常显示。

## 10.3.3 限制

不可能实现完整的系统回滚，即将整个系统恢复到生成快照时完全相同的状态。

### 10.3.3.1 从快照中排除的目录

根文件系统快照并不包含所有目录。请参见第 10.1.3 节“从快照中排除的目录”了解详情和背后的原因。正因为此，这些目录中的数据并不会恢复，也就造成了以下限制。

#### 执行回滚后，附加产品和第三方软件可能会无法使用

如果应用程序和附加产品将一部分数据安装在从快照中排除的子卷（如 /opt）上，将另一部分数据安装在快照中包含的子卷上，回滚后，这些应用程序和附加产品将可能无法工作。要解决此问题，需要重新安装该应用程序或附加产品。

#### 文件访问问题

如果某个应用程序在快照和当前系统之间更改了文件权限和/或所有权，回滚后，该应用程序可能无法访问这些文件。请在回滚后重置受影响的文件的权限和/或所有权。

#### 不兼容的数据格式

如果服务或应用程序在快照和当前系统之间建立了新的数据格式，回滚后，该应用程序可能无法读取受影响的数据文件。

#### 混合了代码和数据的子卷

/srv 这样的子卷可能同时包含代码和数据。回滚可能会导致代码失效。例如，降级 PHP 的版本可能会导致 Web 服务器的 PHP 脚本被破坏。

## 用户数据

如果回滚操作从系统中删除了用户，这些用户在快照中未包含的目录上所拥有的数据并不会删除。如果使用相同的用户 ID 创建新用户，该用户便会继承原用户的文件。请使用 `find` 之类的工具找到并删除孤立的文件。

### 10.3.3.2 不回滚引导加载程序数据

无法回滚引导加载程序，因为引导加载程序的所有“阶段”必须相互匹配。在进行 `/boot` 回滚时无法保证这一点。

## 10.4 在用户主目录中启用 Snapper

您可以为用户的 `/home` 目录启用快照，以便为以下几个用例提供支持：

- 个人用户可以管理自己的快照和回滚。
- 系统用户，例如要跟踪配置文件、文档等数据副本的数据库、系统和网络管理员。
- Samba 与主目录和 Btrfs 后端进行共享。

每个用户的目录都是 `/home` 的 Btrfs 子卷。您可以手动设置快照（请参见第 10.4.3 节“手动启用主目录中的快照”），但更便捷的方式是使用 `pam_snapper`。`pam_snapper` 软件包用于安装 `pam_snapper.so` 模块和帮助程序脚本，这些模块和脚本可自动创建用户和配置 Snapper。

`pam_snapper` 提供与 `useradd` 命令、可插入身份验证模块 (PAM) 及 Snapper 的集成。默认情况下，它会在用户登录和注销时创建快照；当某些用户在很长一段时间内都处于登录状态时，它还会创建基于时间的快照。您可以使用常规 Snapper 命令和配置文件更改默认值。

### 10.4.1 安装 pam\_snapper 和创建用户

最简单的方法是使用 Btrfs 格式的新 `/home` 目录且在没有现有用户的情况下开始。安装 `pam_snapper`：

```
# zypper in pam_snapper
```

将下面一行添加到 `/etc/pam.d/common-session` 中：

```
session optional pam_snapper.so
```

使用 `/usr/lib/pam_snapper/pam_snapper_useradd.sh` 脚本创建新用户和主目录。该脚本默认会执行试运行。请编辑脚本，将 `DRYRUN=1` 更改为 `DRYRUN=0`。现在，您便可以创建新用户：

```
# /usr/lib/pam_snapper/pam_snapper_useradd.sh \
username group passwd=password
Create subvolume '/home/username'
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
```

用户首次登录时，`/etc/skel` 中的文件将会复制到其主目录。列出您的 Snapper 配置，以校验是否已创建用户的配置：

```
# snapper list --all
Config: home_username, subvolume: /home/username
Type  | # | Pre # | Date | User | Cleanup | Description | Userdata
-----+---+-----+-----+-----+-----+-----+-----+
single | 0 |       |      | root |       | current |
```

一段时间过后，此输出中将会填充快照列表，用户可以使用标准 Snapper 命令对其进行管理。

## 10.4.2 去除用户

使用 `/usr/lib/pam_snapper/pam_snapper_userdel.sh` 脚本去除用户。该脚本默认会执行试运行，请编辑脚本，将 `DRYRUN=1` 更改为 `DRYRUN=0`。此操作将去除用户、用户的主子卷、Snapper 配置，并会删除所有快照。

```
# /usr/lib/pam_snapper/pam_snapper_userdel.sh username
```

## 10.4.3 手动启用主目录中的快照

使用 Snapper 手动设置用户主目录的步骤如下。`/home` 必须为 Btrfs 格式，并且目前尚未创建用户。

```
# btrfs subvol create /home/username
# snapper -c home_username create-config /home/username
# sed -i -e "s/ALLOW_USERS=\\"\"/ALLOW_USERS=\"username\"/g" \
/etc/snapper/configs/home_username
# yast users add username=username home=/home/username password=password
# chown username.group /home/username
# chmod 755 /home/username/.snapshots
```

## 10.5 创建并修改 Snapper 配置

每一个分区或 Btrfs 子卷都有一个专用的配置文件用于定义 Snapper 的行为方式。这些配置文件位于 /etc/snapper/configs/ 下。

如果根文件系统足够大（大约有 12 GB），安装时将自动对根文件系统 / 启用快照。相应的默认配置命名为 root。该配置可创建和管理 YaST 及 Zypper 快照。有关默认值列表，请参见第 10.5.1.1 节 “配置数据”。



### 注意：启用快照所需的最小根文件系统大小

如第 10.1 节 “默认设置” 中所述，要启用快照，根文件系统中需要有额外的可用空间。

所需空间取决于所安装的软件包数量以及快照中包括的卷更改量，另外还取决于快照频率和存档的快照数。

要在安装期间自动启用快照，需要满足最小根文件系统大小。目前，此大小约为 12 GB。将来，这个值可能会发生变化，具体视基础系统的体系结构和大小而定。它取决于安装媒体内 /control.xml 文件中以下标记的值：

```
<root_base_size>
<btrfs_increase_percentage>
```

该值通过下面的公式计算得出：ROOT\_BASE\_SIZE \* (1 + BTRFS\_INCREASE\_PERCENTAGE / 100)

请记住，此值是最小大小。请考虑分给根文件系统更多空间。一般而言，两倍于未启用快照时将使用的大小即可。

您可以为使用 Btrfs 格式化的其他分区或 Btrfs 分区上的现有子卷创建自己的配置。在以下示例中，我们将设置 Snapper 配置，用于备份挂载到 /srv/www 且格式为 Btrfs 的独立分区上存放的 Web 服务器数据。

创建配置后，您可以直接使用 snapper，也可以使用 YaST Snapper 模块，从这些快照恢复文件。在 YaST 中，您需要选择您的当前配置，同时还需要使用全局开关 -c 指定 snapper 的配置（例如 snapper -c myconfig list）。

要创建新的 Snapper 配置，请运行 snapper create-config：

```
> sudo snapper -c www-data① create-config /srv/www②
```

- ① 配置文件的名称。
- ② 要生成快照的分区或 Btrfs 子卷的挂载点。

此命令将使用合理的默认值（取自 /etc/snapper/config-templates/default）创建新的配置文件 /etc/snapper/configs/www-data。有关如何调整这些值的指导，请参考[第 10.5.1 节 “管理现有配置”](#)。



## 提示：配置默认值

新配置的默认值取自 /etc/snapper/config-templates/default。要使用自己的一组默认值，请在相同的目录中创建此文件的副本然后按照需要进行调整。要采用这种方法，请在 create-config 命令中指定 -t 选项：

```
> sudo snapper -c www-data create-config -t MY_DEFAULTS /srv/www
```

## 10.5.1 管理现有配置

snapper 命令有多个子命令可用于管理现有配置。您可以列出、显示这些配置，也可以对它们进行删除和修改：

### 列出配置

使用 snapper list-configs 子命令可以获取所有现有配置：

```
> sudo snapper list-configs
```

```
Config | Subvolume
-----+-----
root  | /
usr   | /usr
local  | /local
```

## 显示配置

使用 `snapper -c CONFIG get-config` 子命令可以显示指定的配置。请将 `CONFIG` 替换为 `snapper list-configs` 显示的配置名称之一。有关配置选项的更多信息，请参见第 10.5.1.1 节 “配置数据”。

要显示默认配置，请运行：

```
> sudo snapper -c root get-config
```

## 修改配置

使用 `snapper -c CONFIG set-config OPTION=VALUE` 子命令可以修改指定配置中的选项。请将 `CONFIG` 替换为 `snapper list-configs` 显示的配置名称之一。有关 `OPTION` 和 `VALUE` 的可能值，请参见第 10.5.1.1 节 “配置数据”。

## 删除配置

使用 `snapper -c CONFIG delete-config` 子命令可以删除配置。请将 `CONFIG` 替换为 `snapper list-configs` 显示的配置名称之一。

## 10.5.1.1 配置数据

每个配置都包含一系列选项，这些选项可以通过命令行进行修改。下面的列表提供了每个选项的细节。要更改某个值，请运行 `snapper -c CONFIG set-config "KEY=VALUE"`。

### ALLOW\_GROUPS, ALLOW\_USERS

授予普通用户使用快照的权限。有关更多信息，请参见第 10.5.1.2 节 “以普通用户身份使用 Snapper”。

默认值为 `" "`。

### BACKGROUND\_COMPARISON

定义在创建前后快照后是否应在后台对它们进行比较。

默认值为 "yes"。

#### EMPTY\_\*

为前后快照相同的快照对定义清理算法。有关详细信息，请参见 [第 10.7.3 节 “清理无差异的快照对”](#)。

#### FSTYPE

分区的文件系统类型。不更改。

默认值为 "btrfs"。

#### NUMBER\_\*

定义可在安装和管理快照时使用的清理算法。有关详细信息，请参见 [第 10.7.1 节 “清理编号快照”](#)。

#### QGROUP / SPACE\_LIMIT

将配额支持添加到清理算法。有关详细信息，请参见 [第 10.7.5 节 “添加磁盘配额支持”](#)。

#### SUBVOLUME

分区或子卷生成快照的挂载点。不更改。

默认值为 "/"。

#### SYNC\_ACL

普通用户如要使用 Snapper (请参见 [第 10.5.1.2 节 “以普通用户身份使用 Snapper”](#))，他们必须能访问 `.snapshot` 目录，并且能读取其中的文件。如果 SYNC\_ACL 设置为 `yes`，Snapper 会通过 ACL 自动允许 ALLOW\_USERS 和 ALLOW\_GROUPS 项指定的用户和组访问这些目录及其中的文件。

默认值为 "no"。

#### TIMELINE\_CREATE

如果设置为 `yes`，便会每小时创建一个快照。有效值: `yes`、`no`。

默认值为 "no"。

#### TIMELINE\_CLEANUP / TIMELINE\_LIMIT\_\*

为时间线快照定义清理算法。有关详细信息，请参见 [第 10.7.2 节 “清理时间线快照”](#)。

## 10.5.1.2 以普通用户身份使用 Snapper

默认情况下，Snapper 只能由 `root` 用户使用。但在特定情况下，某些组或用户也需要创建快照或通过还原至快照来撤销更改：

- 想要建立 `/srv/www` 快照的网站管理员
- 想要为自己的主目录生成快照的用户

要实现这些目的，您可以创建用于为用户和/或组授予权限的 Snapper 配置。指定的用户必须能读取并访问相应的 `.snapshots` 目录。要实现这一点，最简单的方法是将 `SYNC_ACL` 选项设置为 `yes`。

### 过程 10.5：让普通用户可以使用 SNAPPER

此过程中的所有步骤都需要由 `root` 用户运行。

1. 如果 Snapper 配置尚不存在，请为用户可在其上使用 Snapper 的分区或子卷创建一个 Snapper 配置。有关指导，请参见[第 10.5 节 “创建并修改 Snapper 配置”](#)。示例：

```
> sudo snapper --config web_data create /srv/www
```

2. 在 `/etc/snapper/configs/CONFIG` 下创建配置文件，其中 “`CONFIG`” 是您在上一步中使用 `-c/--config` 指定的值（例如 `/etc/snapper/configs/web_data`）。根据需要进行调整。有关详细信息，请参见[第 10.5.1 节 “管理现有配置”](#)。
3. 设置 `ALLOW_USERS` 和/或 `ALLOW_GROUPS` 的值，以分别为用户和/或组授予权限。多个条目需要使用 `Space` 分隔。例如，要为用户 `www_admin` 授予权限，可运行：

```
> sudo snapper -c web_data set-config "ALLOW_USERS=www_admin"  
SYNC_ACL="yes"
```

4. 此时，指定的用户和（或）组便可以使用指定的 Snapper 配置。您可以使用 `list` 命令对其进行测试，例如：

```
www_admin:~ > snapper -c web_data list
```

## 10.6 手动创建和管理快照

Snapper 的功能并不仅限于根据配置自动创建和管理快照；您还可以使用命令行工具或 YaST 模块手动创建快照对（“前快照和后快照”）或单一快照。

所有 Snapper 操作皆针对现有配置执行（有关详细信息，请参见第 10.5 节“创建并修改 Snapper 配置”）。您可以只为存在配置的分区或卷生成快照。默认情况下使用系统配置 (root)。要为自己的配置创建或管理快照，则需要明确选择您的配置。使用 YaST 中的当前配置下拉框，或在命令行上指定 `-c (snapper -c MYCONFIG COMMAND)`。

### 10.6.1 快照元数据

每个快照均由快照本身和特定的元数据组成。创建快照时，您还需要指定元数据。修改快照就意味着更改其元数据——您无法修改其内容。使用 `snapper list` 可显示现有快照及其元数据：

#### `snapper --config home list`

列出配置 `home` 的快照。要列出默认配置 (root) 的快照，请使用 `snapper -c root list` 或 `snapper list`。

#### `snapper list -a`

列出所有现有配置的快照。

#### `snapper list -t pre-post`

列出默认 (root) 配置的所有前快照和后快照对。

#### `snapper list -t single`

列出默认 (root) 配置的所有 `single` 类型的快照。

每一张快照可以使用以下元数据：

- **类型**：快照类型。有关细节，请参见第 10.6.1.1 节“快照类型”。不能更改此数据。
- **编号**：快照的唯一编号。不能更改此数据。
- **前快照编号**：指定相应前快照的编号。仅适用于后类型。不能更改此数据。

- **说明**: 快照的说明。
- **用户数据**: 扩展的说明。您可使用逗号分隔的“键=值”列表格式指定自定义数据: `reason=testing, project=foo`。此字段也可用于将快照标记为重要 (`important=yes`) 以及列出创建快照的用户 (`user=tux`)。
- **清理算法**: 快照的清理算法。有关细节, 请参见[第 10.7 节 “快照的自动清理”](#)。

### 10.6.1.1 快照类型

Snapper 能够分清三种不同类型的快照: 前快照、后快照以及单一快照。从物理上讲, 这三种快照没有什么不同, 但 Snapper 会针对不同类型采用不同的处理方式。

#### pre

修改前的文件系统快照。每个 `pre` 快照对应于一个 `post` 快照。例如, 用于自动创建 YaST/Zypper 快照。

#### post

修改后的文件系统快照。每个 `post` 快照对应于一个 `pre` 快照。例如, 用于自动创建 YaST/Zypper 快照。

#### single

独立的快照。例如, 用于每小时自动创建快照。此为创建快照时的默认类型。

### 10.6.1.2 清理算法

Snapper 提供有三种清理旧快照的算法。这些算法在日常的 `cron` 作业中执行。您可以定义要在 Snapper 配置中保留的不同类型的快照数 (有关细节, 请参见[第 10.5.1 节 “管理现有配置”](#))。

#### **number**

当达到某一快照计数时将删除旧快照。

#### **timeline**

删除超过一定时限的旧快照, 但会保留若干每小时、每天、每月和每年快照。

## empty-pre-post

将删除无差异的前后快照对。

### 10.6.2 创建快照

要创建快照，请运行 **snapper create**，或者在 YaST 模块 Snapper 中单击创建。以下示例解释了如何从命令行创建快照。本文档未明确介绍 Snapper 的 YaST 界面，它所提供的功能相同。



#### 提示：快照说明

为了便于日后确定快照的用途，请始终指定有意义的说明。您还可以通过 **--userdata** 选项指定其他信息。

**snapper create --from 17 --description "with package2"**

从现有的快照（由 **snapper list** 返回的快照编号指定）创建独立的快照（单一类型）。（这适用于 Snapper 0.8.4 和更高版本。）

**snapper create --description "Snapshot for week 2 2014"**

创建默认 (**root**) 配置的独立快照（单一类型）并附加说明。因为没有指定清理算法，将不会自动删除快照。

**snapper --config home create --description "Cleanup in ~tux"**

为名为 **home** 的自定义配置创建独立快照（单一类型）并附加说明。因为没有指定清理算法，将不会自动删除快照。

**snapper --config home create --description "Daily data backup" --cleanup-algorithm timeline>**

为名为 **home** 的自定义配置创建独立快照（单一类型）并附加说明。一旦符合为配置中的时间线清理算法指定的条件，便会自动删除快照。

**snapper create --type pre --print-number --description "Before the Apache config cleanup" --userdata "important=yes"**

创建 **pre** 类型的快照并列显快照编号。创建用于保存“之前”和“之后”状态的快照对所需的首个命令。该快照标记为重要。

```
snapper create --type post --pre-number 30 --description "After the  
Apache config cleanup" --userdata "important=yes"
```

创建 `post` 类型的快照，与快照编号为 `30` 的 `pre` 快照配对。创建用于保存“之前”和“之后”状态的快照对所需的第二个命令。该快照标记为重要。

```
snapper create --command COMMAND --description "Before and after  
COMMAND"
```

运行 `COMMAND` 前后自动创建快照对。此选项仅在于命令行上使用 `snapper` 时可用。

### 10.6.3 修改快照元数据

Snapper 允许您修改说明、清理算法及快照的用户数据。其他元数据均无法更改。以下示例解释了如何从命令行修改快照。使用 YaST 界面会比较简单。

要在命令行上修改快照，您需要知道其编号。使用 `snapper list` 可以显示所有快照及其编号。

YaST 的 Snapper 模块已列出所有快照。从列表中选择一个快照，然后单击修改。

```
snapper modify --cleanup-algorithm "timeline" 10
```

修改默认 (`root`) 配置的第 10 张快照的元数据。清理算法设置为 `timeline`。

```
snapper --config home modify --description "daily backup" -cleanup-  
algorithm "timeline" 120
```

修改名为 `home` 的自定义配置的第 120 张快照的元数据。将设置新的说明并取消设置清理算法。

### 10.6.4 删除快照

要使用 YaST 的 Snapper 模块删除快照，请从列表中选择快照，然后单击删除。

要使用命令行工具删除快照，需要知道其编号。运行 `snapper list` 命令可以获取该编号。要删除快照，请运行 `snapper delete NUMBER`。

不允许删除当前的默认子卷快照。

使用 Snapper 删除快照时，在后台运行的 Btrfs 进程将会回收释放的空间。因此，可用空间的可见性与可用性会延迟。如果您希望通过删除快照释放的空间立即可供使用，请将选项 `--sync` 与 `delete` 命令使用。



## 提示：删除快照对

每次删除 `pre` 快照时，也应删除与其对应的 `post` 快照（反之亦然）。

### **`snapper delete 65`**

删除默认 (`root`) 配置的第 65 张快照。

### **`snapper -c home delete 89 90`**

删除名为 `home` 的自定义配置的第 89 张和第 90 张快照。

### **`snapper delete --sync 23`**

删除默认 (`root`) 配置的快照 23，并使释放的空间立即可用。



## 提示：删除未参照的快照

有时，虽然 Btrfs 快照存在，但却缺少包含 Snapper 元数据的 XML 文件。这种情况表示快照对 Snapper 不可见，需要手动将其删除：

```
btrfs subvolume delete /.snapshots/SNAPSHOTNUMBER/snapshot
rm -rf /.snapshots/SNAPSHOTNUMBER
```



## 提示：旧快照占用的磁盘空间更多

如果您要删除快照以释放硬盘上的空间，请务必先删除旧快照。快照生成的时间越长，其占用的空间就越大。

也可以通过日常的 cron 作业自动删除快照。有关细节，请参考[第 10.6.1.2 节 “清理算法”](#)。

## 10.7 快照的自动清理

快照会占用磁盘空间，随着时间的推移，快照占用的磁盘空间可能会变得非常多。为了防止磁盘上的空间耗尽，Snapper 提供了用于自动删除旧快照的算法。这些算法根据时间线快照和编号快照（管理快照与安装快照对）而有所不同。您可以指定要为每种类型保留的快照数。

此外，您也可以选择指定一个磁盘空间配额，定义快照可占用的最大磁盘空间大小。系统还可以自动删除前快照与后快照没有任何不同的快照对。

清理算法始终绑定到单个 Snapper 配置，因此您需要为每个配置指定算法。要防止自动删除特定的快照，请参见[问](#)。

默认设置 (root) 配置为清理编号快照以及空的前快照与后快照对。已启用配额支持 - 快照占用的空间不可超过根分区可用磁盘空间的 50%。时间线快照默认处于禁用状态，因此，时间线清理算法也处于禁用状态。

### 10.7.1 清理编号快照

编号快照（管理快照与安装快照对）的清理由 Snapper 配置的以下参数控制。

#### NUMBER\_CLEANUP

启用或禁用安装快照与管理快照对的清理。如果启用该参数，则当快照总数超过 NUMBER\_LIMIT 和/或 NUMBER\_LIMIT\_IMPORTANT 指定的数字以及 NUMBER\_MIN\_AGE 指定的时限时，系统会删除快照对。有效值：yes（启用）、no（禁用）。默认值为 "yes"。

用于更改或设置值的示例命令：

```
> sudo snapper -c CONFIG set-config "NUMBER_CLEANUP=no"
```

#### NUMBER\_LIMIT / NUMBER\_LIMIT\_IMPORTANT

定义要保留多少个普通和/或重要安装快照与管理快照对。如果 NUMBER\_CLEANUP 设置为 "no"，则会忽略此参数。

NUMBER\_LIMIT 的默认值为 "2-10"，NUMBER\_LIMIT\_IMPORTANT 的默认值为 "4-10"。该清理算法会删除超出指定最大值的快照，而不考虑快照和文件系统空间。该算法还会删除超出最小值的快照，直到达到快照和文件系统数量限制。

用于更改或设置值的示例命令：

```
> sudo snapper -c CONFIG set-config "NUMBER_LIMIT=10"
```



## 重要：范围值与常量值的比较

如果启用配额支持（请参见第 10.7.5 节“添加磁盘配额支持”），则需要将限制指定为最小值-最大值范围，例如 2-10。如果禁用配额支持，则需要提供常量值，例如 10，否则清理将会失败并返回错误。

### NUMBER\_MIN\_AGE

定义快照在自动删除前必须保留的最小时限（以秒为单位）。保留时间小于此处指定值的快照不会删除，不管这样的快照有多少。

默认值为 "1800"。

用于更改或设置值的示例命令：

```
> sudo snapper -c CONFIG set-config "NUMBER_MIN_AGE=864000"
```



## 注意：限制和时限

系统始终会评估 NUMBER\_LIMIT、NUMBER\_LIMIT\_IMPORTANT 和 NUMBER\_MIN\_AGE。只有同时符合全部条件才会删除快照。

如果您希望始终保留 NUMBER\_LIMIT\* 所定义数量的快照，而不考虑它们的时限，可将 NUMBER\_MIN\_AGE 设置为 0。

下面的示例显示了保留最近 10 个重要和 10 个普通快照（不论保留期限）的配置：

```
NUMBER_CLEANUP=yes
NUMBER_LIMIT_IMPORTANT=10
NUMBER_LIMIT=10
NUMBER_MIN_AGE=0
```

如果不想保留超过特定时限的快照，可将 NUMBER\_LIMIT\* 设置为 0，并用 NUMBER\_MIN\_AGE 指定时限。

下面的示例显示了只保留十天以内的快照的配置：

```
NUMBER_CLEANUP=yes
```

```
NUMBER_LIMIT_IMPORTANT=0
NUMBER_LIMIT=0
NUMBER_MIN_AGE=864000
```

## 10.7.2 清理时间线快照

时间线快照的清理由 Snapper 配置的以下参数控制。

### TIMELINE\_CLEANUP

启用或禁用时间线快照的清理。如果启用该参数，则当快照总数超过 TIMELINE\_LIMIT\_\* 指定的数字以及 TIMELINE\_MIN\_AGE 指定的时限时，将删除快照。有效值：yes、no。  
默认值为 "yes"。

用于更改或设置值的示例命令：

```
> sudo snapper -c CONFIG set-config "TIMELINE_CLEANUP=yes"
```

TIMELINE\_LIMIT\_DAILY, TIMELINE\_LIMIT\_HOURLY, TIMELINE\_LIMIT\_MONTHLY,  
TIMELINE\_LIMIT\_WEEKLY, TIMELINE\_LIMIT\_YEARLY

快照保留的时长，分别以小时、天、月、周和年为单位。

每项的默认值均为 "10"，但 TIMELINE\_LIMIT\_WEEKLY 除外，该参数默认设置为 "0"。

### TIMELINE\_MIN\_AGE

定义快照在自动删除前必须保留的最小时限（以秒为单位）。

默认值为 "1800"。

#### 例 10.1：时间线配置示例

```
TIMELINE_CLEANUP="yes"
TIMELINE_CREATE="yes"
TIMELINE_LIMIT_DAILY="7"
TIMELINE_LIMIT_HOURLY="24"
TIMELINE_LIMIT_MONTHLY="12"
TIMELINE_LIMIT_WEEKLY="4"
TIMELINE_LIMIT_YEARLY="2"
```

```
TIMELINE_MIN_AGE="1800"
```

此示例配置能够实现按小时生成将自动清理的快照。`TIMELINE_MIN_AGE` 和 `TIMELINE_LIMIT_*` 始终会一起评估。在本示例中，快照删除前的最小保留时限设置为 30 分钟（1800 秒）。因为我们会每小时创建一次快照，所以确保了只会保留最近的快照。如果 `TIMELINE_LIMIT_DAILY` 设置为非零值，则表示还会保留当天的首张快照。

#### 保留的快照

- 每小时：创建的最后 24 个快照。
- 每天：保留最近 7 天内每天创建的首个快照。
- 每月：保留最近 12 个月内每月最后一天创建的首个快照。
- 每周：保留最近 4 周内每周最后一天创建的首个快照。
- 每年：保留最近 2 年内每年最后一天创建的首个快照。

### 10.7.3 清理无差异的快照对

如第 10.1.2 节 “快照类型” 中所述，每当您运行 YaST 模块或执行 Zypper 时，将在启动时创建一个前快照，在退出时创建一个后快照。如果您未进行过任何更改，则前快照与后快照没有差异。在 Snapper 配置中设置以下参数可自动删除此类 “空” 快照对：

#### EMPTY\_PRE\_POST\_CLEANUP

如果设置为 yes，系统会删除前后快照相同的快照对。

默认值为 "yes"。

#### EMPTY\_PRE\_POST\_MIN\_AGE

定义前后快照相同的快照对在自动删除之前必须保留的最短时限（以秒为单位）。

默认值为 "1800"。

### 10.7.4 清理手动创建的快照

Snapper 未针对手动创建的快照提供自定义清理算法。但是，您可以向手动创建的快照指派 `number` 或 `timeline` 清理算法。如此，该快照将加入您所指定的算法的 “清理队列”。可以在创建快照时或通过修改现有快照来指定清理算法：

```
snapper create --description "Test" --cleanup-algorithm number
```

为默认 (root) 配置创建独立快照 (单一类型) 并指派 number 清理算法。

```
snapper modify --cleanup-algorithm "timeline" 25
```

使用数字 25 修改快照, 并指派 timeline 清理算法。

## 10.7.5 添加磁盘配额支持

除了上述 number 和/或 timeline 清理算法外, Snapper 还支持配额。您可以定义允许快照占用的可用空间百分比。此百分比值始终适用于相应 Snapper 配置中定义的 Btrfs 子卷。

Btrfs 配额将应用到子卷, 而非应用到用户。除了使用 Btrfs 配额之外, 还可以将磁盘空间配额应用到用户和组 (例如, 使用 quota 命令)。

如果在安装期间启用了 Snapper, 则会自动启用配额支持。如果您是在安装后的某个时间手动启用 Snapper 的, 则可以通过运行 snapper setup-quota 来启用配额支持。这需要您具有有效的配置 (有关详细信息, 请参见[第 10.5 节 “创建并修改 Snapper 配置”](#) )。

配额支持由 Snapper 配置的以下参数控制。

### QGROUP

Snapper 使用的 Btrfs 配额组。如果未设置, 请运行 snapper setup-quota。如果已设置, 则仅在您熟悉 man 8 btrfs-qgroup 的情况下才可对其进行更改。此值是使用 snapper setup-quota 设置的, 不应更改。

### SPACE\_LIMIT

允许快照使用的空间限制, 以 1 (100%) 的分数表示。有效值范围为 0 到 1 (0.1 = 10%, 0.2 = 20%...)。

需遵守以下限制和指导原则:

- 只能在已激活现有 number 和/或 timeline 清理算法的前提下才能激活配额。如果未激活任何清理算法，则无法应用配额限制。
- 启用配额支持后，Snapper 将根据需要执行两轮清理。第一轮清理将应用针对编号快照和时间线快照指定的规则。仅当完成这一轮清理后超出配额时，在第二轮清理中才会应用配额特定的规则。
- 即使启用了配额支持，Snapper 也始终会保留 NUMBER\_LIMIT\* 和 TIMELINE\_LIMIT\* 值所指定数量的快照，而不考虑是否会超出配额。因此，建议为 NUMBER\_LIMIT\* 和 TIMELINE\_LIMIT\* 指定范围值 (MIN-MAX)，以确保配额可以生效。  
例如，如果设置了 NUMBER\_LIMIT=5-20，Snapper 将执行第一轮清理，并将普通的编号快照数量减至 20 个。如果这 20 个快照超出配额，Snapper 将在第二轮清理中删除最旧的快照，直到符合配额限制。系统始终会至少保留 5 个快照，不管这些快照占用了多少空间。

## 10.8 显示快照使用的独占磁盘空间

快照会共享数据以有效利用存储空间，因此，使用 du 和 df 等普通命令无法准确度量已用的磁盘空间。想要在启用了配额的 Btrfs 上释放磁盘空间，需要知道每个快照使用的独占磁盘空间量，而不是共享的空间量。Snapper 0.6 和更高版本会在 Used Space 列中报告每个快照使用的磁盘空间：

```
# snapper--iso list
# | Type      | Pre # | Date                  | User | Used Space | Cleanup |
Description          | Userdata
-----+-----+-----+-----+-----+-----+-----+
+-----+-----+
0  | single   |      |                      | root |          |          |
current             |
1* | single   |      | 2019-07-22 13:08:38 | root | 16.00 KiB |          |
root filesystem |
2  | single   |      | 2019-07-22 14:21:05 | root | 14.23 MiB | number   | after
installation      | important=yes
3  | pre      |      | 2019-07-22 14:26:03 | root | 144.00 KiB | number   |
zypp(zypper)        | important=no
```

```

4 | post | 3 | 2019-07-22 14:26:04 | root | 112.00 KiB | number |
| important=no
5 | pre | | 2019-07-23 08:19:36 | root | 128.00 KiB | number |
zypp(zypper) | important=no
6 | post | 5 | 2019-07-23 08:19:43 | root | 80.00 KiB | number |
| important=no
7 | pre | | 2019-07-23 08:20:50 | root | 256.00 KiB | number | yast
sw_single |
8 | pre | | 2019-07-23 08:23:22 | root | 112.00 KiB | number |
zypp(ruby.ruby2.5) | important=no
9 | post | 8 | 2019-07-23 08:23:35 | root | 64.00 KiB | number |
| important=no
10 | post | 7 | 2019-07-23 08:24:05 | root | 16.00 KiB | number |
|

```

**btrfs** 命令提供快照已使用空间的另一种视图：

```

# btrfs qgroup show -p /
qgroupid      rfer      excl parent
-----  -----
0/5          16.00KiB    16.00KiB  ---
[...]
0/272        3.09GiB    14.23MiB 1/0
0/273        3.11GiB    144.00KiB 1/0
0/274        3.11GiB    112.00KiB 1/0
0/275        3.11GiB    128.00KiB 1/0
0/276        3.11GiB    80.00KiB  1/0
0/277        3.11GiB    256.00KiB 1/0
0/278        3.11GiB    112.00KiB 1/0
0/279        3.12GiB    64.00KiB  1/0
0/280        3.12GiB    16.00KiB  1/0
1/0          3.33GiB    222.95MiB ---

```

qgroupid 列显示每个子卷的标识号，并指派 qgroup 级别/ID 的组合。

rfer 列显示子卷中引用的数据总量。

excl 列显示每个子卷中的独占数据。

parent 列显示子卷的父 qgroup。

最后一个项目 `1/0` 显示父 qgroup 的总计。在上面的示例中，如果去除所有子卷，将释放 222.95 MiB 空间。运行以下命令查看与每个子卷关联的快照：

```
# btrfs subvolume list -st /
ID gen top level path
-----
267 298 266 @/.snapshots/1/snapshot
272 159 266 @/.snapshots/2/snapshot
273 170 266 @/.snapshots/3/snapshot
274 171 266 @/.snapshots/4/snapshot
275 287 266 @/.snapshots/5/snapshot
276 288 266 @/.snapshots/6/snapshot
277 292 266 @/.snapshots/7/snapshot
278 296 266 @/.snapshots/8/snapshot
279 297 266 @/.snapshots/9/snapshot
280 298 266 @/.snapshots/10/snapshot
```

将一个服务包升级到另一个服务包会导致快照占用系统子卷上的大量磁盘空间。因此对于不再需要的快照，建议手动删除。有关详细信息，请参见 [第 10.6.4 节 “删除快照”](#)。

## 10.9 常见问题解答

**问：**为何 Snapper 从不显示 `/var/log`、`/tmp` 和其他目录中的更改？

**答：**因为我们将特定目录排除在了快照之外。详细的列表和具体原因请参见 [第 10.1.3 节 “从快照中排除的目录”](#)。为了将路径从快照中排除，我们为该路径创建了子卷。

**问：**我可以从引导加载程序引导快照吗？

**答：**可以。有关细节，请参考 [第 10.3 节 “通过从快照引导来执行系统回滚”](#)。

**问：**可以防止删除快照吗？

**答：**目前，Snapper 尚无防止手动删除快照的功能。不过，您可以防止清理算法自动删除快照。系统不会为手动创建的快照（请参见 [第 10.6.2 节 “创建快照”](#)）指派清理算法，除非您使用 `--cleanup-algorithm` 来指定。自动创建的快照始终会被指派 `number` 或 `timeline` 算法。要从一个或多个快照去除此类指派，请执行以下操作：

1. 列出所有可用快照：

```
> sudo snapper list -a
```

2. 记住您要防止删除的快照数。
3. 运行以下命令并将数字占位符替换为您记住的数字：

```
> sudo snapper modify --cleanup-algorithm "" #1 #2 #n
```

4. 再次运行 **snapper list -a** 检查结果。在 Cleanup 列中，您修改的快照所对应的项现在应该为空。

**问：** 何处能获得有关 Snapper 的详细信息？

**答：** 请访问 Snapper 的主页，网址为：<http://snapper.io/>。

# 11 使用 KLP 的在线内核修补

本文档介绍内核实时修补 (KLP) 技术的基本原理，并提供 SLE Live Patching 服务的使用准则。

KLP 让您无需重引导便可应用 Linux 内核的最新安全性更新。如此可最大限度地提高系统正常运行时间和可用性，这对于任务关键型系统而言尤为重要。

本文档中提供的信息与 AMD64/Intel 64、POWER 和 IBM Z 体系结构相关。

## 11.1 内核实时修补的优势

KLP 提供了多项优势。

- 对于需获得或维护特定合规认证的组织而言，确保大量服务器自动保持最新状态至关重要。KLP 可在帮助实现合规的同时，减少对费用高昂的维护期的需求。
- 签订了服务级别协议合同的公司必须保证其系统可用性和正常运行时间达到特定的级别。实时修补可在不产生停机时间的情况下修补系统。
- 由于 KLP 是标准系统更新机制的一部分，因此无需专门培训或引入复杂的维护例程。

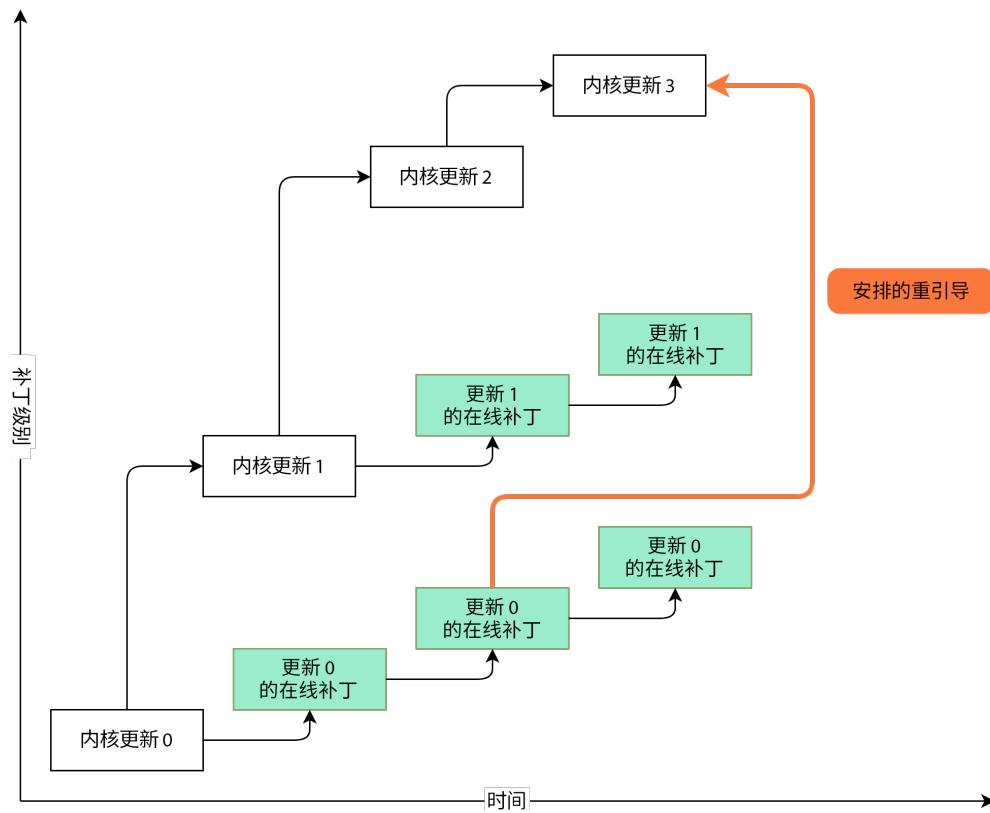
## 11.2 内核实时修补概述

内核实时补丁通过内含经过修改的代码的软件包提供，此类软件包与主内核软件包不同。实时补丁具有累积性，因此最新的补丁包含内核软件包先前补丁的所有修复。每个内核在线软件包都与为其发布的确切内核修订版关联。每新增一个修复，实时补丁软件包版本号都会相应增加。

### ! 重要：实时补丁和内核更新的对比

实时补丁仅包含关键修复，不会替代常规的内核更新，后者需要重引导系统。可以将实时补丁视为在执行适当的内核更新和重引导前用于暂时保护内核的措施。

下图阐述了实时补丁与内核更新之间的总体关系。使用 `klp -v patches` 命令可以查看当前有效的实时补丁所解决的 CVE 列表和缺陷报告。



您可以安装多个版本的内核软件包及其实时补丁。这些软件包不会冲突。您可以为正在运行的内核安装更新的内核软件包及实时补丁。在此情况下，您可能会收到重引导系统的提示。订阅了 SLE Live Patching 的用户有权获享技术支持，只要正在运行的内核存在实时补丁更新（请参见第 11.5.1 节 “检查实时补丁的失效日期”）。

激活 KLP 后，每个内核更新都会附带实时补丁软件包。此实时补丁不包含任何修复，而是充当相应内核未来实时补丁的种子。这些空种子补丁称为 `initial patches`。

### 11.2.1 内核实时修补范围

SLE Live Patching 的修补范围包括 SUSE 通用漏洞评分系统 (CVSS; SUSE CVSS 基于 CVSS v3.1 系统开发) 7 级以上系统稳定或数据损害相关漏洞和 bug 修复。但从技术上而言，为属于指定类别的所有修复都创建实时补丁可能并不现实。因此，SUSE 保留在因技术原因而无法创

建内核实时补丁的情况下跳过相应修复的权利。目前超过 95% 的合格修复都是以实时补丁的形式发布的。有关 CVSS (SUSE CVSS 评分基础) 的详细信息, 请参见 [Common Vulnerability Scoring System SIG \(https://www.first.org/cvss/\)](https://www.first.org/cvss/)。

## 11.2.2 内核实时修补局限性

KLP 涉及替换功能和妥善处理相互依赖的功能集的替换。这通过将旧代码调用重定向到不同内存位置的更新代码来完成。数据结构的变化使情况变得更加复杂, 因为数据会保持原状, 无法扩展或重新解译。尽管使用某些技术可以间接更改数据结构, 但某些修复无法转换为实时补丁。在此情况下, 只能通过系统重启动来应用修复。

## 11.3 使用 YaST 激活内核实时修补

要在您的系统上激活 KLP, 您需要拥有有效的 SLES 和 SLE Live Patching 订阅。请访问 [SUSE Customer Center \(https://scc.suse.com/\)](https://scc.suse.com/) 检查您的订阅的状态, 并获取 SLE Live Patching 订阅的注册代码。

要在您的系统上激活内核实时修补, 请执行以下步骤:

1. 运行 `yast2 registration` 命令并单击选择扩展。
2. 在可用扩展列表中选择 SUSE Linux Enterprise Live Patching 15, 然后单击下一步。
3. 确认许可条款并单击下一步。
4. 输入您的 SLE Live Patching 注册代码并单击下一步。
5. 检查安装摘要和选定的软件集。系统应该会自动选择安装 Live Patching 和 SLE Live Patching Lifecycle Data 软件集以及一些其他软件包, 以满足依赖关系。
6. 单击接受完成安装。这样会在您的系统上安装基础内核实时修补组件、初始实时补丁, 以及所需的依赖项。

## 11.4 从命令行激活内核实时修补

要激活内核实时修补，您需要拥有有效的 SLES 和 SLE Live Patching 订阅。请访问 [SUSE Customer Center \(https://scc.suse.com/\)](https://scc.suse.com/) 查看您的订阅的状态，并获取 SLES Live Patching 订阅的注册代码。

1. 运行 `sudo SUSEConnect --list-extensions`. 请记住 SLES Live Patching 的确切激活命令。示例命令输出（缩略版）：

```
$ SUSEConnect --list-extensions
...
SUSE Linux Enterprise Live Patching 15 SP5 x86_64
Activate with: SUSEConnect -p sle-module-live-patching/15.5/x86_64 \
-r ADDITIONAL REGCODE
```

2. 使用获得的命令后跟 `-r LIVE_PATCHING_REGISTRATION_CODE` 激活 SLES Live Patching，例如：

```
SUSEConnect -p sle-module-live-patching/15.5/x86_64 \
-r LIVE_PATCHING_REGISTRATION_CODE
```

3. 使用 `zypper install -t pattern lp_sles` 命令安装所需的软件包和依赖项。

至此，系统已完成实时修补。

后台进行的过程如下：当软件包安装系统检测到某个安装的内核可以实时修补，并且软件通道中存在相应的实时补丁时，系统会选择安装该实时补丁。然后内核会在软件包安装过程中接收到实时补丁修复。在产品安装完成前，内核就会进行实时修补。

## 11.5 执行内核实时修补

内核实时补丁是在常规的系统更新过程中安装的。但您应了解以下几点。

- 如果已为正在运行的内核安装 `kernel-livepatch-*` 软件包，内核会进行实时修补。您可以使用 `zypper se --details kernel-livepatch-*` 命令来检查您的系统上安装了哪些内核实时补丁软件包。
- 如果安装了 `kernel-default` 软件包，更新管理器会提示您重引导系统。为避免此消息显示，您可以将内核更新从修补操作中筛选。这可以通过使用 Zypper 添加软件包锁来实现。您也可以使用 SUSE Manager 过滤通道内容（请参见 [Live Patching with SUSE Manager](https://documentation.suse.com/external-tree/en-us/suma/4.1/suse-manager/administration/live-patching.html) (<https://documentation.suse.com/external-tree/en-us/suma/4.1/suse-manager/administration/live-patching.html>) ）。
- 可以使用 `klp status` 命令来检查修补状态。要检查安装的补丁，请运行 `klp -v patches` 命令。
- 请记住，虽然系统上可能会安装多个内核软件包，但任何时间只有一个软件包在运行。同样，虽然系统上可能会安装多个实时补丁软件包，但只有一个实时补丁会装载到内核中。
- 有效的实时补丁包含在 `initrd` 中。这表示如果发生了意外重引导，系统启动时会应用实时补丁修复，因此不需要再次执行修补。

### 11.5.1 检查实时补丁的失效日期

请确保已安装 `lifecycle-data-sle-module-live-patching`，然后运行 `zypper lifecycle` 命令。在输出的 `Package end of support if different from product` 部分应该可以看到实时补丁的失效日期。

每个实时补丁自底层内核软件包发布起一年内都会收到更新。您可以使用 [Maintained kernels, patch updates and lifecycle](https://www.suse.com/products/live-patching/current-patches/) (<https://www.suse.com/products/live-patching/current-patches/>) 页面根据运行的内核版本检查失效日期，无需安装产品扩展。

## 11.6 内核实时修补问题查错

### 11.6.1 手动降级补丁

如果您发现了最新的实时补丁有问题，可以将当前安装的实时补丁降级回之前的版本。我们建议在系统开始显现出问题前执行补丁降级。请注意，在系统日志中跟踪内核警告或内核错误的系统可能不适合使用补丁降级过程。如果您不确定系统是否满足补丁降级的要求，请联系 SUSE 技术支持部门获得帮助。

#### 过程 11.1：手动降级补丁

1. 使用 **klp -v patches** 命令确定正在运行的实时补丁。以 **RPM:** 开头的行中显示的就是当前正在运行的补丁。例如：

```
RPM: kernel-livepatch-5_3_18-24_29-default-2-2.1.x86_64
```

上例中的 5\_3\_18-24\_29-default 指的就是正在运行的内核版本。

2. 使用 **zypper search -s kernel-livepatch-RUNNING\_KERNEL\_VERSION-default** 命令搜索补丁的先前版本。该命令会返回可用软件包版本列表。请注意，每个新实时补丁软件包发行版的版本号会加 1。请务必选择版本号低于当前版本号的发行版。
3. 使用 **zypper in --oldpackage kernel-livepatch-RUNNING\_KERNEL\_VERSION-default=DESIRED\_VERSION** 命令安装所需的版本。

# 12 用户空间实时修补

本章介绍用户空间实时修补的基本原理和用法。

## 12.1 关于用户空间实时修补

用户空间实时修补 (ULP) 是指在不中断正在运行的进程的情况下，将补丁应用到该进程所使用的库的过程。每次以实时补丁的形式提供安全修复程序时，在应用实时补丁后，客户服务的安全性即会得到保障，而无需重启相关进程。

实时修补操作使用作为 `libpulp` 的一部分的 `ulp` 工具来执行。`libpulp` 是一个由 `libpulp.so` 库和 `ulp` 二进制文件组成的框架。该二进制文件用于使库可实时修补以及应用实时补丁。



### 提示

您可以通过 `sudo` 机制以普通用户或特权用户身份运行 `ulp` 命令。区别在于通过 `sudo` 运行 `ulp`，您可以查看 `root` 所运行的进程或修补进程的信息。

### 12.1.1 先决条件

要使 ULP 正常工作，必须满足两个要求。

- 运行以下命令在系统上安装 ULP：

```
> sudo zypper in libpulp0 libpulp-tools
```

- 必须在加载 `libpulp.so.0` 库之前启动具有所需实时修补支持的应用程序。有关详细信息，请参见第 12.1.3 节 “使用 `libpulp`”。

## 12.1.2 支持的库

当前，仅支持 `glibc` 和 `openssl (openssl1_1)`。其他软件包将在它们准备好实时修补后可用。要接收 `glibc` 和 `openssl` 实时补丁，请安装 `glibc-livepatches` 和 `openssl-livepatches` 软件包：

```
> zypper install glibc-livepatches openssl-livepatches
```

## 12.1.3 使用 `libpulp`

要实现对某个应用程序的实时修补，您需要在启动该应用程序时预装载 `libpulp.so.0` 库：

```
> LD_PRELOAD=/usr/lib64/libpulp.so.0 APPLICATION_CMD
```

### 12.1.3.1 检查库是否可实时修补

要检查库是否可实时修补，请使用以下命令：

```
> ulp livepatchable PATH_TO_LIBRARY
```

### 12.1.3.2 检查 `.so` 文件是否为实时补丁容器

如果共享对象 (`.so`) 包含嵌入其中的 ULP 补丁描述，则表示其为实时补丁容器。您可以使用以下命令验证它：

```
> readelf -S SHARED_OBJECT | grep .ulp
```

如果输出显示共享对象中有 `.ulp` 和 `.ulp.rev` 部分，则表示其为实时补丁容器。

### 12.1.3.3 应用实时补丁

可以使用 `ulp trigger` 命令应用实时补丁，例如：

```
> ulp trigger -p PID LIVEPATCH.so
```

以正在运行的进程（使用了要修补的库）的进程 ID 替换 PID，以实际的实时补丁文件替换 LIVEPATCH.so。该命令会返回以下状态消息之一：

### SUCCESS

实时修补操作成功。

### SKIPPED

已跳过该补丁，因为它不适用于进程中装载的任何库。

### ERROR

发生错误，您可以检查 libpulp 内部消息缓冲区以获得更多信息。有关更多信息，请参见第 12.1.3.6 节“[查看内部消息队列](#)”。

您也可以使用通配符来应用多个实时补丁，例如：

```
> ulp trigger '*.so'
```

该命令会尝试将当前文件夹中的每个补丁应用到已装载 libpulp 库的每个进程。如果补丁不适合该进程，则会自动跳过该补丁。最后，该工具会显示它成功地将多少个补丁应用到多少个进程。

### 12.1.3.4 还原实时补丁

您可以使用 ulp trigger 命令还原实时补丁。可以通过两种方法还原实时补丁。您可以使用 --revert 开关传递实时补丁容器来还原实时补丁：

```
> ulp trigger -p PID --revert LIVEPATCH.so
```

或者，您也可以去除与特定库关联的所有补丁，例如：

```
> ulp trigger -p PID --revert-all=LIBRARY
```

在示例中，LIBRARY 指的是实际的库，例如 libcrypto.so.1.1。

当原始实时补丁的源代码不可用时，后一种方法可能很有用。或者，您希望在目标应用程序仍在运行安全代码时去除特定的旧补丁并应用新补丁，例如：

```
> ulp trigger -p PID --revert-all=libcrypto.so.1.1 new_livepatch2.so
```

### 12.1.3.5 查看应用的补丁

您可以运行以下命令来验证已为哪些应用程序应用实时补丁：

```
> ulp patches
```

输出会显示可实时修补的库和程序中装载的补丁，以及补丁解决的 bug：

```
PID: 10636, name: test
Livepatchable libraries:
  in /lib64/libc.so.6:
    livepatch: libc_livepatch1.so
      bug labels: jsc#SLE-0000
  in /usr/lib64/libpulp.so.0:
```

您还可以查看实时补丁修补的功能：

```
> ulp dump LIVEPATCH.so
```

### 12.1.3.6 查看内部消息队列

来自 libpulp.so 的日志消息存储在库内的缓冲区中，除非用户请求，否则不会显示。要显示这些消息，请运行：

```
> ulp messages -p PID
```

## 12.2 更多信息

项目的 [Git repository \(https://github.com/SUSE/libpulp\)](https://github.com/SUSE/libpulp) 中提供了有关 libpulp 的更多信息。

## 13 事务更新

事务更新在 SUSE Linux Enterprise Desktop 中以技术预览形式提供，当根文件系统处于只读模式时，可使用技术更新来更新 SLES。事务更新具有原子性（仅当所有更新都成功时，才会应用所有更新），且支持回滚。它不影响正在运行的系统，因为只有在重引导系统后，才会激活更改。由于重引导是中断性操作，管理员必须判断重引导的开销是否高于中断运行中服务的开销。如果重引导的开销过高，则不要使用事务更新。

事务更新由 **`transactional-update`** 脚本每天运行。该脚本将检查可用的更新。如果存在任何更新，则它会在后台创建根文件系统的新快照，然后从发布渠道提取更新。完全更新新快照后，该快照将标记为活动快照，并在下一次重引导系统后，成为新的默认根文件系统。当 **`transactional-update`** 设置为自动运行（默认行为）时，该脚本还会重引导系统。更新运行时间以及重引导维护时段均可配置。

只能更新属于根文件系统快照的软件包。如果软件包中包含不属于该快照的文件，更新可能会失败或中断系统。

无法更新需要接受许可证的 RPM。

### 13.1 技术预览的限制

技术预览的功能存在某些限制。**`transactional-update`** 对于以下软件包不起作用：

- `nginx` 的默认 `index.html` 页可能不可用
- `tomcat-webapps` 和 `tomcat-admin-webapps`
- `phpMyAdmin`
- `sca-appliance-*`
- `mpi-selector`
- `emacs` 可正常工作（Emacs 游戏除外）

- bind 和 bind-chrootenv
- docbook\*
- sblim-sfcb\*
- texlive\*
- iso\_ent
- openjade
- opensp
- pcp
- plymouth
- postgresql-server-10
- pulseaudio-gdm-hooks
- smartmontools

系统安装程序的更新程序组件不适用于只读文件系统，因为此类文件系统不支持事务更新。

其他注意事项：

- 最好在更新系统之后尽早重引导计算机。
- 每次只能应用一项更新。在应用一项更新之后到应用下一项更新之前，请务必重引导。
- 在事务更新之后直到将计算机重引导，都不应运行 update-alternatives。
- 在事务更新之后直到重引导，都不要创建新的系统用户或系统组。允许创建普通用户和组 (UID > 1000, GID > 1000)。
- YaST 尚不能识别事务更新。如果 YaST 模块需要安装额外的软件包，此操作将无法正常执行。而只是修改 /etc 中配置文件的常规系统操作可正常进行。
- 对于 php7-fastcgi，必须手动创建指向 /usr/bin/php-cgi 的符号链接 /srv/www/cgi-bin/php。

- ntp 是要从较低 SLES 版本迁移的 Legacy Module 的一部分。它在新的 SUSE Linux Enterprise Desktop 安装中不再受支持，已被 chrony 取代。如果您继续使用 ntp，需要进行全新安装才能正常使用事务更新。
- sblim-sfcb：整个 sblim 生态系统都与事务更新不兼容。
- btrfsmaintenance 软件包中的 **btrfs-defrag** 无法与只读根文件系统搭配使用。
- 对于 btrfs-balance, /etc/sysconfig/btrfsmaintenance 中的变量 BTRFS\_BALANCE\_MOUNTPOINTS 必须从 / 更改为 /.snapshots。
- 对于 btrfs-scrub, /etc/sysconfig/btrfsmaintenance 中的变量 BTRFS\_SCRUB\_MOUNTPOINTS 必须从 / 更改为 /.snapshots。

## 13.2 启用 transactional-update

必须在系统安装期间启用 Transactional Server 模块，然后选择 Transactional Server 系统角色。不支持稍后再于正在运行的系统中从 Transactional Server 模块安装任何软件包，此操作可能会使系统中断。

不支持更改根分区的子卷布局，或者将根分区的子目录或子卷置于各自的分区中（/home、/var、/srv 和 /opt 除外），此类操作很可能会使系统中断。

## 13.3 管理自动更新

自动更新由每天运行一次的 systemd.timer 控制。此操作会应用所有更新，并告知 rebootmgrd 应重引导计算机。您可以调整更新运行时间，具体请参见“systemd.timer(5)”。要调整维护时段（即 rebootmgrd 重引导系统的时间），请参见“rebootmgrd(8)”。

您可以使用以下命令禁用自动事务更新：

```
# systemctl --now disable transactional-update.timer
```

## 13.4 **transactional-update** 命令

**transactional-update** 命令会启用更新的原子性安装或去除，即仅当所有更新都可以成功安装时，才会应用这些更新。在应用更新之前，**transactional-update** 会创建系统的快照，您可以恢复此快照。重引导后，所有更改才会生效。

### --continue

使用 **--continue** 选项可在不重引导的情况下对现有快照进行多次更改。

默认的 **transactional-update** 行为是从当前根文件系统创建新快照。如果您忘记了某项操作，例如，忘记了安装某个新软件包，必须重引导来应用先前的更改，然后再次运行 **transactional-update** 以安装忘记安装的软件包，并再次重引导。不能在未进行重引导的情况下多次运行 **transactional-update** 命令来向快照添加更多更改，因为这样做只会创建多个独立的快照，这些快照并不会包含先前快照中的更改。

您可使用 **--continue** 选项在不重引导的情况下，进行任意次数的更改。每次会创建一个独立的快照，且每个快照都包含您在先前快照中进行的所有更改以及新的更改。视需要重复此过程任意次数，当最终快照包含所需的所有更改时，再重引导系统，然后最终快照便会成为新的根文件系统。

**--continue** 选项的另一个有用功能是，可让您选择任一现有快照作为新快照的基础快照。以下示例演示如何运行 **transactional-update** 在基于快照 13 的某个快照中安装新软件包，然后再次运行该命令来安装另一个软件包：

```
# transactional-update pkg install package_1
```

```
# transactional-update --continue 13 pkg install package_2
```

**--continue [num]** 选项会调用 **snapper create --from**，请参见第 10.6.2 节“[创建快照](#)”。

### cleanup

如果当前根文件系统与活动的根文件系统相同（在重引导之后，**transactional-update** 创建包含更新的新快照之前），将为没有清理算法的所有旧快照设置一个清理算法。如此可确保 Snapper 会删除旧快照（请参见“[snapper\(8\)](#)”中有关清理算法的章节）。此操作还会去除 **/var/lib/overlay** 中所有未引用（因此也未使用）的 **/etc** 覆盖目录：

```
# transactional-update cleanup
```

## pkg in/install

使用 **zypper install** 命令从可用渠道安装单个软件包。此命令还可用于安装程序临时修复 (PTF) RPM 文件。

```
# transactional-update pkg install package_name
```

或

```
# transactional-update pkg install rpm1 rpm2
```

## pkg rm/remove

使用 **zypper remove** 命令从活动快照中去除单个软件包。此命令还可用于去除 PTF RPM 文件。

```
# transactional-update pkg remove package_name
```

## pkg up/update

使用 **zypper update** 命令更新活动快照中的单个软件包。只能更新属于基础文件系统快照的软件包。

```
# transactional-update pkg update package_name
```

## up/update

如果有新的可用更新，将会创建一个新快照，并使用 **zypper up/update** 更新该快照。

```
# transactional-update up
```

## dup

如果有新的可用更新，将会创建一个新快照，并使用 **zypper dup --no-allow-vendor-change** 更新该快照。然后，该快照将会激活，并在重引导后成为新的根文件系统。

```
# transactional-update dup
```

## patch

如果有新的可用更新，将会创建一个新快照，并使用 **zypper patch** 更新该快照。

```
# transactional-update patch
```

## rollback

此选项会设置默认子卷。在使用读/写文件系统的系统上可调用 **snapper rollback**。在只读文件系统上，如果未提供任何参数，当前系统将设置为新的默认根文件系统。如果您指定了编号，则该快照将用作默认的根文件系统。在只读文件系统上，不会创建任何额外快照。

```
# transactional-update rollback snapshot_number
```

## grub.cfg

此参数会创建新的 GRUB2 配置。有时需要调整引导配置，例如，添加额外的内核参数。编辑 **/etc/default/grub**，运行 **transactional-update grub.cfg**，然后重引导以使更改生效。您必须立即重引导，否则，下一次运行 **transactional-update** 时会使用默认值重写新的 GRUB2 配置。

```
# transactional-update grub.cfg
```

## reboot

此参数在完成操作后会触发重引导。

```
# transactional-update dup reboot
```

## --help

此时会输出包含选项和子命令的帮助屏幕。

```
# transactional-update --help
```

## 13.5 查错

如果升级失败，请运行 **supportconfig** 来收集日志数据。将生成的文件（包括 **/var/log/transactional-update.log**）提供给 SUSE 支持人员。

# 14 使用 VNC 的远程图形会话

利用虚拟网络计算 (VNC) 可以通过图形桌面来访问远程计算机，并运行远程图形应用程序。VNC 与平台无关，可以从任何操作系统访问远程计算机。本章介绍如何使用桌面客户端 `vncviewer` 和 `Remmina` 连接到 VNC 服务器，以及如何操作 VNC 服务器。

SUSE Linux Enterprise Desktop 支持两种不同类型的 VNC 会话：一种是一次性会话，在客户端 VNC 连接持续期间处于“活跃”状态，另一种是永久会话，在被明确终止前始终处于“活跃”状态。

VNC 服务器可在不同端口上同时提供两种会话，但打开的会话不能从一种类型转换为另一种类型。

## 14.1 `vncviewer` 客户端

要连接到服务器提供的 VNC 服务，需要使用客户端。SUSE Linux Enterprise Desktop 中的默认客户端是 `tigervnc` 软件包提供的 `vncviewer`。

### 14.1.1 使用 `vncviewer` CLI 进行连接

要启动 VNC 查看器并发起与服务器的会话，请使用以下命令：

```
> vncviewer jupiter.example.com:1
```

若不使用 VNC 显示器编号，您也可以指定带两个冒号的端口号：

```
> vncviewer jupiter.example.com::5901
```



#### 注意：显示号和端口号

您在 VNC 客户端中指定的实际显示号或端口号必须与在目标计算机上通过 `vncserver` 命令提取的显示号或端口号相同。有关更多信息，请参见第 14.4 节“配置永久 VNC 服务器会话”。

## 14.1.2 使用 vncviewer GUI 进行连接

如果在不指定 `--listen` 或要连接的主机的情况下运行 `vncviewer`，它会显示一个窗口，要求您输入连接细节。按第 14.1.1 节 “使用 vncviewer CLI 进行连接” 中所述在 VNC 服务器字段中输入主机，然后单击连接。



图 14.1：VNCVIEWER

## 14.1.3 未加密连接通知

VNC 协议支持不同类型的加密连接，请不要将这些连接与口令身份验证相混淆。如果某个连接未使用 TLS，VNC 查看器的窗口标题中可能会出现 “（连接未加密！）” 文本。

## 14.2 Remmina：远程桌面客户端

Remmina 是功能丰富的新式远程桌面客户端。它支持多种访问方法，例如 VNC、SSH、RDP 和 Spice。

### 14.2.1 安装

要使用 Remmina，请校验系统上是否安装了 `remmina` 软件包，如未安装，请加以安装。记得还要安装适用于 Remmina 的 VNC 插件：

```
# zypper in remmina remmina-plugin-vnc
```

## 14.2.2 主窗口

通过输入 **remmina** 命令运行 Remmina。

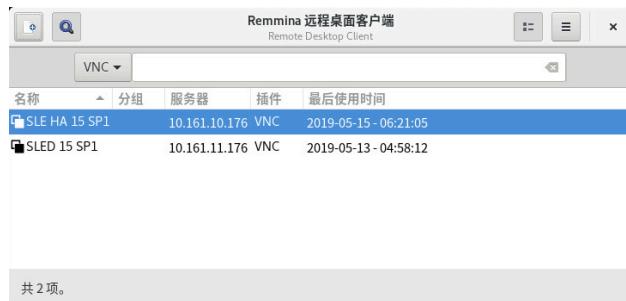


图 14.2：REMMINA 的主窗口

该应用程序主窗口显示存储的远程会话列表。在这里，您可以添加和保存新远程会话、快速启动新会话而不保存、启动以前保存的会话，或设置 Remmina 的全局首选项。

## 14.2.3 添加远程会话

要添加和保存新远程会话，请单击主窗口左上方的 。远程桌面首选项窗口即会打开。

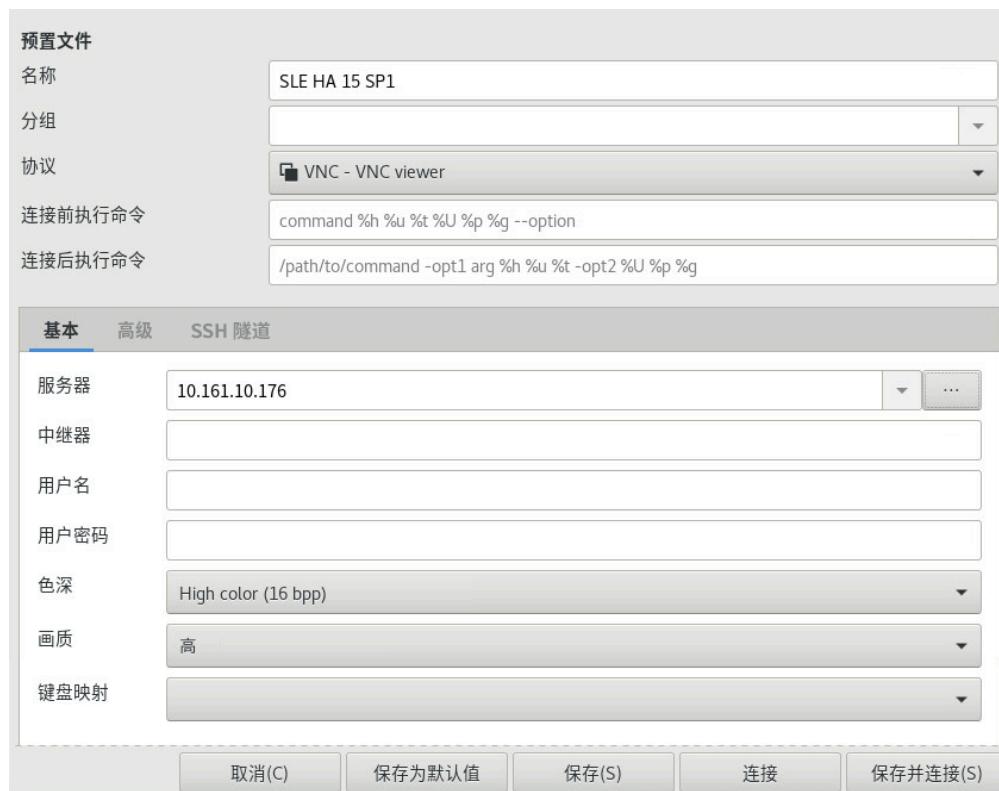


图 14.3：远程桌面首选项

在用于指定新添加的远程会话配置文件的字段中填写信息。最重要的技术包括：

### 名称

配置文件的名称，将列于主窗口中。

### 协议

连接到远程会话时要使用的协议，例如 VNC。

### 服务器

远程服务器的 IP 或 DNS 地址和显示号。

### 用户名、口令

要用于进行远程身份验证的身份凭证。保留为空表示不进行身份验证。

### 色深、质量

根据连接速度和质量选择最佳选项。

选择高级选项卡可输入更具体的设置。



## 提示：禁用加密

如果客户端与远程服务器之间的通讯不加密，请激活禁用加密，否则连接会失败。

选择 SSH 选项卡可显示高级 SSH 隧道通讯进程和身份验证选项。

单击保存进行确认。新设置的配置文件将列在主窗口中。

### 14.2.4 启动远程会话

您可以启动以前保存的会话，也可以快速启动一个远程会话而不保存连接细节。

#### 14.2.4.1 快速启动远程会话

要快速启动远程会话而不添加并保存连接细节，请使用主窗口顶部的下拉框和文本框。



图 14.4：快速启动

从下拉框中选择通讯协议（例如“VNC”），然后输入 VNC 服务器 DNS 或 IP 地址，后跟一个冒号和显示号，然后按 **Enter** 确认。

#### 14.2.4.2 打开保存的远程会话

要打开特定的远程会话，请从会话列表中双击该会话。

#### 14.2.4.3 远程会话窗口

远程会话会在新窗口的标签中打开。每个标签托管一个会话。窗口左侧的工具栏可用来管理窗口/会话。例如，切换全屏模式、调整窗口大小以适应会话的显示大小、将特定按键发送到会话、对会话进行屏幕截图，或设置图像质量。

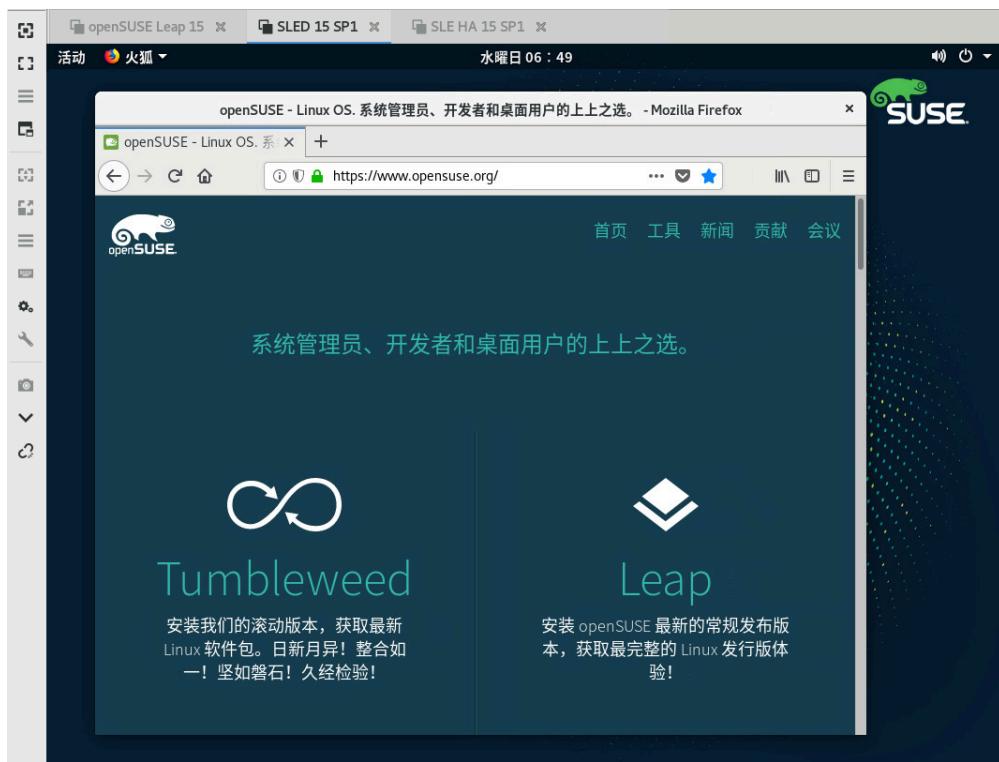


图 14.5：正在查看远程会话的 REMMINA

## 14.2.5 编辑、复制和删除保存的会话

要**编辑**保存的某个远程会话，请在 Remmina 的主窗口中右键单击其名称，然后选择**编辑**。有关相关字段的说明，请参见**第 14.2.3 节 “添加远程会话”**。

要**复制**保存的某个远程会话，请在 Remmina 的主窗口中右键单击其名称，然后选择**复制**。在远程桌面首选项窗口中，更改配置文件的名称，（可选）调整相关选项，然后单击**保存确认**。

要**删除**保存的某个远程会话，请在 Remmina 的主窗口中右键单击其名称，然后选择**删除**。在下一个对话框中，单击**是确认**。

## 14.2.6 从命令行运行远程会话

如果您需要从命令行或使用批处理文件打开远程会话，而不先打开应用程序主窗口，请使用以下语法：

```
> remmina -c profile_name.remmina
```

Remmina 的配置文件存储在您主目录下的 `~/.local/share/remmina/` 目录中。要确定哪个配置文件属于您要打开的会话，请运行 Remmina，在主窗口中单击会话名称，然后在窗口底部的状态行中查看配置文件的路径。



图 14.6：读取配置文件的路径

如果 Remmina 未在运行，您可以将配置文件重命名为更合理的文件名，例如 `sle15.remmina`。您甚至可以将配置文件复制到自定义目录，并从该目录中使用 `remmina -c` 命令来运行它。

## 14.3 在 VNC 服务器上配置一次性会话

一次性会话由远程客户端启动。它在服务器上启动图形登录屏幕。这样您可以选择启动会话的用户，并且如果登录管理器支持，还可以选择桌面环境。取消与此类 VNC 会话的客户端连接时，此会话中启动的所有应用程序也将终止。一次性 VNC 会话不能共享，但可以在一台主机上同时存在多个会话。

### 过程 14.1：启用一次性 VNC 会话

1. 启动 YaST > 网络服务 > 远程管理 (VNC)。
2. 选中允许远程管理（不含会话管理）。
3. 如果您打算在 Web 浏览器窗口中访问 VNC 会话，请激活允许使用 Web 浏览器访问。
4. 如果需要，还可以选中打开防火墙中的端口（例如，当网络接口配置为在外部区域中时）。如果有多个网络接口，请通过防火墙细节将打开防火墙端口限制为特定的接口。

5. 单击下一步确认您的设置。

6. 如果不是所有需要的软件包都可使用，则需要批准安装缺少的软件包。

### 提示：重启显示管理器

YaST 对显示管理器设置进行了更改。您需要注销当前图形会话并重启显示管理器，以使更改生效。



图 14.7：远程管理

#### 14.3.1 可用配置

SUSE Linux Enterprise Desktop 上的默认配置以 1024x768 像素（颜色深度 16 位）显示会话。会话在端口 5901（对于“普通”VNC 查看器，等同于 VNC 显示器 1）和端口 5801 上可用（对于 Web 浏览器）。

其他配置可在不同端口上使用。如果需要修改配置，请咨询您的系统管理员了解细节。

VNC 显示器编号和 X 显示器编号是独立于一次性会话的。VNC 显示器编号手动指派给服务器支持的每个配置（如上例中的 :1）。只要 VNC 会话启动时带任一配置，就会自动获取可用 X 显示器编号。

默认情况下，VNC 客户端与服务器将尝试通过安装后生成的自我签名 SSL 证书安全通讯。您可以使用默认的证书，也可以将它替换为您自己的证书。使用自我签名证书时，需要在首次连接之前，在 VNC 查看器和 Web 浏览器中确认证书的签名。



## 提示

某些 VNC 客户端拒绝通过默认的自我签名证书建立安全连接。例如，Vnagre 客户端会向 GnuTLS 全局可信证书存储区验证证书，如果证书是自我签名证书，则不会建立连接。在这种情况下，请使用 x509 以外的加密方法，或者为 VNC 服务器生成正确签名的证书并将其导入到客户端的系统可信证书存储区。

### 14.3.2 启动一次性 VNC 会话

要连接一次性 VNC 会话，必须安装 VNC 查看器，另请参见第 14.1 节 “[vncviewer 客户端](#)”。或者，可在支持 JavaScript 的 Web 浏览器中输入以下 URL 来查看 VNC 会话：<http://jupiter.example.com:5801>

### 14.3.3 配置一次性 VNC 会话

如果不需或想修改默认配置，则可以跳过此部分。

一次性 VNC 会话通过 systemd 套接字 xvnc.socket 启动。该套接字默认提供六个配置块：三个用于 VNC 查看器（vnc1 到 vnc3），另外三个用于 JavaScript 客户端（vnchttpd1 到 vnchttpd3）。默认情况下，只有 vnc1 和 vnchttpd1 是活动的。

要在引导时激活 VNC 服务器套接字，请运行以下命令：

```
> sudo systemctl enable xvnc.socket
```

要立即启动套接字，请运行：

```
> sudo systemctl start xvnc.socket
```

**Xvnc** 服务器可通过 server\_args 选项配置。有关选项列表，请参见 **Xvnc --help**。

当添加自定义配置时，请确保它们未使用已由其他配置、其他服务或同一主机上的现有永久 VNC 会话使用的端口。

通过输入以下命令激活配置更改：

```
> sudo systemctl reload xvnc.socket
```

## ! 重要：防火墙和 VNC 端口

按照[过程 14.1 “启用一次性 VNC 会话”](#) 中的描述激活远程管理时，端口 5801 和 5901 将在防火墙中打开。如果用于 VNC 会话的网络接口受防火墙保护，则为 VNC 会话激活更多端口时，需要手动打开各个端口。有关指导，请参见《安全和强化指南》，第 23 章“[掩蔽和防火墙](#)”。

## 14.4 配置永久 VNC 服务器会话

可以从多个客户端同时访问持续会话。为了便于演示，我们选择一个较为理想的配置，一个客户端具有完全访问权限，所有其他客户端只具有查看访问权限。另一个用例是：教员可能需要在其中访问学员桌面的培训会话。



### 提示：连接永久 VNC 会话

要连接持续 VNC 会话，必须安装 VNC 查看器。有关更多细节，请参见[第 14.1 节“\[vncviewer 客户端\]\(#\)”](#)。或者，可在支持 JavaScript 的 Web 浏览器中输入以下 URL 来查看 VNC 会话：<http://jupiter.example.com:5801>

持续 VNC 会话分为以下两类：

- 使用 `vncserver` 启动的 VNC 会话
- 使用 `vncmanager` 启动的 VNC 会话

### 14.4.1 使用 `vncserver` 启动的 VNC 会话

此类型的持续 VNC 会话在服务器上启动。该会话和其上启动的所有应用程序运行时不考虑客户端连接，直到会话被终止。永久会话访问受到两种可用口令类型的保护：

- 授予完全访问权限的普通口令或
- 可选仅查看口令，授予非交互（仅查看）访问权限。

一个会话可一次具有两种类型的多个客户端连接。

#### 过程 14.2：使用 `vncserver` 启动永久 VNC 会话

1. 打开外壳，确保以拥有 VNC 会话的用户身份登录。
2. 如果用于 VNC 会话的网络接口受防火墙保护，则需要手动打开防火墙中您的会话所使用的端口。如果启动多个会话，还可以选择打开一个端口范围。有关如何配置防火墙的细节，请参见《安全和强化指南》，第 23 章“掩蔽和防火墙”。  
`vncserver` 对显示器 `:1` 使用端口 `5901`，对显示器 `:2` 使用端口 `5902`，以此类推。对于永久会话，VNC 显示器和 X 显示器的编号通常都是相同的。
3. 要其他具有 `1024x768` 像素和颜色深度为 `16` 的会话，请输入以下命令：

```
vncserver -alwaysshared -geometry 1024x768 -depth 16
```

`vncserver` 命令会在您未指定时选取一个未使用的显示编号，并打印输出它的选择。有关更多选项，请参见 `man 1 vncserver`。

当您首次运行 `vncserver` 时，它会要求您输入一个口令，以获取会话的完全访问权限。如果需要，还可以提供口令用于会话的仅查看访问。

这里提供的口令还可供同一用户用来启动以后的会话。这些口令可以用 `vncpasswd` 命令更改。

### ! 重要：安全注意事项

确保使用长度够长的高强度口令（八个或更多字符）。不要共享这些口令。

要取消会话，请从 VNC 查看器中关闭在 VNC 会话内部运行的桌面环境，就像关闭一般本地 X 会话一样。

如果希望手动取消会话,请在 VNC 服务器上打开外壳,并务必以拥有要取消的 VNC 会话的用户身份登录。运行以下命令可取消在显示器 :1 上运行的会话: **vncserver -kill :1**

#### 14.4.1.1 配置永久 VNC 会话

通过编辑 `$HOME/.vnc/xstartup` 可以配置持续 VNC 会话。默认情况下,此外壳脚本会启动它启动时所处的同一个 GUI/窗口管理器。在 SUSE Linux Enterprise Desktop 中,此 GUI/窗口管理器为 GNOME 或 IceWM。要使用您选择的窗口管理器启动会话,请设置变量 `WINDOWMANAGER`:

```
WINDOWMANAGER=gnome vncserver -geometry 1024x768
WINDOWMANAGER=icewm vncserver -geometry 1024x768
```

#### 注意: 每个用户一种配置

持续 VNC 会话在单个按用户配置中进行配置。由同一个用户启动的多个会话都使用相同的启动文件和口令文件。

#### 14.4.2 使用 vncmanager 启动的 VNC 会话

##### 过程 14.3: 启用永久 VNC 会话

1. 启动 YaST > 网络服务 > 远程管理 (VNC)。
2. 激活允许远程管理 (含会话管理)。
3. 如果您打算在 Web 浏览器窗口中访问 VNC 会话,请激活允许使用 Web 浏览器访问。
4. 如果需要,还可以选中打开防火墙中的端口 (例如,当网络接口配置为在外部区域中时)。如果有多个网络接口,请通过防火墙细节将打开防火墙端口限制为特定的接口。
5. 单击下一步确认您的设置。
6. 如果不是所有需要的软件包都可使用,则需要批准安装缺少的软件包。



## 提示：重启动显示管理器

YaST 对显示管理器设置进行了更改。您需要注销当前图形会话并重启动显示管理器，以使更改生效。

### 14.4.2.1 配置永久 VNC 会话

按过程 14.3 “启用永久 VNC 会话” 中所述启用 VNC 会话管理后，便可以使用您喜欢的 VNC 查看器（例如 `vncviewer` 或 Remmina）正常连接到远程会话。此时将显示登录屏幕。登录后，您桌面环境的系统托盘中将出现“VNC”图标。单击该图标可打开 VNC 会话窗口。如果该图标未出现，或者您的桌面环境不支持图标放在系统托盘中，请手动运行 `vncmanager-controller`。



图 14.8：VNC 会话设置

有几个设置会影响 VNC 会话的行为：

### 非持续，私用

这相当于一次性会话。其他用户将看不到此会话，它在您断开连接后即会终止。有关更多信息，请参考[第 14.3 节 “在 VNC 服务器上配置一次性会话”](#)。

### 持续，可见

其他用户可以看见此类会话，它在您断开连接后仍保持运行。

### 会话名称

您可以在此处指定持续会话的名称，以便在重新连接时可以轻松识别它。

### 不需要口令

任何人不必使用用户身份凭证登录即可访问会话。

### 需要用户登录

需要使用有效的用户名和口令登录后才能访问会话。该选项会在允许的用户文本框中列出有效的用户名。

### 一次允许一个客户端

防止多个用户同时加入该会话。

### 一次允许多个客户端

允许多个用户同时加入持续会话。适合用于远程演示或培训会话。

单击确定进行确认。

## 14.4.2.2 加入永久 VNC 会话

按[第 14.4.2.1 节 “配置永久 VNC 会话”](#) 中所述设置持续 VNC 会话后，可通过 VNC 查看器加入它。当 VNC 客户端连接到服务器后，系统将提示您选择是要创建新会话还是加入现有会话：

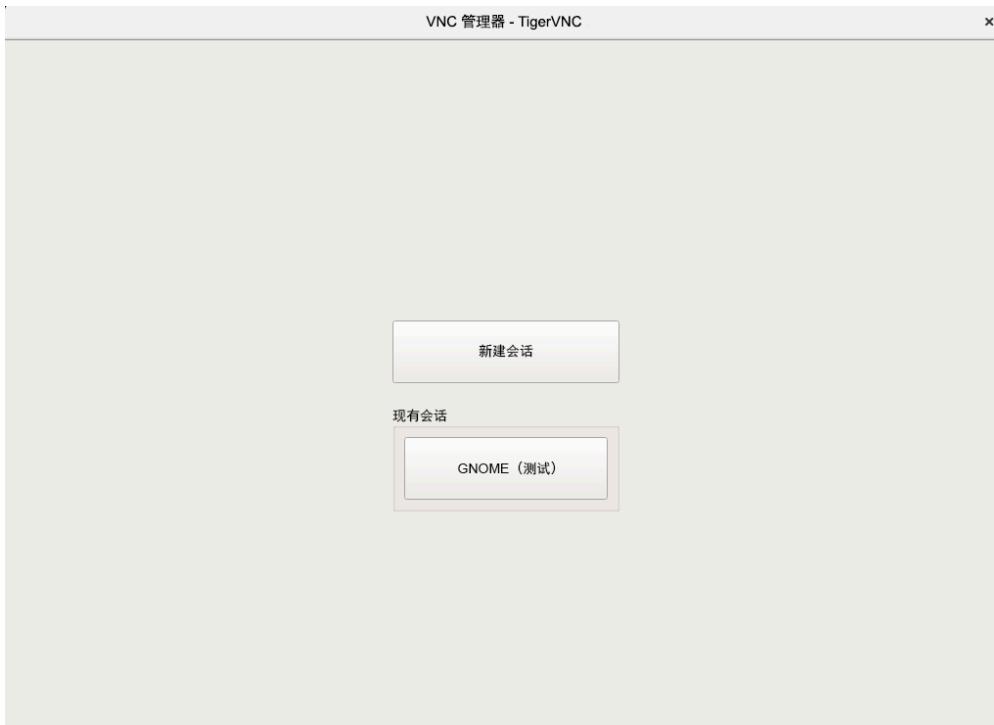


图 14.9：加入永久 VNC 会话

当您单击现有会话的名称后，系统可能要求您输入登录身份凭证，具体取决于持续会话设置。

## 14.5 在 VNC 服务器上配置加密

如果 VNC 服务器设置正确，则 VNC 服务器与客户端之间的所有通讯都会被加密。身份验证在会话开始时进行；实际的数据传输在身份验证后开始。

无论是一次性还是持续 VNC 会话，安全性选项都通过 `server_args` 一行中 `/usr/bin/Xvnc` 命令的 `-securitytypes` 参数来配置。`-securitytypes` 参数用于选择身份验证方法和加密。它的选项如下：

### 身份验证

#### **None、TLSNone、X509None**

无身份验证。

#### **VncAuth、TLSVnc、X509Vnc**

身份验证使用自定义口令。

## Plain、TLSPlain、X509Plain

身份验证使用 PAM 来验证用户的口令。

### 加密

## None、vncAuth、plain

不加密。

## TLSNone、TLSVnc、TLSPlain

匿名 TLS 加密。对所有内容加密，但不校验远程主机。因此，您可以防护被动攻击者，但不能防御中间人攻击者。

## X509None、X509Vnc、X509Plain

使用证书进行 TLS 加密。如果使用自我签名证书，则在第一次连接时，系统将要求您校验证书。在以后的连接中，仅当证书有变动时，系统才会向您发出警告。因此，在第一次连接时，您可以防御中间人攻击之外的所有其他攻击（类似于使用典型的 SSH）。如果使用由证书颁发机构签名且与计算机名称匹配的证书，您的安全将得到全面保障（类似于使用典型的 HTTPS）。



### 提示

某些 VNC 客户端拒绝通过默认的自我签名证书建立安全连接。例如，Vinagre 客户端会向 GnuTLS 全局可信证书存储区验证证书，如果证书是自我签名证书，则不会建立连接。在这种情况下，请使用 x509 以外的加密方法，或者为 VNC 服务器生成正确签名的证书并将其导入到客户端的系统可信证书存储区。



### 提示：证书和密钥的路径

如果使用基于 X509 的加密，需要通过 -X509Cert 和 -X509Key 选项指定 X509 证书和密钥的路径。

如果您选择多种安全性类型（用逗号分隔），将会使用客户端和服务器都支持且允许的第一种安全性。如此，您便可在服务器上配置随机加密。如果您需要支持不支持加密的 VNC 客户端，此功能将十分有用。

在客户端上，您也可以指定允许的安全性类型，以防在您连接到已知启用了加密的服务器时遭到降级攻击（虽然在该情况下，我们的 vncviewer 会发出“连接未加密！”消息来警告您）。

## 14.6 与 Wayland 的兼容性

远程管理 (VNC) 功能依赖于 X11，如果启用 Wayland，系统可能会显示空白屏幕。显示管理器必须配置为使用 X11 而不是 Wayland。对于 gdm，请编辑 /etc/gdm/custom.conf。在 [daemon] 部分中，向配置文件添加 WaylandEnable=false。登录时，用户还必须选择与 X11 兼容的会话。如果想要为 GNOME 去除 Wayland 选项，可以去除并锁定 gnome-session-wayland 软件包。

# 15 使用 RSync 复制文件

当今时代，用户通常都会有数台计算机：家用计算机和办公计算机、笔记本电脑、智能手机或平板电脑。因而，在多个设备之间保持文件和文档同步的任务就变得越发重要。



## 警告：数据丢失风险

您在开始使用同步工具之前，应该先熟悉其特性和功能。请务必备份您的重要文件。

## 15.1 概念概述

对于要通过慢速网络连接同步大量数据的情况，Rsync 提供了可靠的方法来只传输文件中的更改。此方法不仅适用于文本文件，还适用于二进制文件。为了检测文件之间的差异，Rsync 将文件分为多个块，并计算它们的校验和。

检测更改对计算能力有一定的要求。因此，请确保两端的计算机均具有足够的资源，包括 RAM。

当需要定期传输大量只包含微小更改的数据时，Rsync 特别有用。进行备份时就常常用到该工具。Rsync 也非常适合用来镜像试验服务器，此类服务器将 Web 服务器的完整目录树存储到 DMZ 内的某台 Web 服务器中。

Rsync 并不是同步工具，虽然它的名字看上去有些像。Rsync 工具一次只能在一个方向复制数据。它不会也不能反向复制数据。如果您需要既能同步源又能同步目标的双向工具，请使用 Csync。

## 15.2 基本语法

Rsync 是一个命令行工具，基本语法如下：

```
rsync [OPTION] SOURCE [SOURCE]... DEST
```

您可以在任何本地或远程计算机上使用 Rsync，前提是您拥有相应的访问权限和写入权限。可以有多个 SOURCE 项。SOURCE 和 DEST 占位符可以是路径和/或 URL。

下面介绍一些最常用的 Rsync 选项：

-v

输出较详细的文本

-a

存档模式；以递归方式复制文件并保留时间戳、用户/组所有权、文件权限和符号链接

-z

压缩传输的数据



### 注意：尾部斜杠计数

使用 Rsync 时，要特别注意尾部斜杠。目录后面的尾部斜杠表示目录的**内容**。没有尾部斜杠表示**目录本身**。

## 15.3 在本地复制文件和目录

下面的说明假设当前用户拥有 `/var/backup` 目录的写入许可权限。要将单个文件从计算机上的一个目录复制到另一个路径，请使用以下命令：

```
> rsync -avz backup.tar.xz /var/backup/
```

文件 `backup.tar.xz` 会复制到 `/var/backup/`，绝对路径是 `/var/backup/backup.tar.xz`。

请勿忘记在 `/var/backup/` 目录后面加上**尾部斜杠**。如果不插入斜杠，文件 `backup.tar.xz` 会复制到 `/var/backup` (文件) 中，而不是 `/var/backup/` 目录中！

复制目录与复制单个文件相似。下面的示例会将目录 `tux/` 及其内容复制到 `/var/backup/` 目录中：

```
> rsync -avz tux /var/backup/
```

在绝对路径 `/var/backup/tux/` 中可找到副本。

## 15.4 远程复制文件和目录

两台计算机上都需要有 Rsync 工具。要从远程目录复制文件或将文件复制到远程目录，需要提供 IP 地址或域名。如果本地计算机和远程计算机上当前的用户名相同，则可以不指定用户名。要使用相同的用户（在本地和远程主机上）将文件 file.tar.xz 从本地主机复制到远程主机 192.168.1.1，请使用以下命令：

```
> rsync -avz file.tar.xz tux@192.168.1.1:
```

根据您的个人偏好，也可以使用下面的命令，它们的作用相同：

```
> rsync -avz file.tar.xz 192.168.1.1:~  
> rsync -avz file.tar.xz 192.168.1.1:/home/tux
```

在使用标准配置的所有情况下，系统会提示您输入远程用户的通行口令。此命令会将 file.tar.xz 复制到用户 tux 的主目录（通常为 /home/tux）。

远程复制目录与在本地复制目录相似。下面的示例将目录 tux/ 及其内容复制到 192.168.1.1 主机上的远程目录 /var/backup/：

```
> rsync -avz tux 192.168.1.1:/var/backup/
```

假设您在主机 192.168.1.1 上拥有写入许可权限，便可在绝对路径 /var/backup/tux 中找到副本。

## 15.5 配置和使用 Rsync 服务器

Rsync 可作为在默认端口 873 上侦听传入连接的守护程序 (rsyncd) 运行。此守护程序可以接收“复制目标”。

下面的说明介绍如何在 jupiter 上创建具有备份目标的 Rsync 服务器。此目标可用于存储您的备份。要创建 Rsync 服务器，请执行以下操作：

### 过程 15.1：设置 RSYNC 服务器

1. 在 jupiter 上，创建用于存储您所有备份文件的目录。此示例中使用了 /var/backup：

```
# mkdir /var/backup
```

2. 指定所有权。在此示例中，该目录为 users 组中的用户 tux 所拥有：

```
# chown tux.users /var/backup
```

3. 配置 rsyncd 守护程序。

我们将配置文件分割成一个主文件，以及某些用于存放您的备份目标的“模块”。如此，以后便可更轻松地添加其他目标。全局值可以存储在 /etc/rsyncd.d/\*.inc 文件中，而模块则存放于 /etc/rsyncd.d/\*.conf 文件中：

- a. 创建目录 /etc/rsyncd.d/：

```
# mkdir /etc/rsyncd.d/
```

- b. 在主配置文件 /etc/rsyncd.conf 中，添加以下几行：

```
# rsyncd.conf main configuration file
log file = /var/log/rsync.log
pid file = /var/lock/rsync.lock

&merge /etc/rsyncd.d ①
&include /etc/rsyncd.d ②
```

- ① 将 /etc/rsyncd.d/\*.inc 文件中的全局值合并到主配置文件中。
- ② 从 /etc/rsyncd.d/\*.conf 文件装载任何模块（或目标）。这些文件不应该包含对全局值的任何参照。

- c. 在文件 /etc/rsyncd.d/backup.conf 中添加以下几行创建您的模块（您的备份目标）：

```
# backup.conf: backup module
[backup] ①
uid = tux ②
gid = users ②
path = /var/backup ③
auth users = tux ④
```

```
secrets file = /etc/rsyncd.secrets ⑤
comment = Our backup target
```

- ① **备份目标**。可以使用您喜欢的任何名称。但最好根据目标的用途来命名，并在 \*.conf 文件中使用相同名称。
  - ② 指定在进行文件传输时所用的用户名或组名。
  - ③ 定义用于存储备份的路径（从**步骤 1**中）。
  - ④ 指定允许的用户的逗号分隔列表。列表以最简单的方式包含允许连接到此模块的用户名。在我们的示例中，只允许用户 tux。
  - ⑤ 指定包含用户名和明文口令的行所在文件的路径。
- d. 创建包含以下内容的 /etc/rsyncd.secrets 文件，并替换 PASSPHRASE:

```
# user:passwd
tux:PASSPHRASE
```

- e. 确保该文件只有 root 用户可读：

```
# chmod 0600 /etc/rsyncd.secrets
```

4. 通过以下命令启动并启用 rsyncd 守护程序：

```
# systemctl enable rsyncd
# systemctl start rsyncd
```

5. 测试是否可访问 Rsync 服务器：

```
> rsync jupiter::
```

您应该会看到类似如下的响应：

```
backup          Our backup target
```

若非如此，请检查您的配置文件、防火墙和网络设置。

上述步骤创建了 Rsync 服务器，现在可以使用它来存储备份。下例还创建了一个列出所有连接的日志文件。此文件存储在 /var/log/rsyncd.log 中。此文件可用来对传输进行调试。

要列出备份目标的内容，请使用以下命令：

```
> rsync -avz jupiter::backup
```

此命令会列出服务器上 `/var/backup` 目录中存在的所有文件。此请求也会记录在日志文件 `/var/log/rsyncd.log` 中。要开始实际传输，请提供源目录。用 `.` 来表示当前目录。例如，下面的命令会将当前目录复制到 Rsync 备份服务器：

```
> rsync -avz . jupiter::backup
```

默认情况下，Rsync 不会在运行时删除文件和目录。要允许删除，必须另外指定选项 `--delete`。要确保不删除任何较新的文件，可改为使用选项 `--update`。必须手动解决所有冲突。

## 15.6 更多信息

### Csync

双向文件同步工具，请参见 <https://csync.org/>。

### RSnapshot

创建增量备份，请参见 <https://rsnapshot.org/>。

### Unison

与 CSync 类似的文件同步工具，但具有图形界面，请参见 <https://www.seas.upenn.edu/~bcpierce/unison/>。

### Rear

一种灾难恢复框架，请参见 Administration Guide of the SUSE Linux Enterprise High Availability Extension, chapter Disaster Recovery with Rear (Relax-and-Recover) (<https://documentation.suse.com/sle-ha-15/html/SLE-HA-all/cha-ha-rear.html>)。

## II 引导 Linux 系统

- 16 引导过程简介 213
- 17 UEFI (统一可扩展固件接口) 221
- 18 引导加载程序 GRUB 2 229
- 19 `systemd` 守护程序 248

# 16 引导过程简介

引导 Linux 系统涉及不同组件和任务。固件和硬件初始化过程（取决于计算机的体系结构）完成后，系统将通过引导加载程序 GRUB 2 启动内核。在此之后，引导进程完全由操作系统控制，并由 `systemd` 处理。`systemd` 会提供一组“目标”，用于引导与日常使用、维护或紧急情况相关的配置。

## 16.1 术语

本章使用的术语可能存在歧义。为了理解本章中术语的用法，请阅读以下定义：

### init

有两个不同的进程通常命名为“init”：

- 用于挂载根文件系统的 `initramfs` 进程
- 从实际根文件系统执行且用于启动其他所有进程的操作系统进程

在这两种情况下，`systemd` 程序都会处理此任务。首先会从 `initramfs` 执行此进程，以挂载根文件系统。挂载成功后，此进程将作为初始进程从根文件系统重新执行。为了避免混淆这两个 `systemd` 进程，我们将第一个进程称为 **init on initramfs**，将第二个进程称为 **systemd**。

### initrd/initramfs

`initrd`（初始 RAM 磁盘）是一个映像文件，内含将由内核装载并作为临时根文件系统从 `/dev/ram` 挂载的根文件系统映像。挂载此文件系统需要使用文件系统驱动程序。

从内核 2.6.13 开始，`initrd` 已由 `initramfs`（初始 RAM 文件系统）取代，后者无需文件系统驱动程序即可挂载。SUSE Linux Enterprise Desktop 只使用 `initramfs`。但是，由于 `initramfs` 作为 `/boot/initrd` 存储，因此通常将其称为“`initrd`”。本章只使用名称 `initramfs`。

## 16.2 Linux 引导进程

Linux 引导进程包括多个阶段，每个阶段由一个不同组件来代表：

1. 第 16.2.1 节 “初始化和引导加载程序阶段”
2. 第 16.2.2 节 “内核阶段”
3. 第 16.2.3 节 “Init on initramfs 阶段”
4. 第 16.2.4 节 “systemd 阶段”

### 16.2.1 初始化和引导加载程序阶段

在初始化阶段，将设置计算机的硬件并准备好设备。此过程根据硬件体系结构的不同有很大的差别。

SUSE Linux Enterprise Desktop 在所有体系结构中都使用引导加载程序 GRUB 2。根据体系结构和固件，启动 GRUB 2 引导加载程序的过程可能包括多个步骤。引导加载程序的用途是装载内核以及基于 RAM 的初始文件系统 (initramfs)。有关 GRUB 2 的详细信息，请参见[第 18 章 “引导加载程序 GRUB 2”](#)。

#### 16.2.1.1 AArch64 和 AMD64/Intel 64 上的初始化和引导加载程序阶段

在打开计算机之后，BIOS 或 UEFI 将初始化屏幕和键盘并测试主内存。直到这一阶段，计算机不访问任何大容量存储媒体。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。识别引导媒体及其几何尺寸之后，系统控制权将从 BIOS/UEFI 转到引导加载程序。

在装配传统 BIOS 的计算机上，只能装载引导磁盘的第一个 512 字节物理数据扇区（主引导记录，MBR）中的代码。只有极少量的 GRUB 2 代码能够挂载 MBR。引导加载程序的唯一作用就是从 MBR 与第一个分区（MBR 分区表）之间的间隙处，或是从 BIOS 引导分区（GPT 分区表）装载包含文件系统驱动程序的 GRUB 2 核心映像。此映像包含文件系统驱动程序，因此能够访问根文件系统中的 `/boot`。`/boot` 包含 GRUB 2 核心的附加模块以及内核和 initramfs 映像。获取此分区的访问权限后，GRUB 2 会将内核和 initramfs 映像装载到内存中，并将控制权交给内核。

从包含已加密分区 `/boot` 的加密文件系统引导 BIOS 系统时，需要输入解密口令两次。第一次是为了让 GRUB 2 解密 `/boot`，第二次是让 `systemd` 挂载加密的卷。

在装配 UEFI 的计算机上，引导过程比装配传统 BIOS 的计算机要简单得多。固件能够读取包含 GPT 分区表的磁盘的 FAT 格式化系统分区。此 EFI 系统分区（在运行的系统中挂载为 `/boot/efi`）可提供足够的空间，用于存储由固件直接装载和执行的完备 GRUB 2。

如果 BIOS/UEFI 支持网络引导，则也可以配置提供引导加载程序的引导服务器。然后，可以通过 PXE 引导系统。BIOS/UEFI 充当引导加载程序。它会从引导服务器获取引导映像，然后启动系统。这完全不依赖本地硬盘。

### 16.2.1.2 IBM Z 上的初始化和引导加载程序阶段

在 IBM Z 上，必须通过名为 `zipl`（z 初始程序装载）的引导加载程序初始化引导进程。尽管 `zipl` 支持读取不同的文件系统，但它不支持 SLE 默认文件系统 (Btrfs) 或者从快照引导。因此，SUSE Linux Enterprise Desktop 使用两阶段的引导过程来确保引导时完全支持 Btrfs：

1. `zipl` 从 `/boot/zipl` 分区引导，该分区可格式化为 Ext2、Ext3、Ext4 或 XFS 文件系统。此分区包含一个极简的内核，以及一个装载到内存中的 initramfs。initramfs 包含 Btrfs 驱动程序（及其他组件）和引导加载程序 GRUB 2。内核是使用参数 `initgrub`（告知要启动 GRUB 2）启动的。
2. 内核会挂载根文件系统，使 `/boot` 可访问。现在，将从 initramfs 启动 GRUB 2。GRUB 从 `/boot/grub2/grub.cfg` 读取其配置，并从 `/boot` 装载最终的内核和 initramfs。现在，将通过 Kexec 装载新内核。

### 16.2.2 内核阶段

引导加载程序转交系统控制权后，所有体系结构中的引导过程都是相同的。引导加载程序会将内核和基于 RAM 的初始文件系统 (`initramfs`) 都装载到内存中，而内核将接管控制权。

内核设立内存管理并检测 CPU 类型及其功能后，将初始化硬件，并从内存中挂载使用 `initramfs` 装载的临时根文件系统。

## 16.2.2.1 `initramfs` 文件

`initramfs` (初始 RAM 文件系统) 是一个小型 cpio 存档, 可由内核装载到 RAM 磁盘中。它位于 `/boot/initrd` 中。可以使用名为 `dracut` 的工具创建该文件, 有关详细信息, 请参见 [man 8 dracut](#)。

`initramfs` 提供了一个极简的 Linux 环境, 可用于在挂载实际根文件系统之前执行程序。这个最小的 Linux 环境由 BIOS 或 UEFI 例程载入内存, 而且除了需要足够的内存外没有特定的硬件要求。`initramfs` 存档必须始终提供一个名为 `init` 的可执行文件, 该文件执行根文件系统上的 `systemd` 守护程序, 使引导进程得以继续。

在能够挂载 `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。根文件系统所需的模块由 `init` 在 `initramfs` 装载。装载模块后, `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。可以使用 `systemd` 单元 `systemd-udev-trigger.service` 来实现此目的。

### 16.2.2.1.1 重新生成 `initramfs`

由于 `initramfs` 包含多个驱动程序, 因此, 每当其中某个驱动程序有新版本发布时, 都需要更新 `initramfs`。在安装包含驱动程序更新的软件包时可以自动完成这种更新。YaST 或 zypper 通过显示用于生成 `initramfs` 的命令的输出来告知此状况。但在某些情况下, 您需要手动重新生成 `initramfs`:

- 由于更换硬件而需添加驱动程序
- 将系统目录移到 RAID 或 LVM
- 将磁盘添加到包含根文件系统的 LVM 组或 Btrfs RAID
- 更改内核变量

#### 由于更换硬件而需添加驱动程序

如果需要更换硬件 (例如硬盘), 并且引导时此硬件需要内核中的不同驱动程序, 则您必须更新 `initramfs` 文件。

打开或创建 `/etc/dracut.conf.d/10-DRIVER.conf`，并添加下面这行（请注意前导空格）：

```
force_drivers+=" DRIVER1 "
```

用驱动程序的模块名称替换 `DRIVER1`。如果您需要添加多个驱动程序，请将其全部列出并以空格分隔：

```
force_drivers+=" DRIVER1 DRIVER2 "
```

继续[过程 16.1 “生成 initramfs”](#)。

## 将系统目录移到 RAID 或 LVM

每次您要将正在运行的系统中的交换文件或系统目录（例如 `/usr`）移到 RAID 或逻辑卷时，都需要创建包含软件 RAID 或 LVM 驱动程序支持的 `initramfs`。

为此，请在 `/etc/fstab` 中创建相关的项，并挂载新项（例如，使用 `mount -a` 和/或 `swapon -a`）。

继续[过程 16.1 “生成 initramfs”](#)。

## 将磁盘添加到包含根文件系统的 LVM 组或 Btrfs RAID

每当您要在包含根文件系统的逻辑卷组或者 Btrfs RAID 中添加（或去除）磁盘时，都需要创建一个支持扩容的卷的 `initramfs`。请按照[过程 16.1 “生成 initramfs”](#) 中的指导操作。

继续[过程 16.1 “生成 initramfs”](#)。

## 更改内核变量

如果您在 `sysctl` 界面中通过编辑相关文件（`/etc/sysctl.conf` 或 `/etc/sysctl.d/*.conf`）更改了内核变量的值，系统下次重引导时，这项更改将会丢失。即使您在运行时使用 `sysctl --system` 装载这些值，更改也不会保存到 `initramfs` 文件中。您需要根据[过程 16.1 “生成 initramfs”](#) 中所述更新该文件。

### 过程 16.1：生成 INITRAMFS

请注意，下面的过程中的所有命令都需要以 `root` 用户身份执行。

1. 进入 `/boot` 目录：

```
# cd /boot
```

2. 使用 `dracut` 生成一个新 `initramfs` 文件（以您选择的文件名替换 `MY_INITRAMFS`）：

```
# dracut MY_INITRAMFS
```

或者，可以运行 `dracut -f FILENAME` 来替换现有的 `init` 文件。

3. （如果在上一步中运行了 `dracut -f`，请跳过此步骤。）基于您在上一步中创建的 `initramfs` 文件创建指向 `initrd` 的符号链接：

```
# ln -sf MY_INITRAMFS initrd
```

4. 在 IBM Z 体系结构中，另外还需运行 `grub2-install`。

### 16.2.3 Init on initramfs 阶段

由内核从 `initramfs` 挂载的临时根文件系统包含可执行文件 `systemd`（下面称作 `init on initramfs`，另请参见第 16.1 节“术语”）。此程序执行挂载正确根文件系统所需的全部操作。它为所需的文件系统提供内核功能，并为使用 `udev` 的大量存储控制器提供设备驱动程序。`initramfs` 上的 `init` 的主要用途是准备真实 `root` 文件系统的挂载和访问。根据您的系统配置的不同，`initramfs` 上的 `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

内核根据装载的模块生成设备事件。`udev` 会处理这些事件，并在 `/dev` 中的 RAM 文件系统上生成所需的特殊块文件。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 下保存根文件文件系统，则 `initramfs` 上的 `init` 将设置 LVM 或 RAID 以支持以后对根文件系统的访问。

## 管理网络配置

如果将系统配置为使用通过网络挂载的 root 文件系统（通过 NFS 挂载），则 `init` 必须确保装载了正确的网络驱动程序，并确保将其设置为支持访问 root 文件系统。

如果文件系统驻留在一个联网的块设备（如 iSCSI 或 SAN）上，则与存储服务器的连接也由 `initramfs` 上的 `init` 设置。SUSE Linux Enterprise Desktop 支持在主要目标不可用的情况下从次要 iSCSI 目标引导。



## 注意：处理挂载失败问题

如果根文件系统无法从引导环境中挂载，则必须先对其进行检查和修复，才能继续引导。

如果文件系统为 Ext3 和 Ext4，文件系统检查程序将会自动启动。如果是 XFS 和 Btrfs 文件系统，则不会自动开始修复过程，而是向用户显示有关可用于修复文件系统的选项的信息。成功修复文件系统后，退出引导环境将会使系统重试挂载根文件系统。如果挂载成功，将正常继续引导。

### 16.2.3.1 安装过程中的 init on initramfs 阶段

如果在安装过程的初始引导阶段调用 `init on initramfs`，它要执行的任务将与上述任务不同。请注意，安装系统也不会从 `initramfs` 启动 `systemd` — 这些任务由 `linuxrc` 执行。

#### 查找安装媒体

当您启动安装进程时，计算机会装载一个安装内核以及一个包含 YaST 安装程序的特殊 `init`。YaST 安装程序正在 RAM 文件系统中运行，它需要知道安装媒体的位置，才能访问安装媒体以安装操作系统。

#### 启动硬件识别并装载适当的内核模块

如第 16.2.2.1 节 “`initramfs` 文件” 中所述，引导过程从最少的一组驱动程序（可在大多数硬件配置中使用）开始。在 AArch64、POWER 和 AMD64/Intel 64 计算机上，`linuxrc` 会启动初始硬件扫描进程，以确定适合您的硬件配置的驱动程序集。在 IBM Z 上，需要提供驱动程序及其参数的列表（例如，通过 `linuxrc` 或 `parmfile` 提供）。这些驱动程序用于生成引导系统所需的自定义 `initramfs`。如果引导时不需要这些模块，但冷插拔时需要，您可以使用 `systemd` 装载这些模块。有关详细信息，请参见第 19.6.4 节 “装载内核模块”。

## 装载安装系统

系统在正确识别硬件后会装载相应的驱动程序。udev 程序会创建特殊的设备文件，linuxrc 使用 YaST 安装程序启动安装系统。

## 启动 YaST

最后，linuxrc 启动 YaST，后者则启动软件包安装和系统配置。

## 16.2.4 systemd 阶段

找到“实际的”根文件系统后，对其进行错误检查并挂载。如果挂载成功，系统会清理 initramfs 并执行根文件系统上的 systemd 守护程序。systemd 是 Linux 的系统和服务管理器。它是作为 PID 1 启动的父进程，充当用于启动和维护用户空间服务的 init 系统。有关详细信息，请参见 [第 19 章“systemd 守护程序”](#)。

# 17 UEFI (统一可扩展固件接口)

UEFI (统一可扩展固件接口) 是用于系统硬件自带的固件、系统所有的硬件组件以及操作系统之间的接口。

UEFI 在 PC 系统上的应用范围越来越广，因此正在逐渐替代传统的 PC-BIOS。例如，UEFI 能够很好地支持 64 位系统，提供安全的引导（“安全引导”，要求固件为 2.3.1c 或以上版本）。安全引导是其最为重要的特性之一。最后，借助 UEFI，标准固件将能够用于所有 x86 平台。

除此之外，UEFI 还具有以下优点：

- 从带有 GUID 分区表 (GPT) 的大磁盘（超过 2 TiB）引导。
- 独立于 CPU 的架构和驱动程序。
- 带有网络功能的灵活的预操作系统环境。
- 通过 PC-BIOS 式仿真支持引导老式操作系统的 CSM（兼容支持模块）。

有关详细信息，请参见[http://en.wikipedia.org/wiki/Unified\\_Extensible\\_Firmware\\_Interface](http://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface)。以下小节并不是 UEFI 的整体概述，仅提供如何在 SUSE Linux Enterprise Desktop 实现特定功能的提示。

## 17.1 安全引导

在 UEFI 领域中，要想保障引导程序的安全，需要建立一个信任链。“平台”是此信任链的根；在 SUSE Linux Enterprise Desktop 环境中，可将主板和板载固件视为“平台”。换句话说，它就是硬件供应商，信任链是从硬件供应商流向组件制造商、OS 供应商等。

系统通过公共密钥加密法表示信任。硬件供应商将所谓的“平台密钥 (PK)”放入固件中，代表可信根。操作系统供应商与其他方将其密钥与“平台密钥”签署在一起，以此记录他们之间的信任关系。

最后，通过要求固件不得执行任何代码（除非使用其中某个“可信”密钥签署了代码）来建立安全保障。这些代码可能是操作系统引导加载程序、位于特定 PCI Express 卡闪存中或磁盘上的驱动程序，也可能是固件本身的更新。

要使用安全引导，您需要用固件信任的密钥对您的 OS 加载程序签名，并且需要 OS 加载程序校验其加载的内核是否可信。

可以将密钥交换密钥 (KEK) 添加到 UEFI 密钥数据库中。这样，您便可以使用其他证书，只要这些证书是以 PK 的私用部分签署的即可。

### 17.1.1 在 SUSE Linux Enterprise Desktop 上实施

默认情况下安装微软的密钥交换密钥 (KEK)。

#### 注意：需要 GUID 分区表 (GPT)

UEFI/x86\_64 安装中默认会启用安全引导功能。您可在引导加载程序设置对话框的引导代码选项选项卡中找到启用安全引导支持选项。该选项支持在固件中的安全引导已激活时引导，同时也支持在安全引导已停用时引导。

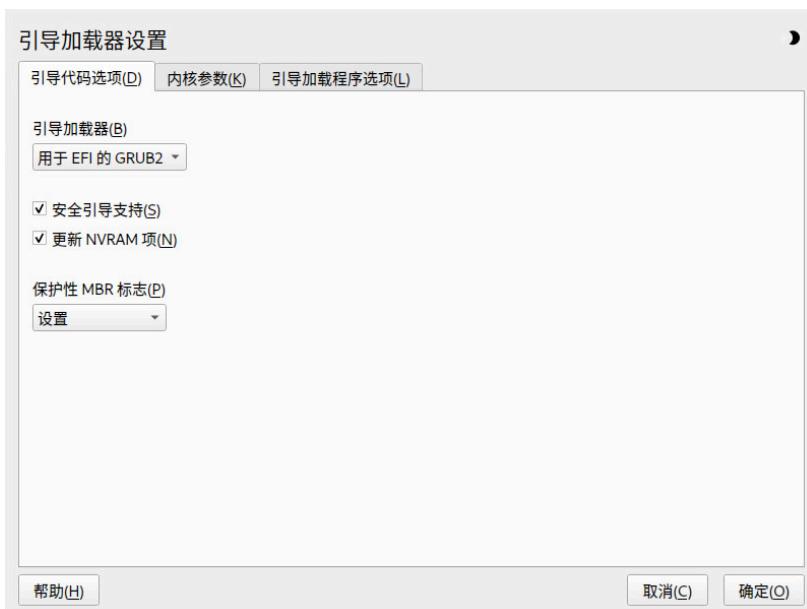


图 17.1：安全引导支持

“安全引导”特性要求 GUID 分区表 (GPT) 使用主引导记录 (MBR) 替代旧的分区。如果安装期间 YaST 检测到 EFI 模式，则会设法创建 GPT 分区。UEFI 预期会在 FAT 格式的 EFI 系统分区(ESP) 上查找到 EFI 程序。

要支持 UEFI 安全引导，需要有引导加载程序，且固件认可其数字签名是可信密钥。该密钥需要先天为固件所信任（无需任何手动干预）。

有两种办法可以实现。一种是与硬件供应商合作，让其签署 SUSE 密钥，然后 SUSE 会使用该 SUSE 密钥签署引导加载程序；另一种是通过微软的 Windows 徽标认证计划使引导加载程序获得认证，并使微软认可 SUSE 签名密钥（也就是让加载程序使用他们的 KEK 签名）。至此，SUSE 使引导加载程序获得了 UEFI 签名服务（在此情况下为 Microsoft）的签名。

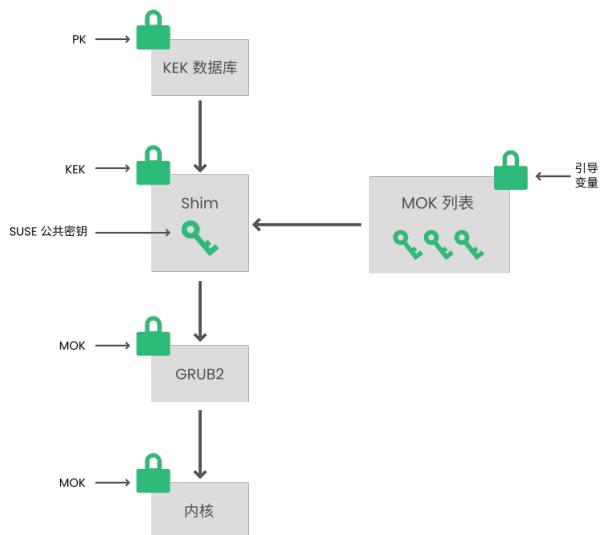


图 17.2：UEFI：安全引导过程

SUSE 在实施层使用默认将会安装的 shim 加载程序。这是一种可以避免法律纠纷的智能解决方案，能够大幅简化认证和签名步骤。shim 加载程序的任务是装载 GRUB 2 等引导加载程序并对其进行校验，之后，此引导加载程序将装载仅由一个 SUSE 密钥签名的内核。SUSE 从全新安装的 SLE11 SP3 开始提供此功能，并要求启用 UEFI 安全引导。

可信用户分为以下两类：

- 首先是持有密钥的用户。平台密钥 (PK) 几乎允许所有操作。密钥交换密钥 (KEK) 的许可范围与 PK 一致，但不能更改 PK。
- 其次是能够以物理方式访问机器的任何用户。具有物理访问权限的用户可以重引导机器并对 UEFI 进行配置。

UEFI 提供下列两类变量以满足这些用户的需求：

- 第一类变量即所谓的“已验证的变量”，它们可从引导过程（所谓的“引导服务环境”）和正在运行的操作系统中更新。仅当对变量新值签名的密钥是用于对变量旧值签名的相同密钥时，才能进行更新。而且，您只能向这类变量追加或将其更改为序列号更高的数值。
- 第二类变量即所谓的“仅供引导服务使用的变量”。引导进程中运行的任何代码都可以获取这些变量。在结束引导进程并准备启动 OS 的时间间隔内，引导加载程序必须调用 `ExitBootServices` 呼叫。此后将无法获取这些变量，OS 也无法接触到这些变量。

UEFI 密钥列表属于第一类，因为该类变量除了允许联机更新外，还允许添加密钥、驱动程序、固件指纹以及将它们列入黑名单。第二类变量即“仅供引导服务使用的变量”。该类变量有助于以安全且支持开源的方式实施安全引导，因此符合 GPLv3 要求。

SUSE 首先启动 `shim`，它是一个小而简单的 EFI 引导加载程序，由 SUSE 和 Microsoft 签名。这样一来，`shim` 即可加载并执行。

`shim` 随后继续验证其想要加载的引导加载程序是否可信。默认情况下，`shim` 会使用其主体中所嵌入的独立的 SUSE 证书。此外，`shim` 还允许“登记”其他密钥，用于覆盖默认的 SUSE 密钥。下文将这些密钥称为“机器拥有者密钥”或缩写为 MOK。

接下来，引导加载程序会验证内核，然后加以引导。该内核将在模块上执行同样的操作。

### 17.1.2 MOK (机器拥有者密钥)

要替换特定内核、驱动程序或引导过程中的其他组件，需要使用机器拥有者密钥 (MOK)。`mokutil` 工具可帮助您管理 MOK。

您可以使用 `mokutil` 创建 MOK 登记请求。该请求存储在名为 `MokNew` 的 UEFI 运行时 (RT) 变量中。在下次引导期间，`shim` 引导加载程序会检测 `MokNew` 并装载 `MokManager`，这将为您提供几个选项。您可以使用从磁盘登记密钥和从磁盘登记哈希选项来将密钥添加到 `MokList`。使用登记 MOK 选项可从 `MokNew` 变量复制密钥。

通常，当 `shim` 无法装载 `grub2` 并改为装载 `MokManager` 时，会从磁盘登记密钥。由于 `MokNew` 尚不存在，您可以选择在 UEFI 分区上查找密钥。

### 17.1.3 引导自定义内核

以下内容基于 [https://en.opensuse.org/openSUSE:UEFI#Booting\\_a\\_custom\\_kernel](https://en.opensuse.org/openSUSE:UEFI#Booting_a_custom_kernel)。

安全引导不会阻止您使用自行编译的内核。您必须使用自己的证书在该内核上签名，并让固件或 MOK 得以识别该证书。

1. 创建一个自定义的 X.509 密钥以及用于签名的证书：

```
openssl req -new -x509 -newkey rsa:2048 -keyout key.asc \
-out cert.pem -nodes -days 666 -subj "/CN=$USER/"
```

有关创建证书的详细信息，请参见 [https://en.opensuse.org/openSUSE:UEFI\\_Image\\_File\\_Sign\\_Tools#Create\\_Your\\_Own\\_Certificate](https://en.opensuse.org/openSUSE:UEFI_Image_File_Sign_Tools#Create_Your_Own_Certificate)。

2. 将密钥和证书打包成 PKCS#12 结构：

```
> openssl pkcs12 -export -inkey key.asc -in cert.pem \
-name kernel_cert -out cert.p12
```

3. 生成用于 pesign 的 NSS 数据库：

```
> certutil -d . -N
```

4. 将 PKCS#12 中包含的密钥和证书导入 NSS 数据库：

```
> pk12util -d . -i cert.p12
```

5. 使用 pesign 将新签名“赋予”内核：

```
> pesign -n . -c kernel_cert -i arch/x86/boot/bzImage \
-o vmlinuz.signed -s
```

6. 列出内核映像上的签名：

```
> pesign -n . -S -i vmlinuz.signed
```

此时，您可以照常在 /boot 中安装内核。由于内核现有一个自定义的签名，因此需要将用于签名的证书导入 UEFI 固件或 MOK 中。

7. 将证书转为 DER 格式，以供导入固件或 MOK：

```
> openssl x509 -in cert.pem -outform der -out cert.der
```

## 8. 将证书复制到 ESP 以简化访问:

```
> sudo cp cert.der /boot/efi/
```

## 9. 使用 mokutil 自动启动 MOK 列表。

- a. 将证书导入到 MOK 中:

```
> mokutil --root-pw --import cert.der
```

--root-pw 选项可直接允许使用 root 用户。

- b. 检查准备注册的证书列表:

```
> mokutil --list-new
```

- c. 重引导系统; shim 应该会启动 MokManager。您需要输入 root 口令以确认将证书导入到 MOK 列表中。

- d. 检查新导入的密钥以前是否注册过:

```
> mokutil --list-enrolled
```

- a. 此外, 也可以采用如下方式手动启动 MOK:  
重引导

- b. 在 GRUB 2 菜单中, 按 c 键。

- c. 类型:

```
chainloader $efibootdir/MokManager.efd  
boot
```

- d. 选择从磁盘登记密钥。

- e. 导航至 cert.der 文件并按 Enter。

- f. 按照指导登记密钥。正常情况下应按 “0” , 然后按 “y” 予以确认。

除此之外, 固件菜单也可能提供了多种向“签名数据库”中添加新密钥的方式。

## 17.1.4 使用非内置驱动程序

在启用安全引导的情况下，不支持在安装过程中添加非内置驱动程序（即，不是 SUSE Linux Enterprise Desktop 自带的驱动程序）。用于 SolidDriver/PLDP 的签名密钥默认不受信任。

您可以通过两种不同的方式，在启用安全引导的情况下于安装期间安装第三方驱动程序。在这两种情况下，都要：

- 在安装前，通过固件或系统管理工具将所需密钥添加到固件数据库中。此选项取决于您当前使用的基本硬件。请咨询您的硬件供应商了解详细信息。
- 使用 <https://drivers.suse.com/> 上或硬件供应商提供的可引导驱动程序 ISO，在首次引导时将所需密钥登记到 MOK 列表中。

要使用可引导驱动程序 ISO 将驱动程序密钥登记到 MOK 列表中，请执行以下步骤：

1. 将上文所述 ISO 映像刻录到空 CD/DVD 媒体中。
2. 使用新的 CD/DVD 媒体开始安装，并准备好标准的安装媒体或网络安装服务器的 URL。  
如果您要进行网络安装，请在引导命令行上使用 `install=` 选项输入网络安装源的 URL。  
如果您是从光学媒体安装，安装程序会先从驱动程序包引导，然后要求插入产品的第一张安装光盘。
3. 安装时将会使用包含经过更新的驱动程序的 `initrd`。

有关详细信息，请参见[https://drivers.suse.com/doc/Usage/Secure\\_Boot\\_Certificate.html](https://drivers.suse.com/doc/Usage/Secure_Boot_Certificate.html)。

## 17.1.5 功能和限制

以安全引导模式引导时，可以使用以下功能：

- 安装到 UEFI 默认的引导加载程序位置，这是为了保留或恢复 EFI 引导项而采用的机制。
- 通过 UEFI 重引导。
- 如果没有其他可回退到的旧版 BIOS，Xen 超级管理程序将使用 UEFI 引导。
- 支持 UEFI IPv6 PXE 引导。

- 支持 UEFI 视频模式，内核可以从 UEFI 检索视频模式，以使用相同的参数配置 KMS 模式。
- UEFI 支持从 USB 设备引导。
- 从 SUSE Linux Enterprise Server 15 SP3 开始，在安全引导模式下支持 Kexec 和 Kdump。

以安全引导模式引导时，存在以下限制：

- 为了确保无法轻易绕过安全引导，系统在安全引导模式下运行时会禁用特定内核功能。
- 引导加载程序、内核以及内核模块必须经过签名。
- 休眠（挂起到磁盘）处于禁用状态。
- 无法访问 `/dev/kmem` 和 `/dev/mem`，root 用户也不例外。
- 无法访问 I/O 端口，连 root 用户也不例外。所有 X11 图形驱动程序必须使用内核驱动程序。
- 无法通过 sysfs 访问 PCI BAR。
- 无法使用 ACPI 中的 `custom_method`。
- 无法使用 asus-wmi 模块的 debugfs。
- `acpi_rsdp` 参数对内核没有任何影响。

## 17.2 更多信息

- <https://www.uefi.org> — UEFI 主页，其中列出了最新的 UEFI 规范。
- 由 Olaf Kirch 与 Vojtěch Pavlík 撰写的博文（上述章节内容主要取材于这些博文）：
  - <https://www.suse.com/c/uefi-secure-boot-plan/>
  - <https://www.suse.com/c/uefi-secure-boot-overview/>
  - <https://www.suse.com/c/uefi-secure-boot-details/>
- <https://en.opensuse.org/openSUSE:UEFI> — UEFI 与 openSUSE。

# 18 引导加载程序 GRUB 2

本章介绍如何配置 SUSE® Linux Enterprise Desktop 中使用的引导加载程序 GRUB 2。GRUB 是传统 GRUB 引导加载程序（现在称作“GRUB Legacy”）的后继产品。从 SUSE® Linux Enterprise Desktop 12 版开始，就已使用 GRUB 2 作为默认的引导加载程序。产品中提供了一个 YaST 模块来配置最重要的设置。[第 16 章“引导过程简介”](#) 中将引导过程作为一个整体进行了介绍。有关 UEFI 计算机的安全引导支持的细节，请参见[第 17 章“UEFI（统一可扩展固件接口）”](#)。

## 18.1 GRUB Legacy 与 GRUB 2 之间的主要差异

- 配置存储在不同的文件中。
- 支持更多的文件系统（例如 Btrfs）。
- 可以直接读取 LVM 或 RAID 设备上存储的文件。
- 用户界面可翻译，并可以改变主题。
- 包含一项用于装载模块的机制，以支持更多功能，例如文件系统等。
- 自动搜索和生成其他内核与操作系统（例如 Windows）的引导项。
- 包含一个类似于 Bash 的精简控制台。

## 18.2 配置文件结构

GRUB 2 的配置基于以下文件：

/boot/grub2/grub.cfg

此文件包含 GRUB 2 菜单项的配置。它替代了 GRUB Legacy 中的 menu.lst。不要编辑 grub.cfg — 它是由命令 grub2-mkconfig -o /boot/grub2/grub.cfg 自动生成的。

## /boot/grub2/custom.cfg

此可选文件在引导时由 grub.cfg 直接检索，可用于向引导菜单添加自定义项。从 SUSE Linux Enterprise Desktop 12 SP2 开始，使用 grub-once 时也将分析这些项目。

## /etc/default/grub

此文件控制 GRUB 2 的用户设置，通常包含背景和主题等其他环境设置。

## /etc/grub.d/ 下的脚本

在执行 grub2-mkconfig -o /boot/grub2/grub.cfg 命令期间，系统会读取此目录中的脚本。主配置文件 /boot/grub/grub.cfg 中集成了这些脚本的指令。

## /etc/sysconfig/bootloader

此配置文件保存一些基本设置，例如引导加载程序类型，以及是否要启用 UEFI 安全引导支持。

## /boot/grub2/x86\_64-efi, /boot/grub2/power-ieee1275

这些配置文件包含特定于体系结构的选项。

GRUB 2 可以通过多种方式进行控制。可以在图形菜单（启动屏幕）中选择现有配置的引导项。配置从文件 /boot/grub2/grub.cfg 装载，而该文件是基于其他配置文件编译的（参见下文）。所有 GRUB 2 配置文件都被视为系统文件，编辑这些配置文件需要拥有 root 特权。



### 注意：激活配置更改

手动编辑 GRUB 2 配置文件后，需要运行 grub2-mkconfig -o /boot/grub2/grub.cfg 以激活更改。但使用 YaST 更改配置时就不需要如此，因为 YaST 会自动运行此命令。

## 18.2.1 文件 /boot/grub2/grub.cfg

带有引导菜单的图形启动屏幕上的内容取决于 GRUB 配置文件 /boot/grub2/grub.cfg，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信息。

系统每次引导时，GRUB 2 会直接从文件系统装载菜单文件。因此，在更改配置文件后不需要重新安装 GRUB 2。安装或去除内核后，系统会自动重构 grub.cfg。

`grub.cfg` 是在运行 `grub2-mkconfig -o /boot/grub2/grub.cfg` 命令时，基于文件 `/etc/default/grub` 以及 `/etc/grub.d/` 目录中的脚本编译的。因此，切勿手动编辑该文件，而应该编辑相关的源文件，或者根据第 18.3 节 “使用 YaST 配置引导加载程序” 中所述，使用 YaST 引导加载程序模块来修改配置。

## 18.2.2 文件 `/etc/default/grub`

此文件包含 GRUB 2 的其他常规选项，例如，显示菜单的时间，或者要引导的默认操作系统。要列出所有可用选项，请查看以下命令的输出：

```
> grep "export GRUB_DEFAULT" -A50 /usr/sbin/grub2-mkconfig | grep GRUB_
```

除了已定义的变量外，用户还可以引入自己的变量，以便日后在 `/etc/grub.d` 目录中的脚本内使用。

编辑 `/etc/default/grub` 后，请使用 `grub2-mkconfig -o /boot/grub2/grub.cfg` 更新主配置文件。



### 注意：范围

此文件中设置的所有选项是会影响所有引导项的常规选项。通过 `GRUB_*_XEN_*` 配置选项可以设置 Xen 内核或 Xen 超级管理程序的特定选项。有关细节，请参见下文。

#### GRUB\_DEFAULT

设置默认会引导的引导菜单项。它的值可以是数字值、菜单项的完整名称或 “saved”。

`GRUB_DEFAULT=2` 引导第三个（从零开始计数）引导菜单项。

`GRUB_DEFAULT="2>0"` 会引导第三个顶级菜单项的第一个子菜单项。

`GRUB_DEFAULT="Example boot menu entry"` 会引导名为 “Example boot menu entry”的菜单项。

`GRUB_DEFAULT=saved` 会引导 `grub2-once` 或 `grub2-set-default` 命令指定的项。`grub2-reboot` 只设置下一次重引导的默认引导项，而 `grub2-set-default` 设置的是永久的默认引导项，直至其被更改。`grub2-editenv list` 列出下一个引导项。

## GRUB\_HIDDEN\_TIMEOUT

等待用户按某个键的指定秒数。在此期间，除非用户按下某个键，否则不显示菜单。如果用户在指定的时间内未按任何键，控制权将转交给 GRUB\_TIMEOUT。GRUB\_HIDDEN\_TIMEOUT=0 首先会检查是否已按下 `Shift`，如果是，则显示引导菜单，否则会立即引导默认的菜单项。如果 GRUB 2 只识别了一个可引导操作系统，则默认行为就是如此。

## GRUB\_HIDDEN\_TIMEOUT\_QUIET

如果指定 `false`，并且启用了 GRUB\_HIDDEN\_TIMEOUT 功能，将会在一个空白屏幕上显示倒数计时器。

## GRUB\_TIMEOUT

在自动引导默认引导项之前，显示引导菜单的期限（以秒为单位）。如果按下某个键，则会取消超时，GRUB 2 将等待您手动做出选择。如果指定 GRUB\_TIMEOUT=-1，则在您手动选择引导项之前，会一直显示菜单。

## GRUB\_CMDLINE\_LINUX

此行中的项将添加到正常和恢复模式的引导项的末尾。使用它可以向引导项添加内核参数。

## GRUB\_CMDLINE\_LINUX\_DEFAULT

与 GRUB\_CMDLINE\_LINUX 一样，但只能在正常模式下追加项。

## GRUB\_CMDLINE\_LINUX\_RECOVERY

与 GRUB\_CMDLINE\_LINUX 一样，但只能在恢复模式下追加项。

## GRUB\_CMDLINE\_LINUX\_XEN\_REPLACE

此项将替代所有 Xen 引导项的 GRUB\_CMDLINE\_LINUX 参数。

## GRUB\_CMDLINE\_LINUX\_XEN\_REPLACE\_DEFAULT

与 GRUB\_CMDLINE\_LINUX\_XEN\_REPLACE 一样，但只会替代 GRUB\_CMDLINE\_LINUX\_DEFAULT 的参数。

## GRUB\_CMDLINE\_XEN

此项只为 Xen guest 内核指定内核参数 — 工作原理与 GRUB\_CMDLINE\_LINUX 相同。

## GRUB\_CMDLINE\_XEN\_DEFAULT

与 GRUB\_CMDLINE\_XEN 一样 — 工作原理与 GRUB\_CMDLINE\_LINUX\_DEFAULT 相同。

## GRUB\_TERMINAL

启用和指定输入/输出终端设备。可以是 console (PC BIOS 和 EFI 控制台)、serial (串行终端)、ofconsole (开放式固件控制台) 或默认值 gfxterm (图形模式输出)。用引号括住所需的选项可以启用多个设备，例如 GRUB\_TERMINAL="console serial"。

## GRUB\_GFXMODE

gfxterm 图形终端使用的分辨率。只能使用您的图形卡 (VBE) 支持的模式。默认值为 “auto”，即尝试选择首选的分辨率。在 GRUB 2 命令行中键入 videoinfo 可以显示 GRUB 2 适用的屏幕分辨率。GRUB 2 引导菜单屏幕显示后，键入 **c** 即可访问命令行。您还可以指定颜色深度，方法是将其值追加到分辨率设置的后面，例如 GRUB\_GFXMODE=1280x1024x24。

## GRUB\_BACKGROUND

设置 gfxterm 图形终端的背景图像。该图像必须是 GRUB 2 在引导时可读的文件，并且必须以 .png、.tga、.jpg 或 .jpeg 后缀结尾。必要时，系统会缩放该图像以适合屏幕。

## GRUB\_DISABLE\_OS\_PROBER

如果将此选项设置为 true，将禁用自动搜索其他操作系统的功能。系统只会检测 /boot/ 中的内核映像，以及 /etc/grub.d/ 中您自己的脚本内的选项。

## SUSE\_BTRFS\_SNAPSHOT\_BOOTING

如果将此选项设置为 true，GRUB 2 可直接引导至 Snapper 快照。有关详细信息，请参见 [第 10.3 节 “通过从快照引导来执行系统回滚”](#)。

要显示完整的选项列表，请参见 [GNU GRUB manual \(<http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration>\)](http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration)。

### 18.2.3 /etc/grub.d 中的脚本

在执行 `grub2-mkconfig -o /boot/grub2/grub.cfg` 命令期间，系统会读取此目录中的脚本。脚本的指令将整合到 `/boot/grub2/grub.cfg` 中。`grub.cfg` 中菜单项的顺序由此目录中文件的运行顺序来决定。具有前导编号的文件先执行，从最小的编号开始。`00_header` 在 `10_linux` 之前运行，而后者又在 `40_custom` 之前运行。如果存在采用字母名称的文件，这些文件将在采用编号命名的文件后面执行。在执行 `grub2-mkconfig` 期间，只有可执行文件才能在 `grub.cfg` 中生成输出。默认情况下，`/etc/grub.d` 目录中的所有文件都是可执行文件。



#### 提示：将自定义内容永久保存在 `grub.cfg` 中

由于每次运行 `grub2-mkconfig` 时都会重新编译 `/boot/grub2/grub.cfg`，因此所有自定义内容都会丢失。要想直接在 `/boot/grub2/grub.cfg` 中插入行并让这些行在运行 `grub2-mkconfig` 后不丢失，需要将它们插入以下两行之间

```
### BEGIN /etc/grub.d/90_persistent ###
```

和

```
### END /etc/grub.d/90_persistent ###
```

`90_persistent` 脚本可确保此类内容会保留下来。

下面列出了最重要的脚本：

#### `00_header`

设置环境变量，例如系统文件位置、显示设置、主题和以前保存的项。它还可以导入 `/etc/default/grub` 中存储的首选项。通常，不需要对此文件进行更改。

#### `10_linux`

识别根设备上的 Linux 内核，并创建相关的菜单项。这包括关联的恢复模式选项（如果已启用）。主菜单页中只显示最新内核，其他内核包含在子菜单中。

#### `30_os-prober`

此脚本使用 `os-prober` 来搜索 Linux 和其他操作系统，并将结果列于 GRUB 2 菜单中。其中的某些部分可以识别其他特定操作系统，例如 Windows 或 macOS。

## 40\_custom

使用此文件可以方便地在 `grub.cfg` 中包含自定义引导项。切勿更改开头的 `exec tail -n +3 $0` 部分。

处理顺序根据前导编号确定，编号最小的脚本最先执行。如果多个脚本的前导编号相同，则根据整个名称的字母顺序来决定执行顺序。



### 提示: `/boot/grub2/custom.cfg`

如果您创建了 `/boot/grub2/custom.cfg` 并在其中填充了内容，系统会在引导时自动将它包含到 `/boot/grub2/grub.cfg` 中，放置在 40\_custom 的后面。

## 18.2.4 BIOS 驱动器与 Linux 设备之间的映射

在 GRUB Legacy 中，`device.map` 配置文件用于基于 BIOS 驱动器号派生 Linux 设备名称。不一定总能正确猜测出 BIOS 驱动器与 Linux 设备之间的映射。例如，如果 IDE 和 SCSI 驱动器的引导顺序在 BIOS 配置中被交换，则 GRUB Legacy 就会发生顺序错误。

GRUB 2 在生成 `grub.cfg` 时使用设备 ID 字符串 (UUID) 或文件系统标签，因此避免了此问题。GRUB 2 实用程序会即时创建一个临时设备映射，这通常足以满足需要，尤其是在单磁盘系统上。

但是，如果您需要覆盖 GRUB 2 的自动设备映射机制，请创建自定义映射文件 `/boot/grub2/device.map`。以下示例将更改映射，使 `DISK 3` 成为引导磁盘。请注意，GRUB 2 分区编号以 `1` 开始，而不是像 GRUB Legacy 中那样以 `0` 开始。

```
(hd1) /dev/disk-by-id/DISK3 ID
(hd2) /dev/disk-by-id/DISK1 ID
(hd3) /dev/disk-by-id/DISK2 ID
```

## 18.2.5 在引导过程中编辑菜单项

当系统由于配置错误而不再能够引导时，如果能够直接编辑菜单项，就会很有帮助。使用菜单编辑器还可以在不更改系统配置的情况下测试新设置。

1. 在图形引导菜单中，使用方向键选择要编辑的项。

2. 按 **E** 打开基于文本的编辑器。

3. 使用方向键移到您要编辑的行。



GNU GRUB version 2.04

```
set root='hd0,gpt2'
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint='hd0,gpt2' 3c2\51c37-7ebb-4aaa-a658-eca1e810198d
else
  search --no-floppy --fs-uuid --set=root 3c251c37-7ebb-4aaa-a65\8-eca1e810198d
fi
echo      'Loading Linux 5.3.18-8-default ...'
linux    /boot/vmlinuz-5.3.18-8-default root=UUID=3c251c37-7\ebb-4aaa-a658-eca1e810198d ${extra_cmdline} splash=silent resume=/dev/v\da4 mitigations=auto quiet crashkernel=195M,high crashkernel=72M,low
echo      'Loading initial ramdisk ...'
initrd   /boot/initrd-5.3.18-8-default
```

Minimum Emacs-like screen editing is supported. TAB lists completions. Press Ctrl-x or F10 to boot, Ctrl-c or F2 for a command-line or ESC to discard edits and return to the GRUB menu.

图 18.1：GRUB 2 引导编辑器

现在，您可以采取以下两种做法：

- a. 将用空格分隔的参数添加到以 linux 或 linuxefi 开头的行的末尾，以编辑内核参数。<https://en.opensuse.org/Linuxrc> 上提供了完整的参数列表。
  - b. 或者编辑常规选项，以更改内核版本等。按 **→|** 键会显示所有可能的补全建议。
4. 按 **F10** 使用您所做的更改引导系统，或者按 **Esc** 放弃您的编辑，并返回到 GRUB 2 菜单。

通过这种方式进行的更改仅会应用到当前引导过程，而不会永久保存。

## ! 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《Deployment Guide》，第 8 章 “Troubleshooting”，第 8.3 节 “Booting from installation media fails”，US keyboard layout。



## 注意：安装媒体中的引导加载程序

在使用传统 BIOS 的系统上，安装媒体中的引导加载程序仍是 GRUB Legacy。要添加引导参数，请选择一项，然后开始键入。在安装引导项中添加的选项将永久保存在安装的系统中。

### 18.2.6 设置引导口令

即使在操作系统引导之前，GRUB 2 也支持对文件系统的访问。没有 root 权限的用户可以访问 Linux 系统中的文件，而在引导系统后，他们将无权访问这些文件。要阻止此类访问或防止用户引导某些菜单项，请设置引导口令。



## 重要：引导需要口令

如果设置了引导口令，则每次引导时都需要输入该口令，这意味着系统不会自动引导。

按如下步骤设置引导口令。或者使用 YaST（使用口令保护引导加载程序）。

#### 1. 使用 grub2-mkpasswd-pbkdf2：来加密口令：

```
> sudo grub2-mkpasswd-pbkdf2
Password: ****
Reenter password: ****
PBKDF2 hash of your password is
grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

#### 2. 将生成的字符串连同 set superusers 命令一起粘贴到文件 /etc/grub.d/40\_custom 中。

```
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

#### 3. 要将更改导入到主要配置文件中，请运行：

```
> sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

在重引导后，当您尝试引导某个菜单项时，系统会提示您输入用户名和口令。输入 `root` 以及您在执行 `grub2-mkpasswd-pbkdf2` 命令期间键入的口令。如果身份凭证正确，系统将引导选定的引导项。

有关详细信息，请参见<https://www.gnu.org/software/grub/manual/grub.html#Security>。

### 18.2.7 引导菜单项的授权访问权限

您可以将 GRUB 2 配置为允许根据授权级别访问引导菜单项。您可以配置多个受口令保护的用户帐户，并向其指派不同菜单项的访问权限。要在 GRUB 2 中配置授权，请执行以下步骤：

1. 为要在 GRUB 2 中使用的每个用户帐户创建并加密一个口令。按第 18.2.6 节 “设置引导口令” 中所述使用 `grub2-mkpasswd-pbkdf2` 命令。
2. 删除文件 `/etc/grub.d/10_linux`。这样可以防止输出默认的 GRUB 2 菜单项。
3. 编辑 `/boot/grub2/custom.cfg` 文件，并手动添加自定义菜单项。下面的模板是一个示例，请调整该模板，使其更适合您的用例：

```
set superusers=admin
password admin ADMIN_PASSWORD
password maintainer MAINTAINER_PASSWORD

menuentry 'Operational mode' {
    insmod ext2
    set root=hd0,1
    echo 'Loading Linux ...'
    linux /boot/vmlinuz root=/dev/vda1 $GRUB_CMDLINE_LINUX_DEFAULT
    $GRUB_CMDLINE_LINUX mode=operation
    echo 'Loading Initrd ...'
    initrd /boot/initrd
}

menuentry 'Maintenance mode' --users maintainer {
    insmod ext2
    set root=hd0,1
    echo 'Loading Linux ...'
```

```
linux /boot/vmlinuz root=/dev/vda1 $GRUB_CMDLINE_LINUX_DEFAULT
$GRUB_CMDLINE_LINUX mode=maintenance
echo 'Loading Initrd ...'
initrd /boot/initrd
}
```

#### 4. 将更改导入到主配置文件中：

```
> sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

在上面的示例中：

- GRUB 2 菜单包含以下两项：操作模式和维护模式。
- 如果未指定用户，则两个引导菜单项都可以访问，但任何人都无法访问 GRUB 2 命令行或编辑现有菜单项。
- admin 用户可访问 GRUB 2 命令行并编辑现有的菜单项。
- maintenance 用户可选择恢复菜单项。

## 18.3 使用 YaST 配置引导加载程序

在 SUSE Linux Enterprise Desktop 系统中，配置引导加载程序一般选项最简单的方式就是使用 YaST 模块。在 YaST 控制中心，选择系统 > 引导加载程序。该模块会显示您系统的当前引导加载程序配置，并允许您进行更改。

使用引导代码选项选项卡可查看和更改类型、位置和高级加载程序设置的相关设置。您可以选择要在标准模式还是 EFI 模式下使用 GRUB 2。

### ! 重要：EFI 系统要求使用 GRUB2-EFI

如果您使用的是 EFI 系统，则只能安装 GRUB2-EFI，否则您的系统不再能够引导。

### ! 重要：重新安装引导加载程序

要重新安装引导加载程序，请务必在 YaST 中更改设置，然后再将其改回来。例如，要重新安装 GRUB2-EFI，请先选择 GRUB2，然后立即将其切换回 GRUB2-EFI。

否则，可能只会重新安装引导加载程序的一部分。



## 注意：自定义引导加载程序

要使用此处未列出的引导加载程序，请选择不要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。

### 18.3.1 引导加载程序位置和引导代码选项

引导加载程序的默认位置为主引导记录 (MBR) 或 / 分区的引导扇区，具体视分区设置而定。要修改引导加载程序的位置，请遵循以下步骤：

#### 过程 18.1：更改引导加载程序位置

1. 选择引导代码选项选项卡，然后为引导加载程序位置选择以下某个选项：

##### 从主引导记录引导

选择此选项会在包含 /boot 目录的磁盘 MBR 中安装引导加载程序。通常，这将是挂载到 / 的磁盘，但如果 /boot 挂载到其他磁盘上的独立分区中，将会使用该磁盘的 MBR。

##### 从根分区引导

选择此选项会在 / 分区的引导扇区中安装引导加载程序。

##### 自定义根分区

手动使用此选项来指定引导加载程序的位置。

2. 单击确定以应用更改。



图 18.2：引导代码选项

引导代码选项选项卡包含以下额外的选项：

#### 在分区表中设置活动标志，以引导分区

激活包含 `/boot` 目录的分区。对于 POWER 系统，则会激活 PReP 分区。此选项用于具有旧 BIOS 的系统和/或旧式操作系统，因为它们可能无法从非活动的分区引导。您可以放心地启用此选项。

#### 将通用引导代码写入 MBR 中

如果 MBR 包含自定义的“非 GRUB”代码，此选项会用不受操作系统限制的通用代码替换该代码。如果您停用此选项，系统可能变得无法引导。

#### 启用可信引导支持

启动支持可信计算功能（可信平台模块 (TPM)）的 TrustedGRUB2。有关详细信息，请参见<https://github.com/Sirrix-AG/TrustedGRUB2>。

保护性 MBR 标志部分包含以下选项：

#### set

适用于传统旧式 BIOS 引导。

#### remove

适用于 UEFI 引导。

#### do not change

如果您已有一个正常运行的系统，这通常是最适合的选项。

大多数情况下，YaST 默认会使用适合的选项。

### 18.3.2 调整磁盘顺序

如果您的计算机有多个硬盘，您可以指定磁盘的引导顺序。如果从 MBR 引导，将在列表中的第一个磁盘中安装 GRUB 2。默认在该磁盘中安装 SUSE Linux Enterprise Desktop。列表的其余部分是有关 GRUB 2 的设备映射程序的提示（请参见[第 18.2.4 节 “BIOS 驱动器与 Linux 设备之间的映射”](#)）。



#### 警告：无法引导的系统

通常情况下，默认值几乎对所有部署都有效。如果您错误地更改了磁盘的引导顺序，系统下次重引导时可能无法引导。例如，如果列表中的第一个磁盘不在 BIOS 引导序列中，并且列表中的其他磁盘有空 MBR，系统将无法引导。

#### 过程 18.2：设置磁盘顺序

1. 打开引导代码选项选项卡。
2. 单击编辑磁盘引导顺序。
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. 单击确定两次以保存更改。

### 18.3.3 配置高级选项

可以通过引导加载程序选项选项卡来配置高级引导参数。

### 18.3.3.1 引导加载程序选项选项卡



图 18.3：引导加载程序选项

#### 引导加载程序超时

通过键入新值或者用鼠标单击相应方向键来更改超时（以秒为单位）的值。

#### 探测外来操作系统

如果选择该选项，引导加载程序将会搜索其他系统（例如 Windows）或其他 Linux 安装。

#### 引导时隐藏菜单

隐藏引导菜单并引导默认项。

#### 调整默认引导项

从“默认引导部分”列表中选择所需的项。请注意，引导项名称中的“>”符号用于分隔引导部分及其子部分。

## 使用口令保护引导加载程序

使用一个附加的口令保护引导加载程序和系统。有关手动配置的细节，请参见第 18.2.6 节“[设置引导口令](#)”。如果激活此选项，则每次引导时都需要输入该引导口令，这意味着系统不会自动引导。但是，如果您更喜欢 GRUB 1 的行为方式，请额外启用仅防止修改引导项。使用此设置时，任何人都可以选择引导项并引导系统，仅在修改引导项时，才需要提供 GRUB 2 [root](#) 用户的口令。

### 18.3.3.2 内核参数[选项卡](#)



图 18.4：内核参数

#### 可选的内核命令行参数

在此处指定可选的内核参数，以启用/禁用系统功能、添加驱动程序，等等。

#### CPU 缓解措施

SUSE 针对为防御 CPU 边信道攻击而部署的所有软件缓解措施发布了一个或多个内核引导命令行参数。其中一些参数可能会导致性能损失。请根据您的设置选择以下其中一个选项，在安全性和性能之间取得平衡：

**自动：** 启用您的 CPU 型号所需的全部缓解措施，但不防范跨 CPU 线程的攻击。此设置可能会在一定程度上影响性能，具体视工作负载而定。

**自动 + 禁用 SMT：** 提供整套可用的安全缓解措施。启用您的 CPU 型号所需的全部缓解策略。此外，它还会禁用同步多线程 (SMT)，以免受到跨多个 CPU 线程发起的边信道攻击。此设置可能会进一步影响性能，具体取决于工作负载。

**关：** 禁用所有缓解措施。CPU 可能会遭到边信道攻击，具体取决于 CPU 型号。此设置不会对性能造成任何影响。

**手动：** 不设置任何缓解措施级别。请使用内核命令行选项手动指定 CPU 缓解措施。

### 使用图形控制台

如果选中该选项，引导菜单会显示在图形启动屏幕中，而不是以文本模式显示。默认情况下，系统会自动设置引导屏幕的分辨率，但您可以通过控制台分辨率手动进行设置。可以使用控制台主题文件选择器指定图形主题定义文件。请仅在您想要应用自己的定制主题时，才更改此设置。

### 使用串行控制台

如果您的计算机是通过串行控制台控制的，则可激活此选项并指定要使用的 COM 端口及其运行速度。请参阅 [info grub](#) 或 <http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal>。

## 18.4 有用的 GRUB 2 命令

### **grub2-mkconfig**

基于 [/etc/default/grub](#) 以及 [/etc/grub.d/](#) 中的脚本生成新的 [/boot/grub2/grub.cfg](#)。

#### 例 18.1：GRUB2-MKCONFIG 用法

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```



## 提示：语法检查

运行不带任何参数的 **grub2-mkconfig** 会将配置打印到 STDOUT 中供用户查看。

在写入 `/boot/grub2/grub.cfg` 后使用 **grub2-script-check** 可以检查其语法。



## 重要：grub2-mkconfig 无法修复 UEFI 安全引导表

如果您使用的是 UEFI 安全引导，而您的系统无法再正常连接 GRUB 2，您可能需要另外重新安装 Shim 并重新生成 UEFI 引导表。要执行此操作，请使用：

```
# shim-install --config-file=/boot/grub2/grub.cfg
```

## grub2-mkrescue

为安装的 GRUB 2 配置创建一个可引导的救援映像。

### 例 18.2：GRUB2-MKRESCUE 用法

```
grub2-mkrescue -o save_path/name.iso iso
```

## grub2-script-check

检查给定文件中有无语法错误。

### 例 18.3：GRUB2-SCRIPT-CHECK 用法

```
grub2-script-check /boot/grub2/grub.cfg
```

## grub2-once

仅设置下次引导的默认引导项。要获取可用引导项的列表，请使用 `--list` 选项。

### 例 18.4：GRUB2-ONCE 用法

```
grub2-once number_of_the_boot_entry
```



## 提示：grub2-once help

不使用任何选项调用该程序可以获取所有可用选项的完整列表。

## 18.5 更多信息

<https://www.gnu.org/software/grub/> 上提供了有关 GRUB 2 的大量信息。还请参见 [grub](#) 信息页面。您也可以在技术信息搜索 ( “) 中搜索关键字” GRUB 2<https://www.suse.com/> [support](#)，以获取有关特殊问题的信息。

# 19 `systemd` 守护程序

`systemd` 负责初始化系统，它的进程 ID 为 1。`systemd` 由内核直接启动，信号 9（该信号通常会终止进程）对它不起作用。所有其他程序或由 `systemd` 直接启动，或由它的其中一个子进程启动。`systemd` 取代了 System V init 守护程序，并且通过支持 init 脚本与 System V init 完全兼容。

`systemd` 的主要优点是通过并行化服务启动，大大加快了引导速度。另外，`systemd` 只在确实有需要时才会启动服务。即不会在系统引导时无条件启动守护程序，而是在第一次需要时才会启动。`systemd` 还支持内核控制组 (cgroup)、创建快照和恢复系统状态。有关更多细节，请参见 <http://www.freedesktop.org/wiki/Software/systemd/>。

## 19.1 `systemd` 概念

下面的部分解释了 `systemd` 背后的概念。

`systemd` 是适用于 Linux 的系统和会话管理器，它与 System V 及 LSB init 脚本兼容。`systemd` 的主要功能包括：

- 并行化能力
- 使用套接字及 D-Bus 激活来启动服务
- 按需启动守护程序
- 使用 Linux cgroup 跟踪进程
- 创建快照和恢复系统状态
- 维护挂载点和自动挂载点
- 实施精细的、基于事务依赖性的服务控制逻辑

## 19.1.1 单元文件

单元配置文件包含有关以下项目的信息：服务、套接字、设备、挂载点、自动挂载点、交换文件或分区、启动目标、监控的文件系统路径、受 systemd 控制和监管的计时器、临时系统状态快照、资源管理部分或一组外部创建的进程。

“单元文件”是 systemd 用于描述下列各项的通用术语：

- **服务：** 进程相关信息（例如运行守护程序）；文件以 .service 结尾
- **目标：** 用于将单元分组以及在启动期间用作同步点；文件以 .target 结尾
- **套接字：** IPC 或网络套接字或文件系统 FIFO 相关信息，适用于基于套接字的激活（如 inetd）；文件以 .socket 结尾
- **路径：** 用于触发其他单元（例如，在文件更改时运行服务）；文件以 .path 结尾
- **计时器：** 受控制计时器相关信息，适用于基于计时器的激活；文件以 .timer 结尾
- **挂载点：** 通常由 fstab 生成器自动生成；文件名以 .mount 结尾
- **自动挂载点：** 文件系统自动挂载点相关信息；文件以 .automount 结尾
- **交换：** 内存分页的交换设备或文件相关信息；文件以 .swap 结尾
- **设备：** sysfs/udev(7) 设备树中公开的设备相关信息；文件以 .device 结尾
- **范围/部分：** 有关分层管理一组进程的资源的概念；文件以 .scope/.slice 结尾

有关 systemd 单元文件的详细信息，请参见 <http://www.freedesktop.org/software/systemd/man/systemd.unit.html> ↗

## 19.2 基本用法

System V init 系统使用若干命令来处理服务：init 脚本、insserv、telinit 及其他。systemd 可简化服务管理，因为对于大部分处理服务的任务，只需要记住一条命令：systemctl。它使用“命令加子命令”表示法，与 git 或 zypper 相似：

```
systemctl GENERAL OPTIONS SUBCOMMAND SUBCOMMAND OPTIONS
```

有关完整的手册，请参见 man 1 systemctl。



## 提示：终端输出和 Bash 补全

如果输出传递到某个终端（而不是某个管道或文件），systemd 命令默认会将长输出发送到分页器。使用 --no-pager 选项可关闭分页模式。

systemd 还支持 bash 补全，允许您输入子命令的头几个字母，然后按 →。此功能只能在 bash 外壳中使用，并且需要安装 bash-completion 软件包。

### 19.2.1 管理正在运行的系统中的服务

用于管理服务的子命令与使用 System V init 管理服务的子命令相同（start、stop 等）。服务管理命令的一般语法如下所示：

#### systemd

```
systemctl reload|restart|start|status|stop|... MY_SERVICE(S)
```

#### System V init

```
rcMY_SERVICE(S) reload|restart|start|status|stop|...
```

systemd 允许您一次管理多个服务。不用像 System V init 要逐个执行 init 脚本，而是执行如下命令：

```
> sudo systemctl start MY_1ST_SERVICE MY_2ND_SERVICE
```

要列出系统上所有可用的服务，请运行：

```
> sudo systemctl list-unit-files --type=service
```

下表列出了 systemd 和 System V init 最重要的服务管理命令：

表 19.1：服务管理命令

| 任务  | <u>systemd</u> 命令 | <b>System V init</b> 命令 |
|-----|-------------------|-------------------------|
| 启动： | start             | start                   |
| 停止： | stop              | stop                    |

| 任务   | systemd 命令            | System V init 命令 |
|--|-----------------------|------------------|
| <b>重启：</b> 关闭服务，然后再启动这些服务。如果某项服务尚未运行，它将会启动。  | restart               | restart          |
| <b>有条件地重启：</b> 如果服务当前正在运行，则重启它们。对未在运行的服务不执行任何操作。   | try-restart           | try-restart      |
| <b>重新装载：</b> 让服务在不中断操作的情况下重新装载它们的配置文件。使用案例：让 Apache 重新装载修改过的 <code>httpd.conf</code> 配置文件。并非所有服务都支持重新装载。                         | reload                | reload           |
| <b>重新装载或重启：</b> 如果支持重新装载则重新装载服务，否则重启服务。如果某项服务尚未运行，它将会启动。   | reload-or-restart     | n/a              |
| <b>有条件地重新装载或重启：</b> 如果支持重新装载则重新装载服务，否则，如果服务当前正在运行，则重启服务。对未在运行的服务不执行任何操作。   | reload-or-try-restart | n/a              |
| <b>获得详细的状态信息：</b> 列出服务状态的相关信息。systemd 命令会显示说明、可执行文件、状态、cgroup 及服务发出的最新消息等细节（请参见第 19.6.9 节 “调试服务”）。使用 System V init 显示的详细程度因服务而异。 | status                | status           |
| <b>获得简要的状态信息：</b> 显示服务是否处于活动状态。  | is-active             | status           |

## 19.2.2 永久启用/禁用服务

上一节中提到的服务管理命令可让您操控当前会话的服务。`systemd` 还允许您永久启用或禁用服务，让它们在用户要求时自动启动或永远不可用。您可以使用 YaST 或在命令行上运行命令来实现此目的。

### 19.2.2.1 在命令行上启用/禁用服务

下表列出了 `systemd` 和 System V init 用于启用和禁用服务的命令：

#### ! 重要：服务启动

在命令行上启用服务时，服务不会自动启动。它会安排在下一次系统启动或运行级别/目标发生更改时启动。要在启用服务之后立即启动它，请明确运行 `systemctl start MY_SERVICE` 或 `rc MY_SERVICE start`。

表 19.2：用于启用和禁用服务的命令

| 任务                        | <code>systemd</code> 命令  | System V init 命令  |
|---------------------------|--|---|
| 启用：                       | <code>systemctl enable</code><br><code>MY_SERVICE(s)</code>          | <code>insserv</code><br><code>MY_SERVICE(s),</code><br><code>chkconfig -a</code><br><code>MY_SERVICE(s)</code>    |
| 禁用：                       | <code>systemctl disable</code><br><code>MY_SERVICE(s).service</code> | <code>insserv -r</code><br><code>MY_SERVICE(s),</code><br><code>chkconfig -d</code><br><code>MY_SERVICE(s)</code> |
| 检查： 显示某项服务是否启用。           | <code>systemctl is-enabled</code><br><code>MY_SERVICE</code>         | <code>chkconfig</code><br><code>MY_SERVICE</code>   |
| 重新启用： 与重启动服务相似，此命令先禁用服务，然 | <code>systemctl reenable</code><br><code>MY_SERVICE</code>           | 无   |

| 任务  | systemd 命令   | System V init 命令 |
|---|--|------------------|
| 后再启用该服务。可用来使用默认值重新启用服务。                                   |  |                  |
| <b>屏蔽：</b> “禁用” 某项服务之后，仍然可以手动启动它。要彻底禁用服务，您需要屏蔽它。须谨慎使用该功能。 | <code>systemctl mask MY_SERVICE</code>                   | 无                |
| <b>取消屏蔽：</b> 屏蔽某项服务之后，只有在将其取消屏蔽之后才能再次使用它。                 | <code>systemctl unmask</code><br><code>MY_SERVICE</code> | 无                |

## 19.3 系统启动和目标管理

启动和关闭系统的整个过程由 `systemd` 负责维护。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。

### 19.3.1 目标与运行级别的比较

使用 System V init 时，系统引导到所谓的“运行级别”。运行级别定义系统如何启动以及正在运行的系统中有哪些服务可用。运行级别是有编号的；最知名的运行级别是 0（关闭系统）、3（联网的多用户模式）和 5（联网并使用显示管理器的多用户模式）。

`systemd` 使用所谓的“目标单元”引入新的概念。不过，它仍然与运行级别概念完全兼容。目标单元用名称而不是编号来标识，并具有特定的作用。例如，目标 `local-fs.target` 和 `swap.target` 用于挂载本地文件系统和交换空间。

目标 `graphical.target` 提供联网并使用显示管理器的多用户系统，作用与运行级别 5 相当。复杂目标（如 `graphical.target`）通过将一部分其他目标组合起来充当“元”目标。`systemd` 通过组合现有目标简化了创建自定义目标的工作，从而提供了极大的灵活性。

下面的列表显示了最重要的 `systemd` 目标单元。有关完整列表，请参见 `man 7 systemd.special`。

### default.target

默认引导的目标。这并不是“实际”目标，而是指向另一个目标（如 `graphic.target`）的符号链接，可通过 YaST 永久更改（请参见第 19.4 节“[使用 YaST 管理服务](#)”）。要为某个会话更改它，请在引导提示处使用内核参数 `systemd.unit=MY_TARGET.target`。

### emergency.target

在控制台上启动紧急外壳。请仅在引导提示处按以下格式使用它：`systemd.unit=emergency.target`。

### graphical.target

启动联网的具有多用户支持和显示管理器的系统。

### halt.target

关闭系统。

### mail-transfer-agent.target

启动发送和接收邮件所需的所有服务。

### multi-user.target

启动联网的多用户系统。

### reboot.target

重新引导系统。

### rescue.target

启动不联网的单用户系统。

为了与 System V init 运行级别系统保持兼容，`systemd` 提供了名为 `runlevelX.target` 的特殊目标，与编号为 X 的相应运行级别相对应。

如果您想知道当前的目标，请使用命令：`systemctl get-default`

表 19.3：SYSTEM V 运行级别和 `systemd` 目标单元

| System V 运行级别 | <code>systemd target</code>   | 用途                 |
|---------------|---|--------------------|
| 0             | <code>runlevel0.target</code> 、<br><code>halt.target</code> 、<br><code>poweroff.target</code> | 系统关闭               |
| 1、S           | <code>runlevel1.target</code> 、<br><code>rescue.target</code>                                 | 单用户模式              |
| 2             | <code>runlevel2.target</code> 、 <code>multi-user.target</code>                                | 无远程联网的本地多用户模式      |
| 3             | <code>runlevel3.target</code> 、 <code>multi-user.target</code>                                | 完整的联网多用户模式         |
| 4             | <code>runlevel4.target</code>   | 未使用/用户定义           |
| 5             | <code>runlevel5.target</code> 、<br><code>graphical.target</code>                              | 联网并使用显示管理器的完整多用户模式 |
| 6             | <code>runlevel6.target</code> 、<br><code>reboot.target</code>                                 | 系统重引导              |

## ! 重要：`systemd` 会忽略 `/etc/inittab`

System V init 系统中的运行级别在 `/etc/inittab` 中配置。`systemd` 不使用此配置。

有关如何创建您自己的可引导目标的指导，请参考[第 19.5.4 节 “创建自定义目标”](#)。

### 19.3.1.1 用于更改目标的命令

可使用下列命令来操作目标单元：

| 任务              | systemd 命令  | System V init 命令   |
|-----------------|---|--|
| 更改当前目标/运行级别     | <code>systemctl isolate MY_TARGET.target</code>   | <code>telinit X</code>   |
| 更改为默认目标/运行级别    | <code>systemctl default</code>  | 无  |
| 获得当前目标/运行级别     | <code>systemctl list-units --type=target</code><br>对于 <code>systemd</code> ，一般会有多个活动目标。该命令可列出当前处于活动状态的所有目标。   | <code>who -r</code><br>或<br><code>runlevel</code>                |
| 永久更改默认运行级别      | 使用服务管理器或运行以下命令：<br><code>ln -sf /usr/lib/systemd/system/MY_TARGET.target /etc/systemd/system/default.target</code>  | 使用服务管理器或更改以下行<br><code>id: X:initdefault: in /etc/inittab</code> |
| 更改当前引导进程的默认运行级别 | 在引导提示处输入以下选项<br><code>systemd.unit=MY_TARGET.target</code>  | 在引导提示处输入所需的运行级别编号。   |
| 显示目标/运行级别的依赖项   | <code>systemctl show -p "Requires"</code><br><code>MY_TARGET.target</code><br><code>systemctl show -p "Wants"</code><br><code>MY_TARGET.target</code><br>“Requires” 会列出硬性依赖项（必须解决的依赖项），而 “Wants” 会列出软性依赖项（情况允许时解决的依赖项）。 | 无  |

## 19.3.2 调试系统的启动

`systemd` 提供了分析系统启动过程的方法。您可以查看所有服务及其状态的列表（而不必分析 `/var/log/`）。`systemd` 还允许您扫描启动过程，以了解每项服务启动用了多长时间。

### 19.3.2.1 查看服务的启动情况

要查看系统引导后所启动服务的完整列表，请输入命令 `systemctl`。该命令会列出所有活动服务，如下所示（已精简）。要获得特定服务的详细信息，请使用 `systemctl status` `MY_SERVICE`。

#### 例 19.1：列出活动服务

```
# systemctl
UNIT                  LOAD   ACTIVE SUB      JOB DESCRIPTION
[...]
iscsi.service          loaded  active exited  Login and scanning of iSC+
kmod-static-nodes.service loaded  active exited  Create list of required s+
libvирtd.service      loaded  active running  Virtualization daemon
nscd.service           loaded  active running  Name Service Cache Daemon
chronyd.service        loaded  active running  NTP Server Daemon
polkit.service          loaded  active running  Authorization Manager
postfix.service         loaded  active running  Postfix Mail Transport Ag+
rc-local.service        loaded  active exited  /etc/init.d/boot.local Co+
rsyslog.service         loaded  active running  System Logging Service
[...]
LOAD    = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

161 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

如果只想列出无法启动的服务，请使用 `--failed` 选项：

#### 例 19.2：列出失败的服务

```
# systemctl --failed
```

| UNIT                   | LOAD   | ACTIVE | SUB    | JOB | DESCRIPTION               |
|------------------------|--------|--------|--------|-----|---------------------------|
| apache2.service        | loaded | failed | failed |     | apache                    |
| NetworkManager.service | loaded | failed | failed |     | Network Manager           |
| plymouth-start.service | loaded | failed | failed |     | Show Plymouth Boot Screen |
| [...]                  |        |        |        |     |                           |

### 19.3.2.2 调试启动时间

为了调试系统启动时间, `systemd` 提供了 **systemd-analyze** 命令。该命令会显示总启动时间及按启动时间排序的服务列表, 还可以生成 SVG 图, 以显示各服务相对于其他服务的启动时间。

#### 列出系统启动时间

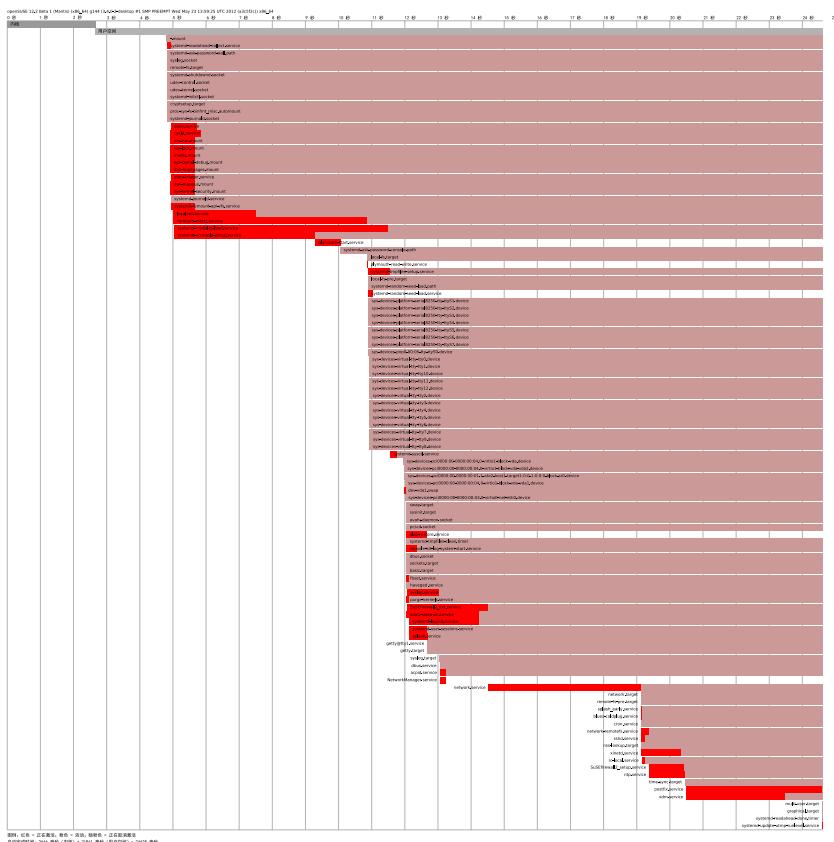
```
# systemd-analyze
Startup finished in 2666ms (kernel) + 21961ms (userspace) = 24628ms
```

#### 列出服务启动时间

```
# systemd-analyze blame
15.000s backup-rpmdb.service
14.879s mandb.service
 7.646s backup-sysconfig.service
 4.940s postfix.service
 4.921s logrotate.service
 4.640s libvirtd.service
 4.519s display-manager.service
 3.921s btrfsmaintenance-refresh.service
 3.466s lvm2-monitor.service
 2.774s plymouth-quit-wait.service
 2.591s firewalld.service
 2.137s initrd-switch-root.service
 1.954s ModemManager.service
 1.528s rsyslog.service
 1.378s apparmor.service
[...]
```

## 服务启动时间图

```
# systemd-analyze plot > jupiter.example.com-startup.svg
```



### 19.3.2.3 查看完整的启动过程

上面的命令列出了已启动的服务及其启动时间。如需更详细的概述，请在引导提示符处指定以下参数，以指示 systemd 创建完整启动过程的详细日志。

```
systemd.log_level=debug systemd.log_target=kmsg
```

现在，systemd 会将日志消息写入内核环缓冲区。可以使用 dmesg 查看该缓冲区：

```
> dmesg -T | less
```

### 19.3.3 System V 兼容性

systemd 与 System V 兼容，因此，您仍可以使用现有的 System V init 脚本。但是，至少有一个已知问题会导致 System V init 脚本不能立即与 systemd 配合使用：通过 init 脚本中的 **su** 或 **sudo** 以其他用户身份启动服务会导致脚本失败，生成“访问被拒绝”错误。

使用 **su** 或 **sudo** 更改用户时，会启动 PAM 会话。完成 init 脚本后会终止此会话。因此，init 脚本启动的服务也会终止。要解决此问题，请执行以下步骤：

1. 创建与 init 脚本同名、文件扩展名为 **.service** 的服务文件封装程序：

```
[Unit]
Description=DESCRIPTION
After=network.target

[Service]
User=USER
Type=forking ①
PIDFile=PATH TO PID FILE ①
ExecStart=PATH TO INIT SCRIPT start
ExecStop=PATH TO INIT SCRIPT stop
ExecStopPost=/usr/bin/rm -f PATH TO PID FILE ①

[Install]
WantedBy=multi-user.target ②
```

将 UPPERCASE LETTERS 中写入的所有值替换为适当的值。

- ① 可选 — 仅当 init 脚本启动守护程序时才使用。
- ② multi-user.target 在系统引导进入 graphical.target 时也会启动 init 脚本。如果只应在引导进入显示管理器时才将它启动，请在此处使用 graphical.target。

2. 使用 **systemctl start APPLICATION** 启动守护程序。

## 19.4 使用 YaST 管理服务

基本服务管理也可以通过 YaST 服务管理器模块实现。该模块支持启动、停止、启用和禁用服务。它还可让您显示服务的状态以及更改默认目标。要启动 YaST 模块，请选择 YaST > 系统 > 服务管理器。



图 19.1：服务管理器

### 更改默认系统目标

要更改系统引导到的目标，请从默认系统目标下拉框中选择某个目标。最常用的目标是图形界面（启动图形登录屏幕）和多用户（以命令行模式启动系统）。

### 启动或停止服务

从表中选择一个服务。状态列显示它当前是（活动）否（非活动）正在运行。通过选择启动或停止可切换其状态。

为当前正在运行的会话启动或停止服务会更改其状态。要更改服务在整个重引导过程中的状态，您需要启用或禁用服务。

### 定义服务启动行为

服务可以在引导时自动启动，也可以手动启动。从表中选择一个服务。启动列显示它当前是手动启动还是引导时启动。通过选择启动模式可切换其状态。

要在当前会话中更改服务状态，您需要按照上述方式启动或停止服务。

## 查看状态消息

要查看某个服务的状态消息，请从列表中选择该服务，然后选择显示细节。您看到的输出与 `systemctl -l status MY_SERVICE` 命令生成的输出完全相同。

# 19.5 自定义 `systemd`

以下几节介绍了 `systemd` 自定义的一些示例。



## 警告：防止您的自定义被覆盖

在自定义 `systemd` 时，请务必使用目录 `/etc/systemd/`，切勿使用 `/usr/lib/systemd/`。否则，您的更改将在 `systemd` 下次更新时被覆盖。

## 19.5.1 自定义单元文件

建议使用 `systemctl edit SERVICE` 命令来自定义单元文件。此命令会启动默认的文本编辑器，并在 `/etc/systemd/system/NAME.service.d/` 中创建一个包含 `override.conf` 文件的目录。该命令还确保会向正在运行的 `systemd` 进程通知这些更改。

或者，您可以通过运行 `systemctl edit --full SERVICE` 打开原始文件的副本进行编辑，而不是打开空白文件。编辑文件时，请确保未去除任何现有部分。

下面将练习如何更改系统等待 MariaDB 启动的时间。以 root 身份运行 `systemctl edit --full mariadb.service`。打开的文件如下所示：

```
[Unit]
Description=MySQL server
Wants=basic.target
Conflicts=mariadb.target
After=basic.target network.target

[Install]
WantedBy=multi-user.target
Alias=mysql.service
```

```
[Service]
Restart=on-abort
Type=notify
ExecStartPre=/usr/lib/mysql/mysql-systemd-helper install
ExecStartPre=/usr/lib/mysql/mysql-systemd-helper upgrade
ExecStart=/usr/lib/mysql/mysql-systemd-helper start

# Configures the time to wait for start-up/stop
TimeoutSec=300

# Prevent writes to /usr, /boot, and /etc
ProtectSystem=full

# Prevent accessing /home, /root and /run/user
ProtectHome=true

UMask=007
```

调整 `TimeoutSec` 值并保存更改。要启用更改，请以 root 身份运行 `systemctl daemon-reload`。

有关详细信息，请参考可使用 `man 1 systemctl` 命令调用的手册页。

## 19.5.2 创建插入式文件

要对配置文件进行细微更改，请使用所谓的插入式文件。使用插入式文件，您无需编辑或覆盖单元文件本身，即可扩展单元文件的配置。

例如，要更改位于 `/usr/lib/systemd/system/FOOBAR.service` 中 `FOOBAR` 服务的单个值，请按以下步骤操作：

1. 创建名为 `/etc/systemd/system/FOOBAR.service.d/` 的目录。

注意 `.d` 后缀。该目录必须命名为与要用插入式文件修补的服务相似的名称。

2. 在该目录中，创建 `your_modification.conf` 文件。

确保文件中仅包含要修改的值所在的那一行。

3. 将更改保存到文件中。



## 注意：避免名称冲突

为避免插入式文件与 SUSE 自带的文件之间出现名称冲突，建议在所有插入式文件名前加上一个两位数和一个短划线：例如，`80-override.conf`。

以下范围为预留值范围：

- `0-19`，为 `systemd` 上游预留
- `20-25`，为 SUSE 自带的 `systemd` 预留
- `26-29`，为 SUSE 软件包（`systemd` 除外）预留
- `50`，为使用 `systemctl set-property` 创建的插入式文件预留

请使用大于此范围的两位数，以确保 SUSE 自带的任何插入式文件都不会覆盖您自己的插入式文件。

您可以使用 `systemctl cat $UNIT` 来列出并校验哪些文件考虑纳入单元配置。

### 19.5.3 将 `xinetd` 服务转换为 `systemd`

从 SUSE Linux Enterprise Desktop 15 版本开始，去除了 `xinetd` 基础架构。本节概述如何将现有的自定义 `xinetd` 服务文件转换为 `systemd` 套接字。

对于每个 `xinetd` 服务文件，您至少要有两个 `systemd` 单元文件：套接字文件 (`*.socket`) 和关联的服务文件 (`*.service`)。套接字文件告诉 `systemd` 要创建哪个套接字，服务文件告诉 `systemd` 要启动哪个可执行文件。

以下面的 `xinetd` 服务文件为例：

```
# cat /etc/xinetd.d/example
service example
{
    socket_type = stream
    protocol = tcp
    port = 10085
    wait = no
    user = user
```

```
group = users
groups = yes
server = /usr/libexec/example/exampled
server_args = -auth=bsdtcp exampledump
disable = no
}
```

要将它转换为 systemd，需要下面两个配套的文件：

```
# cat /usr/lib/systemd/system/example.socket
[Socket]
ListenStream=0.0.0.0:10085
Accept=false

[Install]
WantedBy=sockets.target
```

```
# cat /usr/lib/systemd/system/example.service
[Unit]
Description=example

[Service]
ExecStart=/usr/libexec/example/exampled -auth=bsdtcp exampledump
User=user
Group=users
StandardInput=socket
```

有关 systemd “socket” 和 “service” 文件选项的完整列表，请参见 systemd.socket 和 systemd.service 手册页 (man 5 systemd.socket、man 5 systemd.service) 。

#### 19.5.4 创建自定义目标

System V init SUSE 系统上未使用运行级别 4，以便允许管理员创建自己的运行级别配置。systemd 允许您创建任意数目的自定义目标。建议您从采用 graphical.target 等现有目标开始。

1. 将配置文件 /usr/lib/systemd/system/graphical.target 复制到 /etc/systemd/system/MY\_TARGET.target，并根据需要调整该文件。

2. 上一步中复制的配置文件已涵盖目标的必要（“硬性”）依赖项。如果还要涵盖需要的（“软性”）依赖项，请创建目录 `/etc/systemd/system/MY_TARGET.target.wants`。
3. 对每个需要的服务，创建从 `/usr/lib/systemd/system` 指向 `/etc/systemd/system/MY_TARGET.target.wants` 的符号链接。
4. 设置好目标后，重新装载 `systemd` 配置以使新目标可用：

```
> sudo systemctl daemon-reload
```

## 19.6 高级用法

以下几节介绍了适合系统管理员的高级主题。有关更高级的 `systemd` 文档，请参考 Lennart Pöttering 撰写的适用于管理员的 `systemd` 相关系列，网址为 <http://0pointer.de/blog/projects>。

### 19.6.1 清理临时目录

`systemd` 支持定期清理临时目录。将会自动迁移并激活前一系统版本中的配置。`tmpfiles.d`（负责管理临时文件）会从 `/etc/tmpfiles.d/*.conf`、`/run/tmpfiles.d/*.conf` 和 `/usr/lib/tmpfiles.d/*.conf` 文件中读取其配置。`/etc/tmpfiles.d/*.conf` 中的配置会覆盖其他两个目录中的相关配置（`/usr/lib/tmpfiles.d/*.conf` 是软件包用于存储其配置文件的位置）。

配置格式为每个路径一行，该行包含操作与路径、（可选）模式、所有权、期限和参数字段，具体取决于操作。以下示例将取消链接 X11 锁定文件：

| Type | Path                            | Mode | UID | GID | Age | Argument |
|------|---------------------------------|------|-----|-----|-----|----------|
| r    | <code>/tmp/.X[0-9]*-lock</code> |      |     |     |     |          |

要获取 `tmpfile` 计时器的状态：

```
> sudo systemctl status systemd-tmpfiles-clean.timer
systemd-tmpfiles-clean.timer - Daily Cleanup of Temporary Directories
  Loaded: loaded (/usr/lib/systemd/system/systemd-tmpfiles-clean.timer; static)
```

```
Active: active (waiting) since Tue 2018-04-09 15:30:36 CEST; 1 weeks 6 days ago
Docs: man:tmpfiles.d(5)
      man:systemd-tmpfiles(8)

Apr 09 15:30:36 jupiter systemd[1]: Starting Daily Cleanup of Temporary
Directories.
Apr 09 15:30:36 jupiter systemd[1]: Started Daily Cleanup of Temporary
Directories.
```

有关临时文件处理的详细信息，请参见 [man 5 tmpfiles.d](#)。

## 19.6.2 系统日志

[第 19.6.9 节 “调试服务”](#) 介绍如何查看给定服务的日志消息。但可显示的日志消息并不仅限于服务日志。您还可以访问和查询 `systemd` 写入的完整日志，即所谓的“日志”。使用 `journalctl` 命令可显示完整的日志消息，从最早的项开始。有关诸如应用过滤器或更改输出格式的选项，请参考 [man 1 journalctl](#)。

## 19.6.3 快照

您可以使用 `isolate` 子命令将 `systemd` 的当前状态保存到指定快照中，并在日后还原到该状态。此功能在测试服务或自定义目标时非常有用，因为它可让您随时回到定义的状态。快照仅在当前会话中可用，重引导时将自动删除。快照名称必须以 `.snapshot` 结尾。

### 创建快照

```
> sudo systemctl snapshot MY_SNAPSHOT.snapshot
```

### 删除快照

```
> sudo systemctl delete MY_SNAPSHOT.snapshot
```

### 查看快照

```
> sudo systemctl show MY_SNAPSHOT.snapshot
```

### 激活快照

```
> sudo systemctl isolate MY_SNAPSHOT.snapshot
```

## 19.6.4 装载内核模块

systemd 结合 /etc/modules-load.d 中的配置文件，可以在系统引导时自动装载内核模块。该文件应该命名为 MODULE.conf 并包含以下内容：

```
# load module MODULE at boot time
MODULE
```

如果某个软件包安装了用于装载内核模块的配置文件，该文件将安装到 /usr/lib/modules-load.d。如果存在两个同名的配置文件，将以 /etc/modules-load.d 中的配置文件为准。有关更多信息，请参见 modules-load.d(5) 手册页。

## 19.6.5 装载服务之前执行操作

使用 System V init 时，需在 /etc/init.d/before.local 中指定需要在装载服务之前执行的操作。systemd 不再支持此过程。如果您需要在启动服务之前执行操作，请执行以下步骤：

### 装载内核模块

在 /etc/modules-load.d 目录中创建一个插入式文件（有关语法，请参见 man modules-load.d）

### 创建文件或目录、清理目录、更改所有权

在 /etc/tmpfiles.d 中创建一个插入式文件（有关语法，请参见 man tmpfiles.d）

### 其他任务

基于下面的模板创建一个系统服务文件，例如 /etc/systemd/system/before.service：

```
[Unit]
Before=NAME OF THE SERVICE YOU WANT THIS SERVICE TO BE STARTED BEFORE
[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=YOUR_COMMAND
# beware, executable is run directly, not through a shell, check the man
pages
```

```
# systemd.service and systemd.unit for full syntax
[Install]
# target in which to start the service
WantedBy=multi-user.target
#WantedBy=graphical.target
```

创建服务文件后，应运行以下命令（以 root 身份）：

```
> sudo systemctl daemon-reload
> sudo systemctl enable before
```

每次修改服务文件时，都需要运行：

```
> sudo systemctl daemon-reload
```

## 19.6.6 内核控制组 (cgroup)

在传统 System V init 系统上，并不总是能够明确地将某个进程指派给生成它的服务。一些服务（例如 Apache）会生成大量第三方进程（例如 CGI 或 Java 进程），这些进程本身又会生成更多进程。这使得明确指派变得非常困难，甚至无法做到。另外，服务可能不会正常结束，残留某些子进程保持运行状态。

systemd 通过将每个服务放入它自己的 cgroup 中，解决了这个问题。cgroups 是一项内核功能，允许将进程及其所有子进程聚合到分层组织的组中。systemd 根据每个 cgroup 的服务名称命名各 cgroup。因为非特权进程不允许“离开”它的 cgroup，这提供了一种行之有效的方式，可通过服务名称来标记该服务生成的所有进程。

要列出属于某个服务的所有进程，请使用命令 **systemd-cgls**。结果将如下所示（已精简）：

例 19.3：列出属于某个服务的所有进程

```
# systemd-cgls --no-pager
└─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 20
└─user.slice
  └─user-1000.slice
    ├─session-102.scope
    |  ├─12426 gdm-session-worker [pam/gdm-password]
    |  ├─15831 gdm-session-worker [pam/gdm-password]
    |  ├─15839 gdm-session-worker [pam/gdm-password]
```

```
|  |  |-15858 /usr/lib/gnome-terminal-server
[...]
└system.slice
  ├systemd-hostnamed.service
  |└17616 /usr/lib/systemd/systemd-hostnamed
  ├cron.service
  |└1689 /usr/sbin/cron -n
  ├postfix.service
  |└1676 /usr/lib/postfix/master -w
  |└1679 qmgr -l -t fifo -u
  |└15590 pickup -l -t fifo -u
  ├sshd.service
  |└1436 /usr/sbin/sshd -D
[...]
```

有关 cgroups 的更多信息，请参见《系统分析和微调指南》，第 9 章 “内核控制组”。

### 19.6.7 终止服务（发送信号）

如第 19.6.6 节 “内核控制组 (cgroup)” 中所述，在 System V init 系统中，并不总是能够将某个进程指派给其父服务。这导致难以停止服务及其子进程。尚未终止的子进程会一直保持为僵停状态。

systemd 的概念是将每个服务限定在某个 cgroup 中，从而可以识别一个服务的所有子进程，因此可让您将信号发送给这些进程中的每一个进程。您可使用 systemctl kill 向服务发送信号。有关可用信号的列表，请参考 man 7 signals。

#### 向服务发送 SIGTERM

SIGTERM 是发送的默认信号。

```
> sudo systemctl kill MY_SERVICE
```

#### 向服务发送 SIGNAL

可使用 -s 选项指定应该发送的信号。

```
> sudo systemctl kill -s SIGNAL MY_SERVICE
```

## 选择进程

默认情况下，**kill** 命令会向指定 cgroup 的 all 进程发送信号。您可以将发送目标限制为 control 或 main 进程。后一个选项可用于通过发送 SIGHUP 强制服务重新装载其配置的情况：

```
> sudo systemctl kill -s SIGHUP --kill-who=main MY_SERVICE
```

## 19.6.8 有关 D-Bus 服务的重要说明

D-Bus 服务是 systemd 客户端与作为 pid 1 运行的 systemd 管理器之间进行通讯的消息总线。虽然 dbus 是个独立的守护程序，但它也是 init 基础架构的组成部分。

在正在运行的系统中终止或重启 dbus 的效果类似于尝试终止或重启 pid 1。此操作将中断 systemd 客户端与服务器间的通讯，并使大部分 systemd 功能不可用。

因此，不建议也不支持终止或重启 dbus。

更新 dbus 或与 dbus 有关的软件包需要重引导系统。如果不确定是否需要重引导，请运行 **sudo zypper ps -s**。如果 dbus 显示在所列服务之中，则表明您需要重引导系统。

请记住，即使自动更新配置为跳过需要重引导的软件包，dbus 仍会更新。

## 19.6.9 调试服务

默认情况下，systemd 的输出不会太详细。如果服务启动成功，则不会产生任何输出。如果服务启动失败，则会显示简短的错误消息。不过，**systemctl status** 提供了调试服务的启动和操作的途径。

systemd 附带了自己的日志记录机制（“日志”）来记录系统消息。这可让您一并显示服务消息与状态消息。**status** 命令的工作方式与 **tail** 相似，也可以采用不同的格式显示日志消息，是一个功能强大的调试工具。

### 显示服务启动失败消息

每当服务启动失败时，使用 **systemctl status MY\_SERVICE** 可获得详细的错误消息：

```
# systemctl start apache2
Job failed. See system journal and 'systemctl status' for details.
# systemctl status apache2
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
     Active: failed (Result: exit-code) since Mon, 04 Apr 2018 16:52:26
+0200; 29s ago
   Process: 3088 ExecStart=/usr/sbin/start_apache2 -D SYSTEMD -k start
   (code=exited, status=1/FAILURE)
     CGroup: name=systemd:/system/apache2.service

Apr 04 16:52:26 g144 start_apache2[3088]: httpd2-prefork: Syntax error on
line
205 of /etc/apache2/httpd.conf: Syntax error on li...alHost>
```

## 显示最后 N 条服务消息

status 子命令的默认行为是显示服务发出的最后 10 条消息。如果要更改要显示的消息数目，请使用 --lines=N 参数：

```
> sudo systemctl status chronyd
> sudo systemctl --lines=20 status chronyd
```

## 以附加模式显示服务消息

要显示服务消息的“实时流”，请使用 --follow 选项，效果与 tail -f 相似：

```
> sudo systemctl --follow status chronyd
```

## 消息输出格式

--output=MODE 参数可让您更改服务消息的输出格式。最重要的可用模式包括：

### short

默认格式。显示日志消息，以及用户能看懂的时戳。

### verbose

所有字段的完整输出。

### cat

精简输出，不含时戳。

## 19.7 `systemd` 计时器单元

与 cron 类似, `systemd` 计时器单元提供了一种在 Linux 上安排作业的机制。虽然 `systemd` 计时器单元的用途与 cron 相同, 但其具备几项优点。

- 使用计时器单元安排的作业可以依赖其他 `systemd` 服务。
- 计时器单元被视为常规 `systemd` 服务, 因此可以使用 `systemctl` 进行管理。
- 计时器可以是实时计时器, 也可以是单调计时器。
- 时间单元会被记录到 `systemd` 日志中, 使得更易于监视和查错。

`systemd` 计时器单元通过 `.timer` 文件扩展名标识。

### 19.7.1 `systemd` 计时器类型

计时器单元可以使用单调和实时计时器。

- 与 cron 作业类似, 实时计时器是根据日历事件触发的。实时计时器使用选项 `OnCalendar` 进行定义。
- 单调计时器将从某个特定起点开始经过指定时间后触发。后者可以是系统引导或系统单元激活事件。有几个选项可用于定义单调计时器, 包括 `OnBootSec`、`OnUnitActiveSec` 和 `OnTypeSec`。单调计时器不是永久的, 每次重引导后都将被重置。

### 19.7.2 `systemd` 计时器和服务单元

每个计时器单元都必须有一个其所控制的对应 `systemd` 单元文件。也就是说, 将由 `.timer` 文件激活并管理对应的 `.service` 文件。与计时器搭配使用时, `.service` 文件不需要 `[Install]` 部分, 因为该服务将由计时器管理。

### 19.7.3 实际示例

为了理解 `systemd` 计时器单元的基本知识, 我们设置了一个触发 `foo.sh` 外壳脚本的计时器。

第一步是创建用于控制外壳脚本的 `systemd` 服务单元。为此，请打开一个新的文本文件进行编辑，并添加以下服务单元定义：

```
[Unit]
Description="Foo shell script"

[Service]
ExecStart=/usr/local/bin/foo.sh
```

将文件以名称 `foo.service` 保存到目录 `/etc/systemd/system/` 中。

接下来，打开一个新的文本文件进行编辑，并添加以下计时器定义：

```
[Unit]
Description="Run foo shell script"

[Timer]
OnBootSec=5min
OnUnitActiveSec=24h
Unit=foo.service

[Install]
WantedBy=multi-user.target
```

上述示例中的 `[Timer]` 部分指定了要触发的服务 (`foo.service`) 以及触发时间。在本例中，选项 `OnBootSec` 指定了一个在系统引导后五分钟触发服务的单调计时器，而选项 `OnUnitActiveSec` 会在服务被激活后 24 小时触发服务（即计时器将每天触发一次服务）。最后，选项 `WantedBy` 指定应在系统达到多用户目标时启动计时器。

您也可以不指定单调计时器，而是使用选项 `OnCalendar` 指定实时计时器。下面的实时计时器定义从星期一 12:00 开始，每周触发一次相关服务单元。

```
[Timer]
OnCalendar=weekly
Persistent=true
```

选项 `Persistent=true` 表示如果计时器错过了上一次启动时间（例如，由于系统关闭），则会在计时器激活后立即触发服务。

使用选项 `OnCalendar` 并采用以下格式还可以定义触发服务的具体日期和时间：`DayOfWeek`  
`Year-Month-Day Hour:Minute:Second`。下面的示例会在每天早上 5 点触发服务：

```
OnCalendar=*-*-* 5:00:00
```

您可以使用星号指定任何值，使用逗号列出可能的值。使用由 .. 分隔的两个值可表示一个连续的范围。下面的示例会在每个月的星期五下午 6 点触发服务：

```
OnCalendar=Fri *-*-1..7 18:00:00
```

要在不同的时间触发服务，您可以指定多个 `OnCalendar` 项：

```
OnCalendar=Mon..Fri 10:00  
OnCalendar=Sat,Sun 22:00
```

在上面的示例中，会在工作日的上午 10 点和周末的晚上 10 点触发服务。

编辑好计时器单元文件后，使用名称 `foo.timer` 将其保存到 `/etc/systemd/system/` 目录中。要检查所创建的单元文件是否正确，请运行以下命令：

```
> sudo systemctl-analyze verify /etc/systemd/system/foo.*
```

如果命令未返回任何输出，则表示文件成功通过了校验。

要启动计时器，请使用命令 `sudo systemctl start foo.timer`。要在引导时启用计时器，请运行命令 `sudo systemctl enable foo.timer`。

#### 19.7.4 管理 `systemd` 计时器

由于计时器被视为常规 `systemd` 单元，因此您可以使用 `systemctl` 来管理它们。您可以使用 `systemctl start` 启动计时器，使用 `systemctl enable` 启用计时器，依此类推。此外，您还可以使用命令 `systemctl list-timers` 列出所有活动计时器。要列出所有计时器（包括非活动计时器），请运行命令 `systemctl list-timers --all`。

### 19.8 更多信息

有关 `systemd` 的详细信息，请参考以下联机资源：

## 主页

<http://www.freedesktop.org/wiki/Software/systemd> ↗

## 面向管理员的 systemd

Lennart Pöttering 是 systemd 的作者之一，他撰写了一系列博客文章（写本章时已有 13 篇）。它们可在 <http://0pointer.de/blog/projects> ↗ 找到。

# III 系统

- 20 64 位系统环境中的 32 位和 64 位应用程序 278
- 21 **journalctl**: 查询 `systemd` 日志 280
- 22 **update-alternatives**: 管理命令和文件的多个版本 288
- 23 基本网络知识 296
- 24 打印机操作 357
- 25 图形用户界面 371
- 26 使用 FUSE 访问文件系统 390
- 27 安装多个内核版本 392
- 28 管理内核模块 399
- 29 使用 `udev` 进行动态内核设备管理 403
- 30 特殊系统功能 415
- 31 使用 `NetworkManager` 427

# 20 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。开发商尚未将所有 32 位应用程序移植到 64 位系统。本章简要概述了如何在 64 位 SUSE Linux Enterprise Desktop 平台上实现 32 位支持。

用于 64 位平台 AMD64 和 Intel 64 的 SUSE Linux Enterprise Desktop 可让现有的 32 位应用程序“无需额外配置”即可在 64 位环境中运行。这种支持意味着您可以继续使用首选的 32 位应用程序，而无需等待对应的 64 位端口可用。



## 注意：不支持构建 32 位应用程序

SUSE Linux Enterprise Desktop 不支持编译 32 位应用程序。它只为 32 位二进制文件提供运行时支持。

## 20.1 运行时支持



### 重要：不同应用程序版本之间的冲突

如果某个应用程序同时适用于 32 位和 64 位环境，安装这两个版本可能会导致出现问题。在这种情况下，请决定要安装哪一个版本，以避免潜在的运行时错误。

此规则的一个例外是 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 充当用户和应用程序之间的中间层。在同时运行了 32 位应用程序的 64 位操作系统上，请始终安装 32 位和 64 位 PAM 版本。

每个应用程序都需要一系列的库才能正常执行。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。

为了与 32 位版本保持兼容，64 位和 32 位库将存储在同一位置。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称为 lib64 目录中。以往通常位于 /lib 和 /usr/lib 下的 64 位对象文件现在位于 /lib64 和 /usr/lib64 下。这意味着 /lib 和 /usr/lib 下有可用于存储 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 `/lib` 下的 32 位子目录的数据内容与字词大小无关，则不会移动这些子目录。此方案符合 LSB (Linux 标准库) 和 FHS (文件系统层次标准)。

## 20.2 内核规范

适用于 AMD 64/Intel 64 的 64 位内核提供了 64 位和 32 位两种内核 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 位和 64 位应用程序与 64 位内核的通讯方式是相同的。

针对 64 位内核的 32 位系统调用仿真不支持系统程序使用的某些 API。这取决于平台。因此，必须对 `lspci` 之类的一些应用程序进行编译。

64 位内核只能装载 64 位内核模块。必须专门为 64 位内核编译 64 位模块。无法将 32 位内核模块与 64 位内核搭配使用。



### 提示：内核可装载模块

某些应用程序需要单独的内核可装载模块。如果您想要在 64 位系统环境中使用 32 位应用程序，请与应用程序供应商和 SUSE 联系。确保 64 位版本的内核可装载模块以及 32 位编译版本的内核 API 适用于此模块。

## 21 **journalctl**: 查询 **systemd** 日志

**systemd** 具有自己的日志记录系统，称作**日志**。由于所有系统事件都会写入日志，因此不需要运行基于 syslog 的服务。

日志本身是 systemd 管理的系统服务，全名为 systemd-journald.service。它会根据从内核、用户进程、标准输入和系统服务错误收到的日志记录信息，维护结构化的索引日志，并以此方式来收集和存储日志记录数据。systemd-journald 服务默认处于启用状态。

```
> sudo systemctl status systemd-journald
systemd-journald.service - Journal Service
  Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static)
  Active: active (running) since Mon 2014-05-26 08:36:59 EDT; 3 days ago
    Docs: man:systemd-journald.service(8)
          man:journald.conf(5)
  Main PID: 413 (systemd-journal)
    Status: "Processing requests..."
  CGroup: /system.slice/systemd-journald.service
          └─413 /usr/lib/systemd/systemd-journald
[...]
```

### 21.1 将日志设为永久

默认情况下，日志在 /run/log/journal/ 中存储日志数据。由于 /run/ 目录本质上具有易失性，因此，在重引导时会丢失日志数据。要永久保存日志数据，创建目录 /var/log/journal/ 并确保它具有正确的访问模式和所有权，以便 systemd-journald 服务能够存储其数据。要切换到永久日志记录，请执行以下命令：

```
> sudo mkdir /var/log/journal
> sudo systemd-tmpfiles --create --prefix=/var/log/journal
> sudo journalctl --flush
```

存储在 /run/log/journal/ 中的任何日志数据都将刷新到 /var/log/journal/ 中。

## 21.2 **journalctl**: 有用的开关

本节介绍了一些可用来增强 **journalctl** 默认行为的常见有用选项。**journalctl** 手册页 [man 1 journalctl](#) 中介绍了所有开关。



### 提示：与特定可执行文件相关的消息

要显示与特定可执行文件相关的所有日志消息，请指定该可执行文件的完整路径：

```
> sudo journalctl /usr/lib/systemd/systemd
```

**-f**

仅显示最近的日志消息，另外，在将新的日志项添加到日志时会列显这些新项。

列显消息并跳转到日志末尾，以便在页导航中显示最新的项。

**-r**

以倒序列显日志消息，让最新的项列在最前面。

**-k**

仅显示内核消息。这与字段匹配 `_TRANSPORT=kernel` 相同（请参见第 21.3.3 节“根据字段过滤”）。

**-u**

仅显示指定 `systemd` 单元的消息。这与字段匹配 `_SYSTEMD_UNIT=UNIT` 相同（请参见第 21.3.3 节“根据字段过滤”）。

```
> sudo journalctl -u apache2
[...]
Jun 03 10:07:11 pinkiepie systemd[1]: Starting The Apache Webserver...
Jun 03 10:07:12 pinkiepie systemd[1]: Started The Apache Webserver.
```

## 21.3 过滤日志输出

如果不结合任何开关调用 **journalctl**，它将显示日志的整个内容，最旧的项列在最前面。您可按特定的开关和字段过滤输出。

### 21.3.1 根据引导编号过滤

**journalctl** 可以根据特定的系统引导过滤消息。要列出所有可用引导，请运行

```
> sudo journalctl --list-boots
-1 097ed2cd99124a2391d2cffab1b566f0 Mon 2014-05-26 08:36:56 EDT-Fri 2014-05-30
05:33:44 EDT
0 156019a44a774a0bb0148a92df4af81b Fri 2014-05-30 05:34:09 EDT-Fri 2014-05-30
06:15:01 EDT
```

第一列列出引导偏移：`0` 表示当前引导，`-1` 表示上一次引导，`-2` 表示再上一次引导，以此类推。第二列包含引导 ID 后接特定引导的限制时戳。

显示当前引导中的所有消息：

```
> sudo journalctl -b
```

如果需要查看来自前一引导的日志消息，请加一个偏移参数。下面的示例将输出前一引导的消息：

```
> sudo journalctl -b -1
```

另一种方法是根据引导 ID 列出引导消息。要实现此目的，请使用 `_BOOT_ID` 字段：

```
> sudo journalctl _BOOT_ID=156019a44a774a0bb0148a92df4af81b
```

### 21.3.2 根据时间间隔过滤

您可通过指定开始日期和/或结束日期来过滤 **journalctl** 的输出。日期规范应采用类似于“2014-06-30 9:17:16”的格式。如果省略时间部分，则会假设为午夜。如果省略秒，则会假设为“:00”。如果省略日期部分，则会假设为当日。您也可以不采用数字表示，而是指定关键

字“yesterday”、“today”或“tomorrow”。它们表示当日前一天、当日或者当日后一天的午夜。如果指定“now”，则表示当前时间。您还可以指定以`_`或`+`为前缀的相对时间，分别表示当前时间之前或之后的特定时间。

仅显示从现在开始生成的新消息，并持续更新输出：

```
> sudo journalctl --since "now" -f
```

显示从昨天午夜到 3:20AM 的所有消息：

```
> sudo journalctl --since "today" --until "3:20"
```

### 21.3.3 根据字段过滤

您可以按特定的字段过滤日志输出。要匹配的字段语法为`FIELD_NAME=MATCHED_VALUE`，例如`_SYSTEMD_UNIT=httpd.service`。您可以在单个查询中指定多个匹配条件，以进一步过滤输出消息。有关默认字段的列表，请参见[man 7 systemd.journal-fields](#)。

显示特定进程 ID 生成的消息：

```
> sudo journalctl _PID=1039
```

显示属于特定用户 ID 的消息：

```
# journalctl _UID=1000
```

显示来自内核环缓冲区的消息（与[dmesg](#)的生成结果相同）：

```
> sudo journalctl _TRANSPORT=kernel
```

显示来自服务的标准输出或错误输出的消息：

```
> sudo journalctl _TRANSPORT=stdout
```

仅显示指定服务生成的消息：

```
> sudo journalctl _SYSTEMD_UNIT=avahi-daemon.service
```

如果指定了两个不同的字段，则仅显示同时与两个表达式匹配的项：

```
> sudo journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1488
```

如果两个匹配条件引用了相同的字段，则显示与两个表达式中任意一个匹配的所有项：

```
> sudo journalctl _SYSTEMD_UNIT=avahi-daemon.service _SYSTEMD_UNIT=dbus.service
```

您可以使用“+”分隔符将两个表达式组合成一个逻辑“OR”。以下示例将显示来自进程 ID 为 1480 的 Avahi 服务进程的所有消息，以及来自 D-Bus 服务的所有消息：

```
> sudo journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1480 +
 _SYSTEMD_UNIT=dbus.service
```

## 21.4 调查 systemd 错误

本节将介绍一个简单的示例，演示如何找出并修复在 apache2 启动期间 systemd 报告的错误。

### 1. 尝试启动 apache2 服务：

```
# systemctl start apache2
Job for apache2.service failed. See 'systemctl status apache2' and
'journalctl -xn' for details.
```

### 2. 我们来看看该服务的状态如何：

```
> sudo systemctl status apache2
apache2.service - The Apache Webserver
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
   Active: failed (Result: exit-code) since Tue 2014-06-03 11:08:13 CEST;
             7min ago
     Process: 11026 ExecStop=/usr/sbin/start_apache2 -D SYSTEMD -DFOREGROUND \
               -k graceful-stop (code=exited, status=1/FAILURE)
```

导致错误的进程 ID 为 11026。

### 3. 显示与进程 ID 11026 相关的详细消息：

```
> sudo journalctl -o verbose _PID=11026
[...]
MESSAGE=AH00526: Syntax error on line 6 of /etc/apache2/default-
server.conf:
[...]
```

```
MESSAGE=Invalid command 'DocumenttRoot', perhaps misspelled or defined by a
module
[...]
```

4. 更正 /etc/apache2/default-server.conf 中的拼写错误，启动 apache2 服务，然后列显其状态：

```
> sudo systemctl start apache2 && systemctl status apache2
apache2.service - The Apache Webserver
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
     Active: active (running) since Tue 2014-06-03 11:26:24 CEST; 4ms ago
   Process: 11026 ExecStop=/usr/sbin/start_apache2 -D SYSTEMD -DFOREGROUND
              -k graceful-stop (code=exited, status=1/FAILURE)
 Main PID: 11263 (httpd2-prefork)
    Status: "Processing requests..."
      CGrou: /system.slice/apache2.service
              ├─11263 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
              ├─11280 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
              ├─11281 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
              ├─11282 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
              ├─11283 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
              └─11285 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D
[...]
```

## 21.5 Journald 配置

通过修改 /etc/systemd/journald.conf 可调整 systemd-journald 服务的行为。本节只介绍了基本的选项设置。有关完整的文件描述，请参见 man 5 journald.conf。要使更改生效，需要使用以下命令重启动日志

```
> sudo systemctl restart systemd-journald
```

## 21.5.1 更改日志大小限制

如果将日志数据保存到永久位置（请参见第 21.1 节“将日志设为永久”），这些数据最多会占用 `/var/log/journal` 所在文件系统空间的 10%。例如，如果 `/var/log/journal` 位于 30 GB 的 `/var` 的分区中，则日志最多会占用 3 GB 磁盘空间。要更改此限制，请更改（并取消注释）`SystemMaxUse` 选项：

```
SystemMaxUse=50M
```

## 21.5.2 将日志转发到 `/dev/ttYX`

您可以将日志转发到终端设备，以便在首选的终端屏幕（例如 `/dev/ttY12`）上显示相关的系统消息。将以下 `journald` 选项更改为

```
ForwardToConsole=yes
TTYPath=/dev/ttY12
```

## 21.5.3 将日志转发到系统日志工具

`Journald` 与传统的系统日志实现工具（例如 `rsyslog`）向后兼容。请确保满足以下条件：

- 已安装 `rsyslog`。

```
> sudo rpm -q rsyslog
rsyslog-7.4.8-2.16.x86_64
```

- 已启用 `rsyslog` 服务。

```
> sudo systemctl is-enabled rsyslog
enabled
```

- 已在 `/etc/systemd/journald.conf` 中启用转发到 `syslog` 设置。

```
ForwardToSyslog=yes
```

## 21.6 使用 YaST 过滤 systemd 日志

过滤 systemd 日志的简单方法（无需处理 journalctl 语法）是使用 YaST 日志模块。使用 **sudo zypper in yast2-journal** 安装该模块后，请在 YaST 中选择系统 > Systemd 日志启动该模块。也可以在命令行中输入 **sudo yast2 journal** 来启动该模块。

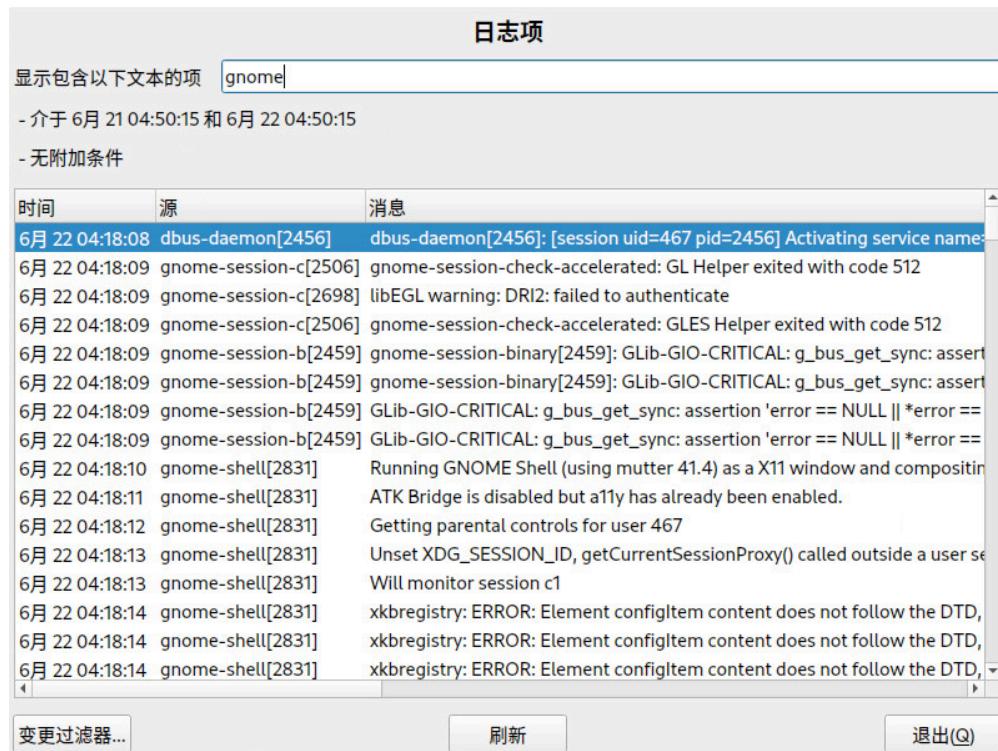


图 21.1：YAST SYSTEMD 日志

模块将在表中显示日志项。使用顶部的搜索框可以搜索包含特定字符的项，这类似于使用 **grep**。要按日期和时间、单位、文件或优先级过滤项，请单击更改过滤器，然后设置相应的选项。

## 21.7 在 GNOME 中查看日志

您可以使用 **GNOME 日志** 来查看日志。可从应用程序菜单启动它。要查看系统日志消息，需要以 root 身份运行它，例如使用 **xdg-su gnome-logs**。按 **Alt + F2** 时可执行此命令。

## 22 **update-alternatives**: 管理命令和文件的多个版本

系统上往往安装同一个工具的多个版本。为了让管理员可以选择，以及能一起安装和使用多个不同的版本，备选项系统提供了以一致的方式管理此类版本的功能。

### 22.1 概览

在 SUSE Linux Enterprise Desktop 上，有几个程序执行的是相同或类似的任务。例如，如果系统上既安装了 Java 1.7，也安装了 Java 1.8，则会从 RPM 软件包中调用备选项系统脚本 (**update-alternatives**)。备选项系统默认将参照版本 1.8：版本越高，优先级就越高。不过，管理员可以更改该默认设置，并可将通用名称指向版本 1.7。

本章使用了下列术语：

#### 术语

##### 管理目录

默认的 /var/lib/rpm/alternatives 目录，其中包含有关备选项当前状态的信息。

##### 替代方案

文件系统中某个特定文件的名称，可使用备选项系统通过通用名称访问。

##### 备选项目录

包含符号链接的默认 /etc/alternatives 目录。

##### 通用名称

该名称（例如 /usr/bin/edit）参照可通过备选项系统使用的多个文件中的一个。

##### 链接组

一组相关的符号链接，可作为一个组更新。

##### 主链接

链接组中用于确定如何配置组中其他链接的链接。

## 从属链接

链接组中受主链接控制的链接。

## 符号链接 (symlink)

在同一文件系统中作为另一个文件的参照的文件。备选项系统使用备选项目目录中的符号链接在一个文件的各个版本之间切换。

管理员可通过 **update-alternatives** 命令修改备选项目目录中的符号链接。

备选项系统提供 **update-alternatives** 命令供您创建、去除、维护和显示有关符号链接的信息。虽然这些符号链接通常是指向命令，但它们也可以指向 JAR 存档、手册页及其他文件。本章中的示例使用了命令和手册页，但它们也适用于其他文件类型。

备选项系统使用备选项目目录来收集指向可能的备选项的链接。安装包含备选项的新软件包时，就会将新的备选项添加至系统。是否将新软件包的备选项选为默认值取决于它的优先级和设置的模式。版本较高的软件包优先级也较高。备选项系统可以在两种模式下工作：

- **自动模式：** 在此模式下，备选项系统会确保组中的链接指向适合组且优先级最高的备选项。
- **手动模式：** 在此模式下，备选项系统不会对系统管理员的设置进行任何更改。

例如，在备选项系统中，**java** 命令具有以下链接层次结构：

### 例 22.1：java 命令的备选项系统

```
/usr/bin/java ①
 -> /etc/alternatives/java ②
      -> /usr/lib64/jvm/jre-10-openjdk/bin/java ③
```

- ① 通用名称。
- ② 备选项目目录中的符号链接。
- ③ 备选项之一。

## 22.2 使用案例

默认从 RPM 软件包中调用 **update-alternatives** 脚本。安装或去除一个软件包时，该脚本会处理该软件包所有的符号链接。不过，您可以从命令行中手动运行该脚本，以便：

- 显示通用名称的当前备选项。
- 更改备选项的默认值。
- 为备选项创建一组相关的文件。

## 22.3 获取备选项的概览

要检索所有已配置备选项的名称，请使用：

```
> ls /var/lib/alternatives
```

要获取所有已配置备选项的概览及其值，请使用

```
> sudo update-alternatives --get-selections
asadmin                  auto    /usr/bin/asadmin-2.7
awk                      auto    /usr/bin/gawk
chardetect                auto    /usr/bin/chardetect-3.6
dbus-launch                auto    /usr/bin/dbus-launch.x11
default-displaymanager     auto    /usr/lib/X11/displaymanagers/gdm
[...]
```

## 22.4 查看有关特定备选项的细节

检查备选项最简单的方法就是查看命令的符号链接。例如，要确定 **java** 命令参照的是什么，请使用以下命令：

```
> readlink --canonicalize /usr/bin/java
/usr/lib64/jvm/jre-10-openjdk/bin/java
```

如果您看到相同的路径（在本例中为 **/usr/bin/java**），则说明此命令没有备选项。

要查看全部的备选项（包括从属链接），请使用 **--display** 选项：

```
> sudo update-alternatives --display java
java - auto mode
  link best version is /usr/lib64/jvm/jre-1.8.0-openjdk/bin/java
  link currently points to /usr/lib64/jvm/jre-1.8.0-openjdk/bin/java
```

```
link java is /usr/bin/java
slave java.1.gz is /usr/share/man/man1/java.1.gz
slave jre is /usr/lib64/jvm/jre
slave jre_exports is /usr/lib64/jvm-exports/jre
slave keytool is /usr/bin/keytool
slave keytool.1.gz is /usr/share/man/man1/keytool.1.gz
slave orbd is /usr/bin/orbd
slave orbd.1.gz is /usr/share/man/man1/orbd.1.gz
[...]
```

## 22.5 设置备选项的默认版本

/usr/bin 中的命令默认会参照优先级最高的备选项目目录。例如，java 命令默认会显示下面的版本号：

```
> java -version
openjdk version "10.0.1" 2018-04-17
OpenJDK Runtime Environment (build 10.0.1+10-suse-lp150.1.11-x8664)
OpenJDK 64-Bit Server VM (build 10.0.1+10-suse-lp150.1.11-x8664, mixed mode)
```

要将默认 java 命令更改为参照先前的版本，请运行：

```
> sudo update-alternatives --config java
root's password:
There are 2 choices for the alternative java (providing /usr/bin/java).

      Selection    Path                      Priority      Status
      -----
* 0            /usr/lib64/jvm/jre-10-openjdk/bin/java    2005      auto mode
  1            /usr/lib64/jvm/jre-1.8.0-openjdk/bin/java  1805      manual mode
  2            /usr/lib64/jvm/jre-10-openjdk/bin/java    2005      manual mode
  3            /usr/lib64/jvm/jre-11-openjdk/bin/java      0        manual mode

Press <enter> to keep the current choice[*], or type selection number:
```

根据您的系统和安装的版本，具体的 Java 版本号会有所不同。选择 1 之后，java 会显示下面的版本号：

```
> java -version
java version "1.8.0_171"
OpenJDK Runtime Environment (IcedTea 3.8.0) (build 1.8.0_171-b11 suse-
lp150.2.3.1-x86_64)
OpenJDK 64-Bit Server VM (build 25.171-b11, mixed mode)
```

另外，请记住以下几点：

- 在手动模式下安装另一个 Java 版本时，备选项系统既不会更改链接，也不会更改通用名称。
- 在自动模式下安装另一个 Java 版本时，备选项系统会更改 Java 主链接及所有从属链接（如第 22.4 节 “查看有关特定备选项的细节” 中所示）。要检查主/从关系，请使用：

```
> sudo update-alternatives --display java
```

## 22.6 安装自定义备选项

本节介绍如何在系统上设置自定义备选项。示例做了以下假设：

- 具有类似功能的脚本有两个，分别是 foo-2 和 foo-3。
- 这些脚本存储在 /usr/local/bin 目录中，以免与 /usr/bin 中的系统工具产生任何冲突。
- 有一个主链接 foo 指向 foo-2 或 foo-3。

要在您的系统上提供备选项，请执行以下步骤：

1. 将您的脚本复制到 /usr/local/bin 目录中。
2. 让脚本可执行：

```
> sudo chmod +x /usr/local/bin/foo-{2,3}
```

3. 针对两个脚本运行 update-alternatives：

```
> sudo update-alternatives --install \
```

```
/usr/local/bin/foo ① \
foo ② \
/usr/local/bin/foo-2 ③ \
200 ④
> sudo update-alternatives --install \
  /usr/local/bin/foo ① \
  foo ② \
  /usr/local/bin/foo-3 ③ \
  300 ④
```

--install 后面的选项具有以下含义：

- ① 通用名称。为了避免混淆，该名称通常是不含任何版本号的脚本名称。
- ② 主链接的名称。必须相同。
- ③ 位于 `/usr/local/bin` 中的原始脚本的路径。
- ④ 优先级。我们给 `foo-2` 设置一个低于 `foo-3` 的优先级。最好使用增大得非常明显的数字来区分优先级。例如，`foo-2` 的优先级是 200，`foo-3` 的优先级是 300。

#### 4. 检查主链接：

```
> sudo update-alternatives --display foo
foo - auto mode
  link best version is /usr/local/bin/foo-3
  link currently points to /usr/local/bin/foo-3
  link foo is /usr/local/bin/foo
  /usr/local/bin/foo-2 - priority 200
  /usr/local/bin/foo-3 - priority 300
```

完成所述步骤后，便可以使用主链接 `/usr/local/bin/foo` 了。

如果需要，可以安装其他的备选项。要去除备选项，请使用以下命令：

```
> sudo update-alternatives --remove foo /usr/local/bin/foo-2
```

去除此脚本之后，`foo` 组的备选项系统看上去如下所示：

```
> sudo update-alternatives --display foo
foo - auto mode
  link best version is /usr/local/bin/foo-3
```

```
link currently points to /usr/local/bin/foo-3
link foo is /usr/local/bin/foo
/usr/local/bin/foo-3 - priority 300
```

## 22.7 定义依赖备选项

如果您有备选项，则仅凭脚本本身不足以完成任务。大部分命令都不是独立的 — 它们随附其他文件，例如扩展、配置或手册页。要创建依赖于主链接的备选项，请使用从属备选项。

我们假设要扩展[第 22.6 节 “安装自定义备选项”](#) 中的示例，并提供手册页和配置文件：

- foo-2.1.gz 和 foo-3.1.gz 这两个手册页，存储在 /usr/local/man/man1 目录中。
- foo-2.conf 和 foo-3.conf 这两个配置文件，存储在 /etc 中。

请执行以下步骤将其他文件添加至备选项中：

1. 将配置文件复制到 /etc 中：

```
> sudo cp foo-{2,3}.conf /etc
```

2. 将手册页复制到 /usr/local/man/man1 目录中：

```
> sudo cp foo-{2,3}.1.gz /usr/local/man/man1/
```

3. 使用 --slave 选项将从属链接添加至主脚本：

```
> sudo update-alternatives --install \
  /usr/local/bin/foo foo /usr/local/bin/foo-2 200 \
  --slave /usr/local/man/man1/foo.1.gz \
  foo.1.gz \
  /usr/local/man/man1/foo-2.1.gz \
  --slave /etc/foo.conf \
  foo.conf \
  /etc/foo-2.conf
> sudo update-alternatives --install \
  /usr/local/bin/foo foo /usr/local/bin/foo-3 300 \
```

```
--slave /usr/local/man/man1/foo.1.gz \
foo.1.gz \
/usr/local/man/man1/foo-3.1.gz \
--slave /etc/foo.conf \
foo.conf \
/etc/foo-3.conf
```

#### 4. 检查主链接：

```
foo - auto mode
link best version is /usr/local/bin/foo-3
link currently points to /usr/local/bin/foo-3
link foo is /usr/local/bin/foo
slave foo.1.gz is /usr/local/man/man1/foo.1.gz
slave foo.conf is /etc/foo.conf
/usr/local/bin/foo-2 - priority 200
  slave foo.1.gz: /usr/local/man/man1/foo-2.1.gz
  slave foo.conf: /etc/foo-2.conf
/usr/local/bin/foo-3 - priority 300
  slave foo.1.gz: /usr/local/man/man1/foo-3.1.gz
  slave foo.conf: /etc/foo-3.conf
```

如果您使用 **update-alternatives --config foo** 将链接更改为 **foo-2**，则所有从属链接也会随之更改。

## 23 基本网络知识

Linux 提供集成进各类网络结构中所需的网络工具和功能。可以通过 YaST 配置使用网络卡进行的网络访问。也可以手动进行配置。本章仅介绍基础机制和相关的网络配置文件。

Linux 和其他 Unix 操作系统均使用 TCP/IP 协议。这不是单个网络协议，而是提供多种服务的一系列网络协议。[TCP/IP 协议系列中的数种协议](#) 中所列的协议用于通过 TCP/IP 在两台计算机之间交换数据。由 TCP/IP 连接而成的网络构成了全球网络，也称作“互联网”。

**RFC 指注释请求** (Request for Comments)。RFC 由一些文档组成，用来描述互联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互联网协议。有关 RFC 的更多信息，请参见 <https://datatracker.ietf.org/>。

### [TCP/IP 协议系列中的数种协议](#)

#### **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 确定在传输过程中是否有任何数据丢失或发生混乱。只要涉及到数据序列就会实施 TCP。

#### **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**

互联网控制消息协议：这不是面向最终用户的协议，而是用来发出错误报告的特殊控制协议，能够控制参与 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**

互联网组管理协议：此协议用于在实施 IP 多广播时控制计算机行为。

如图 23.1 “TCP/IP 的简化层模型” 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 (互联网协议) 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 层由底层硬件相关协议 (例如以太网) 提供支持。

# TCP/IP 模型

# OSI 模型



图 23.1：TCP/IP 的简化层模型

该图为每一层都提供了一到两个示例。层按照**抽象程度**排序。最底层离硬件很近。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。要传送的数据收集在包中（一次无法发送所有数据）。TCP/IP 包最大约为 64 KB。由于网络硬件有所限制，因此包通常都很小。以太网上数据包的大小上限约为 1500 个字节。通过以太网发送数据时，TCP/IP 包的大小不能超过此限制。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。图 23.2 “TCP/IP 以太网数据包” 中演示了一个通过以太网电缆传送的 TCP/IP 数据包示例。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 23.2：TCP/IP 以太网数据包

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。对于应用程序而言，数据是通过无线连接还是有线连接传输都无关紧要。同样，只要数据包的格式正确，传送哪种类型的数据对数据行而言也无关紧要。

## 23.1 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 23.2 节“IPv6 - 新一代互联网”](#)。

### 23.1.1 IP 地址

互联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 23.1 “编写 IP 地址”](#)的第二行所示的格式书写。

#### 例 23.1：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。它在全球只能使用一次。这条规则也有例外，但这些例外与下文无关。

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。但是，此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）。

## 23.1.2 网络掩码和路由

网络掩码用于定义子网的地址范围。如果两台主机位于同一子网中，它们可直接相互访问。如果它们位于不同子网中，则需要用于处理此子网的所有通讯的网关地址才能相互访问。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 23.2 “将 IP 地址链接到网络掩码”](#)。网络掩码有 32 位，它确定 IP 地址有多少属于网络。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。对于所有值为 0 的位，标记其属于子网内。这意味着值为 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，所以也可通过计算网络掩码中的位数来确定。在[例 23.2 “将 IP 地址链接到网络掩码”](#)中，第一个 24 位的网络也可写作 192.168.0.0/24。

### 例 23.2：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连接的所有计算机通常都位于同一子网中，可以直接访问。

即使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）。

## 特定地址

### 基础网络地址

这是网络掩码和该网络中的任意地址，如例 23.2 “将 IP 地址链接到网络掩码” 中的 Result (结果) 所示。不能将此地址指派给任何主机。

### 广播地址

这可以解释为：“访问此子网中的所有主机”。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”。可以使用此地址以及通过 IPv4 定义的完整 127.0.0.0/8 回写网络中的所有地址为您自己的计算机设置一个连接。如果使用 IPv6，则只有一个回写地址 (::1)。

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。这些地址无法与互联网上的其他地址建立任何连接，因为它们不能通过互联网传送。这些地址域在 RFC 1597 中指定，并且列在表 23.1 “专用 IP 地址域” 中。

表 23.1：专用 IP 地址域

| 网络/网络掩码                        | 域                              |
|--------------------------------|--------------------------------|
| <u>10.0.0.0/255.0.0.0</u>      | <u>10.x.x.x</u>                |
| <u>172.16.0.0/255.240.0.0</u>  | <u>172.16.x.x - 172.31.x.x</u> |
| <u>192.168.0.0/255.255.0.0</u> | <u>192.168.x.x</u>             |

## 23.2 IPv6 - 新一代互联网

由于万维网 (WWW) 的出现，过去十五年内，越来越多的计算机通过 TCP/IP 进行通讯，互联网的规模发生了爆炸性的增长。自从 1990 年在 CERN (<http://public.web.cern.ch>) 任职的 Tim Berners-Lee 开创了 WWW，互联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。而且还损失了少许 IP 地址，它们因网络组织方式的原因而无法使用。子网中可用的地址数量是位数的平方减 2。举例来说，一个子网可能有 2 个、6 个或 14 个可用地址。如果要将 128 台主机连接到互联网，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；要在 IPv4 网络中设置主机，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。

### 23.2.1 优点

IPv6 协议带来的最为重要同时也最为显著的改进是极大地扩展了可用的地址空间。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 23.2.2 节“地址类型和结构”。

下面列出了 IPv6 协议的其他优点：

#### 自动配置

IPv6 使网络能够“即插即用”，这意味着无需进行任何手动配置即可将新配置的系统集成到（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（IPv4 中需要使用 DHCP 服务器来自动分配地址）。

不过，如果路由器已连接到交换机，则路由器应发送带标志的定期通告，告知网络中的主机彼此应如何交互。有关详细信息，请参见 RFC 2462 和 `radvd.conf(5)` 手册页以及 RFC 3315。

## 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。这样用户能够轻松访问多个网络，如同使用移动电话公司提供的国际漫游服务一样轻松自如。当您带着手机出国时，手机只要进入通信区域就会自动登录国外的服务，这样无论您身处何地，用同一个号码都可以联系到您，而且您也能像在国内一样拨打电话。

## 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，避免被互联网上的外来者窃听。

## 向后兼容性

实际上，要想将整个互联网一下子从 IPv4 转换为 IPv6 是不可能的。因此，这两个协议不仅要能在互联网上共存，还应能够共存于一个系统中，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），另一方面还要使用多个隧道。请参见[第 23.2.3 节 “IPv4 与 IPv6 并存”](#)。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。

## 通过多播自定义定制服务

使用 IPv4 时，某些服务（如 SMB）需要向本地网络中的所有主机广播其数据包。IPv6 使服务器能够通过**多播**对主机寻址（即将多个主机作为组的一部分寻址），因而提供了更精细的方法。这种方法与通过**广播**对所有主机寻址，或通过**单播**单独对每个主机寻址均不同。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。可以使用预定义的特定组来寻址，例如对所有名称服务器寻址（**所有名称服务器多播组**），或对所有路由器寻址（**所有路由器多播组**）。

## 23.2.2 地址类型和结构

如上所述，当前的 IP 协议存在两个重要限制：IP 地址日益短缺，并且配置网络、维护路由选择表的任务变得越来越复杂繁重。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），第二个问题也得到缓解。

使用 IPv6 时，了解三种类型的地址十分有用：

## 单播

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，使用单播地址可以将包传送到本地网络或互联网上的单个主机。

## 多播

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。如果此服务器无法回复，协议会自动选择第二近的服务器，然后是第三近的服务器，以此类推。

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。它们之间用冒号 (:) 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 23.3 “示例 IPv6 地址”

### 例 23.3：示例 IPv6 地址

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :     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 23.4 “指定前缀长度的 IPv6 地址” 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。与 IPv4 一样，IP 地址使用 AND 结合网络掩码值，以确定主机位于同一子网中还是其他网络中。

### 例 23.4：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。IPv6 前缀中列出了其中的一部分。

### 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。

### 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的网络之内。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

这些是多播地址。

单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）用于通过公共互联网路由数据包。其中包含提供互联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，配置过程有所简化。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，如基于点对点协议 (PPP) 的接口。

在这个基础架构之上，IPv6 还区分五种不同的单播地址：

## :: (未指定)

在首次初始化接口时（此时尚无法通过其他方法确定地址），这类地址可被主机用作其源地址。

## ::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 23.2.3 节“[IPv4 与 IPv6 并存](#)”），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。不能具有此类源地址或目标地址的包路由到互联网或其他子网。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互联网 — 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。其中包含特殊前缀 (fec0::/10)、接口 ID，及指定子网 ID 的 16 位字段。其余部分也会填入零字节。

作为 IPv6 引入的新功能，每个网络接口通常可以获得多个 IP 地址，其优点在于可以通过同一接口访问多个网络。其中一个网络可以用 MAC 和已知前缀自动配置，这样当启用 IPv6（使用链接-本地地址）后，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以传递到该主机，不管它是在本地网络还是其他

任何网络中操作。这可通过 IPv6 引入的新功能来实现，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。预定要发送到转交地址的那些包会直接传送到移动主机，而不会特别绕行。

### 23.2.3 IPv4 与 IPv6 并存

将与互联网相连的所有主机从 IPv4 迁移到 IPv6 是一个逐步的过程。这两种协议将并存一段时间。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。但这仍然没有解决启用了 IPv6 的主机如何与 IPv4 主机通讯，以及应如何通过当前的网络（绝大部分都基于 IPv4）传输 IPv6 数据包的问题。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 23.2.2 节“[地址类型和结构](#)”）。

IPv6 主机孤立于（全球）IPv4 网络，它可通过隧道通讯：IPv6 包会封装成 IPv4 包在 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，并通过支持多播的 IPv4 网络发送。这种方法诱导 IPv6 将整个网络（互联网）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，这种方法的可扩展性不好，而且不易推广，因为 IP 多播在互联网上尚未普及。因此，它提供的解决方案仅适用于支持多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 主机和互联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。

## 23.2.4 配置 IPv6

要配置 IPv6，通常无需在各个工作站上执行任何更改。默认情况下启用 IPv6。要在已安装系统上禁用或启用 IPv6，请使用 YaST 网络设置模块。在全局选项选项卡上，根据需要选择或取消选择启用 IPv6 选项。要暂时启用直至下次重引导，请以 `root` 身份输入 `modprobe -i ipv6`。装载 IPv6 模块后无法将其卸载。

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` 文件设置多种隧道的信息，请参见 `ifcfg-tunnel` 的手册页 (`man ifcfg-tunnel`)。

## 23.2.5 更多信息

上文的概述中并未全面论述 IPv6 这一主题。如果要深入了解这种较新的协议，请参考以下联机文档和书籍：

<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 2460

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。

## 23.3 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称为 bind 的特殊软件来完成。负责这种转换的计算机称为**名称服务器**。这些名称构成了具有层次结构的系统，各个名称组成部分之间用句点分隔。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 指派由于历史原因已经变得混乱。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 年还引入了较长的 TLD 来表示特定的活动范围（例如 .info、.name、.museum）。

在互联网发展的早期阶段（1990 年之前），文件 /etc/hosts 用来存储互联网上表示的所有计算机名称。后来事实证明随着接入互联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，以十分分散的方式存储主机名。这种数据库与名称服务器类似，没有有关互联网上所有主机的立即可用数据，但可以向其他名称服务器发送请求。

位于层次顶级的是 **root 名称服务器**。这些 root 名称服务器管理顶级域，并由网络信息中心 (NIC) 运行。每个 root 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电子邮件的主机，即**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。使用 YaST 可轻松指定此类名称服务器。

whois 协议与 DNS 密切相关。使用此程序可以快速找出负责给定域的服务器。



## 注意：MDNS 和 .local 域名

解析程序会将 `.local` 顶级域视为链接-本地域。DNS 请求作为多播 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。有关更多信息，请参见 `host.conf` 手册页。

要在安装期间关闭 MDNS，请使用 `nomdns=1` 引导参数。

有关多播 DNS 的详细信息，请参见 <http://www.multicastdns.org>。

## 23.4 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 23.6 节“手动配置网络连接”。

在 NetworkManager 默认处于活动状态的 SUSE Linux Enterprise Desktop 上，所有网卡都会加以配置。如果 NetworkManager 不处于活动状态，那么系统只会自动配置使用链路连通的第一个接口（已连接网络电缆）。可随时在已安装系统中配置额外的硬件。以下章节将介绍 SUSE Linux Enterprise Desktop 支持的所有网络连接类型的网络配置。

### 23.4.1 使用 YaST 配置网卡

要在 YaST 中配置以太网卡或 Wi-Fi/蓝牙卡，请选择系统 > 网络设置。启动模块后，YaST 将显示网络设置对话框，其中包括四个选项卡：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，例如网路设置方法、IPv6 和常规 DHCP 选项。有关详细信息，请参见第 23.4.1.1 节“配置全局联网选项”。

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。要手动配置未自动检测到的网卡，请参见第 23.4.1.3 节“配置未检测到的网卡”。要更改所配置网卡的配置，请参见第 23.4.1.2 节“更改网卡的配置”。

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，请参见第 23.4.1.4 节“配置主机名和 DNS”。

路由选择选项卡用于配置路由选择。有关更多信息，请参见第 23.4.1.5 节“配置路由”。



图 23.3：配置网络设置

### 23.4.1.1 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，如使用 NetworkManager、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，请选择 NetworkManager 服务。NetworkManager 非常适用于在多个有线和无线网络之间切换。如果您运行的不是桌面环境，或者您的计算机是 Xen 服务器、虚拟系统或者会在网络中提供 DHCP 或 DNS 等网络服务，请使用 Wicked 服务方法。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 的详细信息，请参见第 31 章 “使用 NetworkManager”。

在 IPv6 协议设置中，选择是否使用 IPv6 协议。可将 IPv6 与 IPv4 一起使用。默认情况下，会启用 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。要禁用 IPv6，请停用启用 IPv6。如果禁用了 IPv6，内核将不再自动装载 IPv6 模块。重启后会应用此设置。

在 DHCP 客户端选项中，配置 DHCP 客户端的选项。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当 DHCP 客户端将消息发送到 DHCP 服务器时，主机名选项字段使用的字符串。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（正向和反向记录）。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。如果保留 AUTO，将发送当前的主机名（即 `/etc/HOSTNAME` 中定义的主机名）。将选项字段保留空白则不会发送任何主机名。

如果您不希望根据 DHCP 中的信息更改默认路由，请停用通过 DHCP 更改默认路由。

### 23.4.1.2 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个网卡，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。

#### 23.4.1.2.1 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 版本 4（用于 DHCPv4）、DHCP 版本 6（用于 DHCPv6）还是 DHCP 版本 4 和 6。

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 NetworkManager 默认处于活动状态的 SUSE Linux Enterprise Desktop 上，所有网卡都会加以配置。

如果使用的是 DSL 线路，但 ISP（互联网服务提供商）没有指派静态 IP，此时还应使用 DHCP。如果决定使用 DHCP，请打开 YaST 网卡配置模块的网络设置对话框，在全局选项选项卡的 DHCP 客户端选项中配置细节。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 比较适合客户端配置，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中，于已检测到的网卡列表中选择一个网卡，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，请以 `/64` 格式使用子网掩码作为前缀长度。  
或者，您可以为此地址输入完全限定的主机名，该主机名将写入 `/etc/hosts` 配置文件。
4. 单击下一步。
5. 要激活配置，请单击确定。



## 注意：接口激活和链路检测

在激活网络接口期间，`wicked` 会检查载波，并且只有在检测到链路之后，才会应用 IP 配置。如果不管链路状态为何，您都需要应用配置（例如，您要测试侦听某个地址的服务），则可以在 `/etc/sysconfig/network/ifcfg` 内的接口配置文件中添加变量 `LINK_REQUIRED=no` 来跳过链路检测。

另外，您可以使用变量 `LINK_READY_WAIT=5` 来指定等待链路的超时值（以秒为单位）。

有关 `ifcfg-*` 配置文件的详细信息，请参见[第 23.6.2.5 节 “/etc/sysconfig/network/ifcfg-”](#) 和 [man 5 ifcfg](#)。

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[第 23.4.1.4 节 “配置主机名和 DNS”](#) 中所述进行。要配置网关，请按照[第 23.4.1.5 节 “配置路由”](#) 中所述进行。

### 23.4.1.2.2 配置多个地址

一个网络设备可以有多个 IP 地址（称为别名或标签）。



## 注意：别名是兼容功能

别名或标签仅适用于 IPv4。利用 `iproute2` 网络接口，您可以使用一个或多个地址。

要使用 YaST 为网卡设置更多地址，请执行以下操作：

1. 在 YaST 网络设置对话框的概述选项卡中，于已检测到的网卡列表中选择一个网卡，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 IPv4 地址标签、IP 地址和网络掩码。请注意，添加 IP 别名时必须指定 `/32` 网络掩码。不要在别名中包含接口名称。
4. 要激活该配置，请确认设置。

### 23.4.1.2.3 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。后一个选项更适合大型服务器，因为便于热插拔网卡。要使用 YaST 设置这些选项，请执行以下步骤：

1. 在 YaST 网络设置对话框的概述选项卡中，于已检测到的网卡列表中选择一个网卡，然后单击编辑。
2. 转到常规选项卡。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. 要激活该配置，请确认设置。

### 23.4.1.2.4 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 允许您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 设置这些选项，请执行以下步骤：

1. 在 YaST 网络设置模块的概述选项卡中，于已检测到的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。在选项中以 `= VALUE` 格式为所选驱动程序输入任何选项。如果使用多个选项，应用空格分隔这些选项。
4. 要激活该配置，请确认设置。

### 23.4.1.2.5 激活网络设备

如果使用结合 `wicked` 的方法，便可以将设备配置为在引导期间、连接电缆时或检测到网卡时启动、以手动方式启动或永不启动设备。要更改设备启动，请如下继续操作：

1. 在 YaST 的系统 > 网络设置中，于已检测到的网卡列表中选择一个网卡，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，可在接口可用时对其进行设置。这与在引导时选项很相似，唯一区别是如果引导时接口不存在，将不会发生错误。选择手动可通过 `ifup` 手动控制接口。选择从不将不启动设备。在 NFSroot 时与在引导时相似，区别是使用 `systemctl stop network` 命令不会关闭接口；如果 `wicked` 处于活动状态，则 `network` 服务还会处理 `wicked` 服务。如果您使用 NFS 或 iSCSI 根文件系统，则选择此选项。
3. 要激活该配置，请确认设置。



#### 提示：NFS 用作根文件系统

在通过网络以 NFS 共享形式挂载根分区的（无磁盘）系统中，配置可供访问 NFS 共享的网络设备时需保持谨慎。

关闭或重引导系统时，默认的处理顺序是关闭网络连接，然后卸载根分区。对于 NFS 根分区，这种顺序会产生问题，因为在尚未激活与 NFS 共享的网络连接的情况下，根分区无法完全卸载。为防止系统停用相关的网络设备，请按第 23.4.1.2.5 节 “激活网络设备” 中所述打开网络设备配置选项卡，然后在设备激活窗格中选择通过 NFSroot。

### 23.4.1.2.6 设置最大传输单元大小

您可为接口设置最大传输单元 (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 的系统 > 网络设置中，于已检测到的网卡列表中选择一个网卡，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. 要激活该配置，请确认设置。

### 23.4.1.2.7 PCIe 多功能设备

系统支持对以下技术提供支持的多功能设备：LAN、iSCSI 和 FCoE。FCoE 客户端 (yast2\_fcoe-client) 会在额外的列中显示私用标志，以允许用户选择用于 FCoE 的设备。网络模块 (yast2\_lan) 会针对网络配置排除“仅存储设备”。

### 23.4.1.2.8 IP-over-InfiniBand (IPoIB) 的 Infiniband 配置

1. 在 YaST 的系统 > 网络设置中选择 InfiniBand 设备，然后单击编辑。
2. 在常规选项卡中，选择一种 IP-over-InfiniBand (IPoIB) 模式：已连接（默认）或数据报。
3. 要激活该配置，请确认设置。

有关 InfiniBand 的详细信息，请参见 [/usr/src/linux/Documentation/infiniband/ipoib.txt](#)。

### 23.4.1.2.9 配置防火墙

您不必按《安全和强化指南》，第 23 章“掩蔽和防火墙”，第 23.4 节“firewalld”中所述执行详细的防火墙设置，只需在设置设备的过程中决定设备的基本防火墙配置。按如下所示继续：

1. 打开 YaST 的系统 > 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. 确定要将接口指派到的防火墙区域。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部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（不受保护）

防火墙正在运行，但不会强制执行任何规则来保护此接口。当计算机属于受外部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。

#### 非管制区域

隔离区域是位于内部网络和（恶意）互联网之前的附加防线。可从内部网络和互联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，并且全面保护其抵御其他假定有害的网络流量。这是默认选项。

4. 要激活该配置，请确认设置。

### 23.4.1.3 配置未检测到的网卡

如果未正确检测到某个网卡，该卡将不会包含在已检测到的网卡列表中。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 的系统 > 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 USB 设备，请激活相应的复选框，并单击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。

在 Ethtool 选项中，您可以为接口设置 `ifup` 使用的 `ethtool` 选项。有关可用选项的信息，请参见 `ethtool` 手册页。

如果选项字符串以 `-` 开头（例如 `-K INTERFACE_NAME rx on`），则会用当前接口名称替换字符串中的第二个词。否则（例如 `autoneg off speed 10`）`ifup` 会在开头添加 `-s INTERFACE_NAME`。

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，请参见第 23.4.1.2 节“更改网卡的配置”。
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。
6. 要激活新的网络配置，请确认设置。

#### 23.4.1.4 配置主机名和 DNS

如果您在安装期间未更改网络配置，并且已有以太网卡可用，则系统会自动为您的计算机生成主机名并激活 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 的系统 > 模块中的网络设置主机名/DNS 选项卡。
2. 输入主机名。请注意，主机名是全局性的，将应用到所有网络接口。

如果使用 DHCP 获取 IP 地址，则计算机的主机名将由 DHCP 服务器自动设置。如果连接到不同网络，您应禁用此行为，因为其他网络可能会指派不同的主机名，而在运行时更改主机名可能会导致混淆图形桌面。要禁止使用 DHCP 获取 IP 地址，请停用通过 DHCP 更改主机名。

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 `/run/netconfig/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）。此默认策略可满足通常情况。

如果选择了仅手动选项，系统将不允许 **netconfig** 修改 `/run/netconfig/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。除完整接口名外，也可使用基本通配符来匹配多个接口。例如，`eth* ppp?` 首先以所有 eth 为目标，然后是 ppp0 到 ppp9 的所有接口。可以使用两个特殊策略值来指示如何应用 `/etc/sysconfig/network/config` 文件中定义的静态设置：

#### **STATIC**

静态设置需要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。

有关详细信息，请参见 **netconfig(8)** 的手册页 (**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（未指定域）的域名。如果使用多个域搜索，则使用逗号或空格分隔域。
5. 要激活该配置，请确认设置。

也可以使用 YaST 从命令行编辑主机名。YaST 进行的更改会立即生效（手动编辑 `/etc/HOSTNAME` 文件时则不是这样）。要更改主机名，请使用以下命令：

```
# yast dns edit hostname=HOSTNAME
```

要更改名称服务器，请使用以下命令：

```
# yast dns edit nameserver1=192.168.1.116
# yast dns edit nameserver2=192.168.1.117
# yast dns edit nameserver3=192.168.1.118
```

### 23.4.1.5 配置路由

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（如果需要是 IPv4 和 IPv6）的 IP 地址。默认网关与每个可能的目标匹配，但是如果存在与所需地址匹配的路由表项，则会使用此项，而不是通过默认网关使用默认路由。
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）。要忽略这些值中的任意一个，请使用减号 -。要在表中输入默认网关，请在目标字段中使用 `default`。



#### 注意：路由优先级

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，请在选项中输入 `- metric NUMBER`。可能的最低度量为 0。度量最低的路由具有最高的优先级，将用作默认路由。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。

4. 如果系统是路由器，请根据需要在网络设置中启用 IPv4 转发和 IPv6 转发。
5. 要激活该配置，请确认设置。

## 23.5 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，您便无需担心配置网络接口并且无需移动时在网络之间切换。



## ! 重要：支持覆盖范围

只有安装了 SLED 扩展或 Workstation Extension 的 SUSE 桌面工作负载支持 NetworkManager。所有服务器认证都通过网络配置工具 **wicked** 来完成，使用 NetworkManager 可能会使它们失效。SUSE 服务器工作负载不支持 NetworkManager。

### 23.5.1 NetworkManager 和 **wicked**

但是 NetworkManager 解决方案并非适用于所有情况，因此您仍可在 **wicked** 控制方法（用于管理网络连接）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 31.2 节 “启用或禁用 NetworkManager”](#) 中所述），然后使用 NetworkManager 配置网络连接。有关用例的列表以及如何配置和使用 NetworkManager 的详细说明，请参见[第 31 章 “使用 NetworkManager”](#)。

wicked 和 NetworkManager 的一些区别：

#### root 特权

如果使用 NetworkManager 进行网络设置，则可以随时使用一个小程序在您的桌面环境内轻松地切换、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。

**wicked** 也提供一些像用户管理的设备那样切换、停止或启动连接的途径（有的需要用户干预，有的则不需要）。但是，此方法始终需要 root 特权才能更改或配置网络设备。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

**wicked** 和 NetworkManager 都可以处理与无线网络（WEP、WPA-PSK 和 WPA-Enterprise 访问）和使用 DHCP 及静态配置的有线网络之间的网络连接。它们还支持拨号连接和 VPN 连接。使用 NetworkManager 还可以连接移动宽带 (3G) 调制解调器或设置 DSL 连接，而传统配置则不能进行这种连接。

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。如果想使用 **wicked** 获得相同的功能，需要执行更多配置工作。

## 23.5.2 NetworkManager 功能和配置文件

用 NetworkManager 创建的单个网络连接设置存储在配置文件中。使用 NetworkManager 或 YaST 配置的系统连接保存在 `/etc/NetworkManager/system-connections/*` 或 `/etc/sysconfig/network/ifcfg-*` 中。对于 GNOME，所有用户定义的连接都存储在 GConf 中。

如果未配置任何配置文件，NetworkManager 将自动创建一个配置文件，并将其命名为 `Auto$INTERFACE-NAME`。这样做是试图无需为任意多的（安全）案例进行任何配置就能使用。如果自动创建的配置文件不适合您的需要，请用 GNOME 提供的网络连接配置对话框按需要修改它们。有关详细信息，请参见第 31.3 节 “配置网络连接”。

## 23.5.3 控制和锁定 NetworkManager 功能

在集中管理的计算机上，可以使用 Polkit 控制或禁用某些 NetworkManager 功能，例如，是否允许用户修改管理员定义的连接，或者是否允许用户定义自己的网络配置。要查看或更改相关的 NetworkManager 策略，请启动 Polkit 的图形授权工具。在左侧的树中 `network-manager-settings` 条目下找到它们。有关 Polkit 的介绍及其使用方法细节，请参见《安全和强化指南》，第 18 章 “Polkit 身份验证框架”。

## 23.6 手动配置网络连接

应该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。

### 23.6.1 **wicked** 网络配置

名为 **wicked** 的工具和库提供了一个用于配置网络的新框架。

传统网络接口管理面临的其中一项挑战是，各种不同的网络管理层混杂在一个脚本中，最多在两个不同的脚本中。这些脚本彼此之间如何交互没有明确的定义。这会导致出现无法预测的问题、不明确的约束和约定等情况。针对各种不同的情况部署多个特殊入侵层增加了维护负担。

所用的地址配置协议是通过 dhcpcd 等守护程序实现的，而这些守护程序与基础架构中其他组件的交互很不通畅。为了持续识别接口，引入了新潮的接口命名模式，而这需要繁重的 udev 支持。

wicked 的理念是通过多种方式剖析问题。它没有采用任何全新的技术，而是尝试将不同项目中的观点集中起来，以建立一个更好的整体解决方案。

实现此目的的方法之一是使用客户端/服务器模型。wicked 可借此为地址配置等任务定义能够很好地集成到整个框架中的标准化工具。例如，使用特定的地址配置时，管理员可能要求应该通过 DHCP 或 IPv4 zeroconf 配置接口。在这种情况下，地址配置服务只会从它的服务器获得租用，并传递到安装了所请求地址和路由的 wicked 服务器进程。

剖析问题的另一种方法是强制实施分层机制。对于任何类型的网络接口，都可以定义一个 dbus 服务，用于配置网络接口的设备层 – VLAN、桥接、绑定或半虚拟化设备。地址配置等常用功能通过在这些设备特定的服务基础上分层的联合服务实现，您不必专门实施这些功能。

wicked 框架使用各种 dbus 服务来实现这两个方面的功能，这些服务将会根据其类型挂接到网络接口。本文提供了 wicked 中的当前对象层次的简要概述。

每个网络接口均以 `/org/opensuse/Network/Interfaces` 的子对象来表示。该子对象的名称由其 `ifindex` 指定。例如，通常 `ifindex` 为 1 的回写接口是 `/org/opensuse/Network/Interfaces/1`，注册的第一个以太网接口是 `/org/opensuse/Network/Interfaces/2`。

每个网络接口都有一个关联的“类”，该类用于选择该接口支持的 dbus 接口。默认情况下，每个网络接口的类为 `netif`，`wickedd` 将自动挂接与此类兼容的所有接口。在当前实施中，这些兼容的接口包括：

### **`org.opensuse.Network.Interface`**

一般网络接口功能，例如，打开或关闭链路、指派 MTU 等。

### **`org.opensuse.Network.Addrconf.ipv4.dhcp`**

### **`org.opensuse.Network.Addrconf.ipv6.dhcp`**

### **`org.opensuse.Network.Addrconf.ipv4.auto`**

适用于 DHCP、IPv4 zeroconf 等的地址配置服务

除此之外，网络接口可能还需要或者提供特殊的配置机制。例如，对于某个以太网设备，您应该能够控制链路速度、校验和卸载等。为了实现此目的，以太网设备都有一个名为 `netif-ethernet` 的自己的类，该类属于 `netif` 的子类。因此，指派给以太网接口的 dbus 接口具有上面列出的所有服务以及 `org.opensuse.Network.Ethernet` 服务，后者只适用于属于 `netif-ethernet` 类的对象。

同样，桥接、VLAN、绑定或 infiniband 等接口类型也存在适用类。

您要如何与某个首先需要创建的接口（例如 VLAN，它实际上是位于以太网设备上的虚拟网络接口）交互呢？为此，wicked 定义了出厂接口，例如 `org.opensuse.Network.VLAN.Factory`。这种出厂接口只提供单一功能，就是让您创建所请求类型的接口。这些出厂接口挂接到 `/org/opensuse/Network/Interfaces` 列表节点。

### 23.6.1.1 wicked 体系结构和功能

如图 23.4 “wicked 体系结构” 中所述，wicked 服务由几个部分组成。

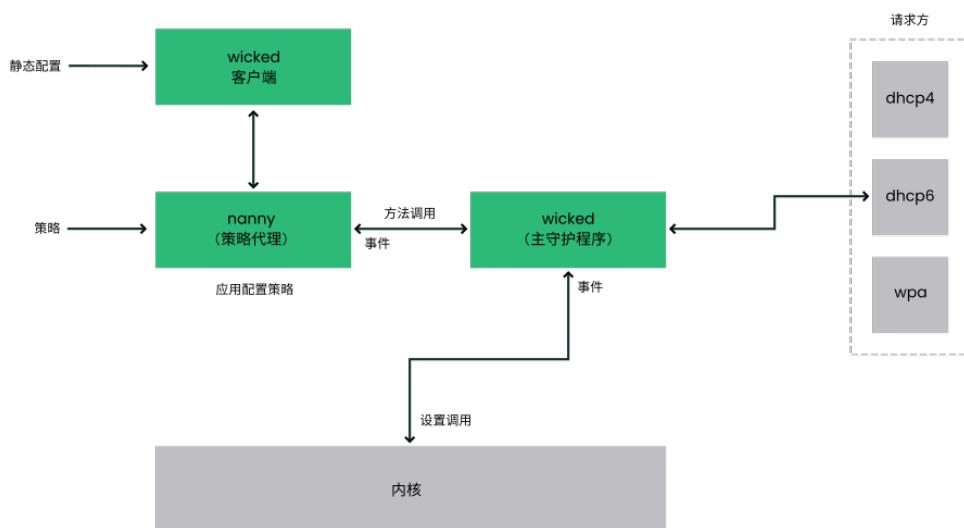


图 23.4：wicked 体系结构

wicked 目前支持以下功能：

- 使用配置文件后端来分析 SUSE 样式的 `/etc/sysconfig/network` 文件。
- 使用内部配置后端以 XML 格式表示网络接口配置。

- 打开和关闭“常规”网络接口，例如以太网或 InfiniBand，VLAN、网桥、绑定、tun、tap、虚设设备、macvlan、macvtap、hs1、qeth、iucv 和无线（当前限制为一个 wpa-psk/eap 网络）设备。
- 内置 DHCPv4 客户端和内置 DHCPv6 客户端。
- 默认启动的 `nanny` 守护程序有助于在设备可用（接口热插入）时自动激活配置的接口，以及在检测到链路（载波）时设置 IP 配置。有关更多信息，请参见[第 23.6.1.3 节“Nanny”](#)。
- `wicked` 过去是以与 `systemd` 集成的 DBus 服务组形式来实现的。因此，常用的 `systemctl` 命令都将适用于 `wicked`。

### 23.6.1.2 使用 `wicked`

在 SUSE Linux Enterprise 上，默认会运行 `wicked`。如果您要检查当前启用了哪个组件以及该组件是否正在运行，请调用：

```
systemctl status network
```

如果已启用 `wicked`，您将看到类似于下面的行：

```
wicked.service - wicked managed network interfaces
  Loaded: loaded (/usr/lib/systemd/system/wicked.service; enabled)
  ...

```

如果运行的是其他组件（例如 `NetworkManager`）并且您想切换到 `wicked`，请先停止正在运行的组件，然后启用 `wicked`：

```
systemctl is-active network && \
systemctl stop      network
systemctl enable --force wicked
```

如此会启用 `wicked` 服务、创建从 `network.service` 到 `wicked.service` 的别名链路，并在下次引导时启动网络。

启动服务器进程：

```
systemctl start wickedd
```

这将会启动 **wickedd** (主服务器) 和关联的请求方:

```
/usr/lib/wicked/bin/wickedd-auto4 --systemd --foreground  
/usr/lib/wicked/bin/wickedd-dhcp4 --systemd --foreground  
/usr/lib/wicked/bin/wickedd-dhcp6 --systemd --foreground  
/usr/sbin/wickedd --systemd --foreground  
/usr/sbin/wickedd-nanny --systemd --foreground
```

然后激活网络:

```
systemctl start wicked
```

或者使用 **network.service** 别名:

```
systemctl start network
```

这些命令使用 **/etc/wicked/client.xml** 中定义的默认或系统配置源。

要启用调试功能, 请在 **/etc/sysconfig/network/config** 中设置 **WICKED\_DEBUG**, 例如:

```
WICKED_DEBUG="all"
```

或者, 要省略一些信息:

```
WICKED_DEBUG="all,-dbus,-objectmodel,-xpath,-xml"
```

使用客户端实用程序显示所有接口的接口信息, 或者显示以 **IFNAME** 指定的接口的接口信息:

```
wicked show all  
wicked show IFNAME
```

XML 格式的输出:

```
wicked show-xml all  
wicked show-xml IFNAME
```

打开一个接口:

```
wicked ifup eth0  
wicked ifup wlan0
```

...

由于未指定配置源，wicked 客户端将检查 `/etc/wicked/client.xml` 中为它定义的默认配置源：

1. firmware: iSCSI 引导固件表 (iBFT)
2. compat: `ifcfg` 文件 – 为实现兼容性而实施

将会应用 `wicked` 从指定接口的这些源中获取的任何设置。预期的重要性顺序为 firmware、compat - 将来此顺序可能会发生变化。

有关更多信息，请参见 wicked 手册页。

### 23.6.1.3 Nanny

Nanny 是一个事件与策略驱动的守护程序，负责热插拔设备等异步或被动性方案。因此，Nanny 守护程序可帮助启动或者重启动延迟的设备或临时消失的设备。Nanny 将监视设备和链路变化，并集成当前策略集定义的新设备。即使 ifup 已经因指定的超时约束而退出，Nanny 也会继续设置。

默认情况下，Nanny 守护程序在系统上处于活动状态。您可在 `/etc/wicked/common.xml` 配置文件中启用该守护程序：

```
<config>
  ...
  <use-nanny>true</use-nanny>
</config>
```

如果使用此设置，`ifup` 和 `ifreload` 会将包含有效配置的策略应用到 Nanny 守护程序；然后，Nanny 将配置 `wickedd`，从而确保支持热插拔。它将在后台等待事件或更改（例如，打开新的设备或载体）。

### 23.6.1.4 启用多个接口

对于绑定和网桥，有效做法是在一个文件 (`ifcfg-bondX`) 中定义整个设备拓扑，并一次性将它激活。然后，当您指定（网桥或绑定的）顶级接口名称时，wicked 可以激活整个配置：

```
wicked ifup br0
```

此命令会按适当的顺序自动设置网桥及其依赖项，而无需分别列出依赖项（端口等）。

要在一个命令中激活多个接口：

```
wicked ifup bond0 br0 br1 br2
```

要激活所有接口：

```
wicked ifup all
```

### 23.6.1.5 将隧道与 wicked 配合使用

如果您需要将隧道与 Wicked 结合使用，可以使用专门针对此用途的 TUNNEL\_DEVICE。它可让您指定一个可选的设备名称，以将隧道绑定至该设备。隧道式包将只能通过此设备路由。

有关详细信息，请参见 man 5 ifcfg-tunnel。

### 23.6.1.6 处理增量变化

有了 wicked，当您要重新配置某个接口时，并不需要真正将它关闭（除非内核有此要求）。例如，要将另一个 IP 地址或路由添加到静态配置的网络接口，请将该 IP 地址添加到接口定义，然后再次执行“ifup”操作。服务器会尽量做到只更新那些已更改的设置。这适用于链路级选项，例如设备 MTU 或 MAC 地址；也适用于网络级设置，例如地址、路由，甚至地址配置模式（例如，从静态配置转为 DHCP 时）。

当然，对于合并了多个实体设备（例如桥接或绑定设备）的虚拟接口，事情会变得有些棘手。对于绑定设备，当设备运行时，您无法更改某些参数，否则会导致出错。

但是，您仍可以添加或去除绑定设备或桥接的子设备，或者选择绑定设备的主接口。

### 23.6.1.7 Wicked 扩展：地址配置

wicked 设计为使用外壳脚本扩展。这些扩展可在 config.xml 文件中定义。

目前支持多个种类的扩展：

- 链路配置：这些脚本负责根据客户端提供的配置来设置设备的链路层，以及负责将链路层再次拆解。
- 地址配置：这些脚本负责管理设备的地址配置。通常，地址配置和 DHCP 由 **wicked** 自身管理，但是，可借助扩展来实现。
- 防火墙扩展：这些脚本可以应用防火墙规则。

通常，扩展中包含一个启动命令和一个停止命令、一个可选的“pid 文件”，以及传递给脚本的一组环境变量。

为了演示此扩展的工作原理，请查看 etc/server.xml 中定义的防火墙扩展：

```
<dbus-service interface="org.opensuse.Network.Firewall">
  <action name="firewallUp"    command="/etc/wicked/extensions/firewall up"/>
  <action name="firewallDown"  command="/etc/wicked/extensions/firewall down"/>

  <!-- default environment for all calls to this extension script -->
  <putenv name="WICKED_OBJECT_PATH" value="$object-path"/>
  <putenv name="WICKED_INTERFACE_NAME" value="$property:name"/>
  <putenv name="WICKED_INTERFACE_INDEX" value="$property:index"/>
</dbus-service>
```

该扩展挂接到 <dbus-service> 标记，并定义了用于进行此接口的各项操作所需执行的命令。此外，声明可以定义并初始化传递给操作的环境变量。

### 23.6.1.8 Wicked 扩展：配置文件

您也可以使用脚本来扩展配置文件的处理。例如，extensions/resolver 脚本根据 server.xml 中配置的行为来最终处理租用中的 DNS 更新：

```
<system-updater name="resolver">
  <action name="backup" command="/etc/wicked/extensions/resolver backup"/>
  <action name="restore" command="/etc/wicked/extensions/resolver restore"/>
  <action name="install" command="/etc/wicked/extensions/resolver install"/>
  <action name="remove" command="/etc/wicked/extensions/resolver remove"/>
</system-updater>
```

当 `wickedd` 收到更新时，系统更新程序例程将分析租用，并调用解析程序脚本中的适当命令（`backup`、`install` 等）。此后便可以使用 `/sbin/netconfig` 或者通过手动写入 `/run/netconfig/resolv.conf`（作为回退）来配置 DNS 设置。

## 23.6.2 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。

### 23.6.2.1 /etc/wicked/common.xml

`/etc/wicked/common.xml` 文件包含所有应用程序都应使用的通用定义。该文件源自/包含在此目录中的其他配置文件中。尽管您可以使用此文件允许在所有 `wicked` 组件间进行调试，但建议使用 `/etc/wicked/local.xml` 文件来实现此目的。执行维护更新后，`/etc/wicked/common.xml` 可能会被覆盖，因此您所做的更改可能会丢失。`/etc/wicked/common.xml` 文件包含默认安装中的 `/etc/wicked/local.xml`，因此您通常不需要修改 `/etc/wicked/common.xml`。

如果要通过将 `<use-nanny>` 设置为 `false` 来禁用 `nanny`，请重启动 `wickedd.service`，然后运行以下命令以应用所有配置和策略：

```
> sudo wicked ifup all
```



#### 注意：配置文件

如果 `wickedd`、`wicked` 或 `nanny` 程序自己的配置文件不存在，它们会尝试读取 `/etc/wicked/common.xml`。

### 23.6.2.2 /etc/wicked/server.xml

`wickedd` 服务器进程会在启动时读取文件 `/etc/wicked/server.xml`。该文件将扩展存储到 `/etc/wicked/common.xml` 中。除此之外，此文件可配置解析程序的处理方式，以及从 `addrconf` 请求方（例如 DHCP）接收信息的方式。

建议您将需要对此文件进行的更改添加到单独的文件 `/etc/wicked/server-local.xml` (`/etc/wicked/server.xml` 会包含该文件)。使用单独的文件可避免在维护更新期间覆盖您的更改。

### 23.6.2.3 `/etc/wicked/client.xml`

`/etc/wicked/client.xml` 供 **wicked** 命令使用。该文件指定发现 ibft 管理的设备时所用脚本的位置，并可配置网络接口配置的位置。

建议您将需要对此文件进行的所有更改都添加到单独的文件 `/etc/wicked/client-local.xml` (`/etc/wicked/server.xml` 会包含该文件的内容)。使用单独的文件可避免在维护更新期间覆盖您的更改。

### 23.6.2.4 `/etc/wicked/nanny.xml`

`/etc/wicked/nanny.xml` 配置链接层的类型。建议您将特定配置添加到单独的文件 `/etc/wicked/nanny-local.xml` 中，以免在进行维护更新期间丢失所作的更改。

### 23.6.2.5 `/etc/sysconfig/network/ifcfg-*`

这些文件包含网络接口的传统配置。



#### 注意： **wicked** 和 `ifcfg-*` 文件

如果您指定 `compat:` 前缀，**wicked** 会读取这些文件。根据 `/etc/wicked/client.xml` 中 SUSE Linux Enterprise Desktop 的默认配置，**wicked** 会尝试先读取这些文件，然后再读取 `/etc/wicked/ifconfig` 中的 XML 配置文件。

提供的 `--ifconfig` 开关主要用于测试。如果指定该开关，则不会应用 `/etc/wicked/ifconfig` 中定义的默认配置源。

ifcfg-\* 文件包含启动模式和 IP 地址等信息。可能的参数在 ifup 的手册页中有所介绍。此外，如果一个常规设置只能用于一个接口，则文件 dhcp 和 wireless 中的大多数变量在 ifcfg-\* 文件中都可以使用。但是，/etc/sysconfig/network/config 中的大多数变量是全局变量，无法在 ifcfg 文件中覆盖。例如，NETCONFIG\_\* 变量就是全局变量。要配置 macvlan 和 macvtap 接口，请参见 ifcfg-macvlan 和 ifcfg-macvtap 手册页。例如，对于 macvlan 接口，请提供使用以下设置的 ifcfg-macvlan0：

```
STARTMODE='auto'  
MACVLAN_DEVICE='eth0'  
#MACVLAN_MODE='vepa'  
#LLADDR=02:03:04:05:06:aa
```

有关 ifcfg.template，请参见 第 23.6.2.6 节 “/etc/sysconfig/network/config, /etc/sysconfig/network/dhcp, 和 /etc/sysconfig/network/wireless”。

### 23.6.2.6 /etc/sysconfig/network/config, /etc/sysconfig/network/dhcp, 和 /etc/sysconfig/network/wireless

文件 config 包含 ifup、ifdown 和 ifstatus 行为的常规设置。dhcp 包含 DHCP 设置，wireless 包含无线局域网卡的设置。所有三个配置文件中的变量均已注释掉。/etc/sysconfig/network/config 中的一些变量也可以在 ifcfg-\* 文件中使用，而且在这些文件中它们的优先级更高。/etc/sysconfig/network/ifcfg.template 文件列出了可以按接口指定的变量。但是，/etc/sysconfig/network/config 中的大多数变量是全局变量，无法在 ifcfg-files 中覆盖。例如，NETWORKMANAGER 或 NETCONFIG\_\* 变量就是全局变量。



#### 注意：使用 DHCPv6

在 SUSE Linux Enterprise 11 中，即使是在未正确配置 IPv6 路由器广播 (RA) 的网络中，DHCPv6 一向也能正常工作。从 SUSE Linux Enterprise 12 开始，DHCPv6 要求网络中至少有一个路由器发出 RA，用于指示此网络由 DHCPv6 管理。

对于无法在其中正确配置路由器的网络，用户可以通过在 ifcfg 文件中指定 DHCLIENT6\_MODE='managed'，使用 ifcfg 选项来覆盖此行为。您也可以在安装系统中使用引导参数来启用这种解决方法：

```
ifcfg=eth0=dhcp6,DHCLIENT6_MODE=managed
```

### 23.6.2.7 /etc/sysconfig/network/routes 和 /etc/sysconfig/network/ifroute-\*

TCP/IP 包的静态路由是由 `/etc/sysconfig/network/routes` 和 `/etc/sysconfig/network/ifroute-*` 文件确定的。可以在 `/etc/sysconfig/network/routes` 中指定各种系统任务所需的所有静态路由：到主机的路由、通过网关到主机的路由以及到网络的路由。对于需要个别路由的每个接口，请定义另一个配置文件：`/etc/sysconfig/network/ifroute-*`。将通配符 (\*) 替换为接口名称。路由选择配置文件中的项如下所示：

| # | Destination | Gateway | Netmask | Interface | Options |
|---|-------------|---------|---------|-----------|---------|
|---|-------------|---------|---------|-----------|---------|

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。应该以 CIDR 表示法（地址加上关联的路由前缀长度）输入网络名称，例如 `10.10.0.0/16`（对于 IPv4 路由）或 `fc00::/7`（对于 IPv6 路由）。关键字 `default` 表示该路由是与网关位于相同地址族中的默认网关。对于没有网关的设备，请使用显式 `0.0.0.0/0` 或 `::/0` 目标。

第二列包含默认网关或通过其可访问主机或网络的网关。

第三列已弃用；它用于包含目标的 IPv4 网络掩码。对于 IPv6 路由、默认路由，或者如果在第一列中使用了前缀长度（CIDR 表示法），请在此处输入破折号 (-)。

第四列包含接口名称。如果使用破折号 (-) 将它保留空白，可能会导致 `/etc/sysconfig/network/routes` 出现意外的行为。有关更多信息，请参见 `routes` 手册页。

第五列（可选）可用于指定特殊选项。有关详细信息，请参见 `routes` 手册页。

#### 例 23.5：通用网络接口和一些静态路由

```
# --- IPv4 routes in CIDR prefix notation:
# Destination      [Gateway]      -      Interface
127.0.0.0/8        -            -      lo
204.127.235.0/24  -            -      eth0
default            204.127.235.41 -      eth0
207.68.156.51/32  207.68.145.45 -      eth1
```

```

192.168.0.0/16      207.68.156.51      -          eth1

# --- IPv4 routes in deprecated netmask notation"
# Destination      [Dummy/Gateway]  Netmask          Interface
#
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

# --- IPv6 routes are always using CIDR notation:
# Destination      [Gateway]          -          Interface
2001:DB8:100::/64  -          -          eth0
2001:DB8:100::/32  fe80::216:3eff:fe6d:c042 -          eth0

```

### 23.6.2.8 [/var/run/netconfig/resolv.conf](#)

主机所属的域在 [/var/run/netconfig/resolv.conf](#) 中指定（关键字 `search`）。使用 `search` 选项最多可以指定六个域，总共 256 个字符。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。使用 `nameserver` 选项最多可以指定三个名称服务器，每行指定一个。注释以井号或分号（# 或 ;）开头。有关示例，请参见[例 23.6 “/var/run/netconfig/resolv.conf”](#)。

但是，您不应手动编辑 [/etc/resolv.conf](#)。此文件由 `netconfig` 脚本生成，是 [/run/netconfig/resolv.conf](#) 的符号链接。要定义静态 DNS 配置而不使用 YaST，请手动编辑 [/etc/sysconfig/network/config](#) 文件中的适当变量：

#### NETCONFIG\_DNS\_STATIC\_SEARCHLIST

用于主机名查找的 DNS 域名列表

#### NETCONFIG\_DNS\_STATIC\_SERVERS

用于主机名查找的名称服务器 IP 地址列表

#### NETCONFIG\_DNS\_FORWARDER

需要配置的 DNS 转发器名称，例如 `bind` 或 `resolver`

## NETCONFIG\_DNS\_RESOLVER\_OPTIONS

将写入 /var/run/netconfig/resolv.conf 的任意选项，例如：

```
debug attempts:1 timeout:10
```

有关更多信息，请参见 resolv.conf 手册页。

## NETCONFIG\_DNS\_RESOLVER\_SORTLIST

最多包含 10 项的列表，例如：

```
130.155.160.0/255.255.240.0 130.155.0.0
```

有关更多信息，请参见 resolv.conf 手册页。

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 netconfig(8) 手册页 (man 8 netconfig)。

例 23.6：/var/run/netconfig/resolv.conf

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

### 23.6.2.9 /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服务或相似操作的 netconfig 模块将所需更改应用于系统。

netconfig 可以识别三种主要操作。netconfig modify 和 netconfig remove 命令由诸如 DHCP 或 PPP 的守护程序用于在 netconfig 中提供设置或从中删除设置。用户仅可使用 netconfig update 命令：

## **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 会从标准输入或从使用 `--lease-file FILENAME` 选项指定的文件中读取设置，并将其存储于内部，直到系统重引导（或者执行下一个修改或删除操作）为止。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去除由编辑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。如果只想更新指定的服务（`dns`、`nis` 或 `ntp`），请使用 `-m MODULE_TYPE` 参数。

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。自动配置工具（例如 DHCP 或 PPP）提供的动态配置设置由这些工具通过 **netconfig modify** 和 **netconfig remove** 操作直接递送。启用 NetworkManager 时，netconfig（在策略模式 `auto` 中）仅使用 NetworkManager 设置，忽略使用传统 ifup 方法配置的任何其他接口的设置。如果 NetworkManager 未提供任何设置，将使用静态设置作为后备设置。不支持混合使用 NetworkManager 和 **wicked** 方法。

有关 **netconfig** 的详细信息，请参见 `man 8 netconfig`。

### 23.6.2.10 `/etc/hosts`

在此文件中，如例 23.7 “`/etc/hosts`” 中所示，将为主机名指派 IP 地址。如果未实施名称服务器，则将与其建立 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数据，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。注释一律以 `#` 符号开头。

#### 例 23.7：/etc/hosts

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

#### 23.6.2.11 /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 23.8 “/etc/networks” 。

#### 例 23.8：/etc/networks

```
loopback      127.0.0.0
localnet      192.168.0.0
```

#### 23.6.2.12 /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。每个参数都必须始终在单独的一行中输入。注释以 `#` 符号开头。表 23.2 “/etc/host.conf 的参数” 显示了可用的参数。例 23.9 “/etc/host.conf” 中显示了 `/etc/host.conf` 的示例。

表 23.2：/ETC/HOST.CONF 的参数

|                                  |  |
|----------------------------------|--|
| order <b>hosts</b> 、 <b>bind</b> | 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）：<br><br><b>hosts</b> : 搜索 <code>/etc/hosts</code> 文件<br><br><b>bind</b> : 访问名称服务器<br><br><b>nis</b> : 使用 NIS |
| multi <b>on/off</b>              | 定义 <code>/etc/hosts</code> 中输入的主机是否可以具有多个 IP 地址。   |

|  |   |
|--|---|
| <b>nospoof on</b> <b>spoofalert on/off</b> | 这些参数影响名称服务器 <b>spoofing</b> ，但对网络配置没有任何影响。  |
| <b>trim domainname</b>                     | 在主机名解析后，指定的域名将与主机名分开（前提是主机名包含域名）。只有在本地域分隔出来的名称包含在 <u>/etc/hosts</u> 文件中，但仍应通过附带的域名进行识别时，才可以使用此选项。 |

#### 例 23.9：/etc/host.conf

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

#### 23.6.2.13 /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 库参考手册**》。

查询的顺序是在文件 /etc/nsswitch.conf 中定义的。例 23.10 “/etc/nsswitch.conf” 中显示了 nsswitch.conf 的示例。注释以 # 符号开头。在此例中，hosts 数据库下的项表示通过 DNS 将请求发送到 /etc/hosts (files)。

#### 例 23.10：/etc/nsswitch.conf

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files
```

```

rpc:      files
ethers:   files
netmasks: files
netgroup: files nis
publickey: files

bootparams: files
automount:  files nis
aliases:    files nis
shadow:     compat

```

表 23.3 “通过 `/etc/nsswitch.conf` 可用的数据库” 中列出了 NSS 上可用的“数据库”。表 23.4 “NSS “数据库” 的配置选项” 中列出了 NSS 数据库的配置选项。

表 23.3：通过 `/ETC/NSSWITCH.CONF` 可用的数据库

|                  |   |
|------------------|---|
| <u>aliases</u>   | <code>sendmail</code> 实施的邮件别名；请参见 <a href="#">man 5 aliases</a> 。 |
| <u>ethers</u>    | 以太网地址。  |
| <u>netmasks</u>  | 网络及其子网掩码的列表。只有在使用子网划分时才需要。  |
| <u>group</u>     | <code>getgrent</code> 使用的用户组。另请参见 <a href="#">group</a> 的手册页。     |
| <u>hosts</u>     | <code>gethostbyname</code> 和类似函数使用的主机名和 IP 地址。                    |
| <u>netgroup</u>  | 在网络中用于控制访问权限的有效主机和用户列表，请参见 <a href="#">netgroup(5)</a> 手册页。       |
| <u>networks</u>  | <code>getnetent</code> 使用的网络名称和地址。                                |
| <u>publickey</u> | NFS 和 NIS+ 使用的 Secure_RPC 的公钥和密钥。                                 |

|                  |   |
|------------------|---|
| <u>passwd</u>    | <u>getpwent</u> 使用的用户口令；请参见 <u>passwd(5)</u> 手册页。       |
| <u>protocols</u> | <u>getprotoent</u> 使用的网络协议；请参见 <u>protocols(5)</u> 手册页。 |
| <u>rpc</u>       | <u>getrpcbyname</u> 和类似函数使用的远程过程调用名称和地址。                |
| <u>services</u>  | <u>getservent</u> 使用的网络服务。                              |
| <u>shadow</u>    | <u>getspnam</u> 使用的用户阴影口令；请参见 <u>shadow(5)</u> 手册页。     |

表 23.4： NSS “数据库” 的配置选项

|                             |   |
|-----------------------------|---|
| <u>files</u>                | 直接访问文件，例如 <u>/etc/aliases</u>                         |
| <u>db</u>                   | 通过数据库访问   |
| <u>nis</u> , <u>nisplus</u> | NIS，另请参见《安全和强化指南》，第 3 章<br>“使用 NIS”                   |
| <u>dns</u>                  | 仅可用作 <u>hosts</u> 和 <u>networks</u> 的扩展               |
| <u>compat</u>               | 仅可用作 <u>passwd</u> 、 <u>shadow</u> 和 <u>group</u> 的扩展 |

### 23.6.2.14 /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，nscd 会缓存 passwd、groups 和 hosts 的系统项。这对 NIS 和 LDAP 等目录服务的性能而言非常重要，否则，每次访问名称、组或主机都需要网络连接。

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。使用以下命令重启 nscd，缩短这段等待时间：

```
> sudo systemctl restart nscd
```

### 23.6.2.15 /etc/HOSTNAME

/etc/HOSTNAME 包含完全限定的主机名 (FQHN)。完全限定的主机名是附有域名的主机名。此文件只能包含一行（在此行中设置主机名）。计算机引导时会读取此文件。

## 23.6.3 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。

命令 ip 会直接更改网络配置，而不会将其保存到配置文件中。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。



### 注意：ifconfig 和 route 已过时

ifconfig 和 route 工具已过时。请改用 ip。例如，ifconfig 会将接口名限制为 9 个字符。

### 23.6.3.1 使用 ip 配置网络接口

ip 是用来显示和配置网络设备、路由选择、策略路由选择以及隧道的工具。

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：

#### link

此对象表示网络设备。

#### address

此对象表示设备的 IP 地址。

#### neighbor

此对象表示 ARP 或 NDISC 缓存项。

## route

此对象表示路由选择表项。

## rule

此对象表示路由选择策略数据库中的规则。

## maddress

此对象表示多播地址。

## mroute

此对象表示多播路由缓存项。

## 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](#)）。

使用以下命令更改设备的状态：

```
> sudo ip link set DEV_NAME
```

例如，要停用设备 eth0，请输入

```
> sudo ip link set eth0 down
```

要再次激活该设备，请使用

```
> sudo ip link set eth0 up
```



## 提示：断开 NIC 设备的连接

如果使用以下命令停用某个设备

```
> sudo ip link set DEV_NAME down
```

该命令会在软件级别禁用网络接口。

如果您想要模拟拔下以太网网线或关闭所连接的交换机之类的链接丢失，请运行

```
> sudo ip link set DEV_NAME carrier off
```

例如，`ip link set DEV_NAME down`会丢弃使用`DEV_NAME`的所有路由，而`ip link set DEV carrier off`则不会。请注意，`carrier off`需要网络设备驱动程序的支持。

要将设备再次连接回物理网络，请运行

```
> sudo ip link set DEV_NAME carrier on
```

激活设备后，可对设备进行配置。要设置IP地址，请使用

```
> sudo ip addr add IP_ADDRESS + dev DEV_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`，请输入

```
> sudo ip addr add 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

```
> sudo ip route add default via gateway_ip_address
```

要显示所有设备，请使用

```
> sudo ip link ls
```

要只显示正在运行的接口，请使用

```
> sudo ip link ls up
```

要输出设备的接口统计信息，请输入

```
> sudo ip -s link ls DEV_NAME
```

要查看其他有用信息（特别是有关虚拟网络设备的信息），请输入

```
> sudo ip -d link ls DEV_NAME
```

另外，要查看设备的网络层（IPv4、IPv6）地址，请输入

```
> sudo ip addr
```

在输出中，可以找到有关设备MAC地址的信息。要显示所有路由，请使用

```
> sudo ip route show
```

有关使用 **ip** 的详细信息, 请输入 **ip help** 或参见 **man 8 ip** 手册页。**help** 选项还可用于所有 **ip** 子命令, 例如:

```
> sudo ip addr help
```

可在 **/usr/share/doc/packages/iproute2/ip-cref.pdf** 中找到 **ip** 手册。

### 23.6.3.2 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。如果成功, **ping** 将显示表示这一结果的消息。这表示网络链路正在运作。

**ping** 不仅能测试两台计算机之间的连接功能: 它还能提供关于连接质量的一些基本信息。

在**例 23.11 “命令 ping 的输出”** 中, 可查看 **ping** 输出示例。倒数第二行包含有关已传输的包数、丢失的包和 **ping** 的总运行时间的信息。

对于目标, 您可以使用主机名或 IP 地址, 例如 **ping example.com** 或 **ping 192.168.3.100**。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** 选项来限制数据包数量。例如, 要将 **ping** 限制为三个数据包, 请输入 **ping -c 3 example.com**。

#### 例 23.11：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** 来更改间隔。例如, 要将 **ping** 间隔增加到十秒, 请输入 **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** 选项与选定设备名称搭配使用, 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。



## 提示: Ping IPv6 地址

对于 IPv6 地址, 请使用 `ping6` 命令。请注意, 要 ping 链接-本地地址, 必须使用 `-I` 指定接口。如果通过 `eth1` 可获取地址, 则以下命令有效:

```
ping6 -I eth1 fe80::117:21ff:fed:a425
```

## 23.6.4 单元文件和启动脚本

除了上面介绍的配置文件之外, 还存在一些负责在引导计算机时装载网络服务的 `systemd` 单元文件和多个脚本。系统切换到 `multi-user.target` 目标后, 即会启动这些文件和脚本。[网络程序的一些单元文件和启动脚本](#) 中介绍了其中的部分单元文件和脚本。有关 `systemd` 的详细信息, 请参见[第 19 章 “systemd 守护程序”](#) ; 有关 `systemd` 目标的详细信息, 请参见 `systemd.special` 的手册页 (`man systemd.special`)。

### 网络程序的一些单元文件和启动脚本

#### `network.target`

`network.target` 是网络的 `systemd` 目标, 但其具体含义取决于系统管理员提供的设置。

有关详细信息, 请参见 <http://www.freedesktop.org/wiki/Software/systemd/NetworkTarget/>。

#### `multi-user.target`

`multi-user.target` 是包含所有必需网络服务的多用户系统的 `systemd` 目标。

#### `rpcbind`

启动用于将 RPC 程序号转换为通用地址的 `rpcbind` 实用程序。它是 RPC 服务所必需的, 如 NFS 服务器。

#### `ypserv`

启动 NIS 服务器。

## ypbind

启动 NIS 客户端。

## /etc/init.d/nfsserver

启动 NFS 服务器。

## /etc/init.d/postfix

控制 postfix 进程。

## 23.7 设置绑定设备

对于某些系统，需要实施高于典型以太网设备的标准数据安全性或可用性要求的网络连接。在这些情况下，可以将多个以太网设备聚合到单个绑定设备。

绑定设备的配置通过绑定模块选项来完成。其行为主要受绑定设备模式的影响。该模式默认为 active-backup，也就是说，如果活动端口发生故障，另一个绑定端口将变成活动端口。可以使用以下绑定模式：

### 0 (balance-rr)

数据包依次通过第一个到最后一个可用接口传输。提供容错和负载平衡。

### 1 (active-backup)

只有一个网络接口处于活动状态。如果它发生故障，另一个接口将变成活动状态。此设置是 SUSE Linux Enterprise Desktop 的默认设置。提供容错。

### 2 (balance-xor)

流量会根据绑定中包含的设备数量在所有可用接口间拆分。这需要交换机的支持。提供容错和负载平衡。

### 3 (broadcast)

在所有接口上广播所有通讯。需要交换机的支持。提供容错。

### 4 (802.3ad)

将接口聚合成共享相同速度和双工设置的组。需要接口驱动程序中的 ethtool 支持，以及支持 IEEE 802.3ad 动态链路聚合并进行了相应配置的交换机。提供容错和负载平衡。

## 5 (balance-tlb)

自适应传输负载平衡。需要接口驱动程序中的 `ethtool` 支持，但不需要交换机支持。提供容错和负载平衡。

## 6 (balance-alb)

自适应负载平衡。需要接口驱动程序中的 `ethtool` 支持，但不需要交换机支持。提供容错和负载平衡。

有关各种模式的详细说明，请参见<https://www.kernel.org/doc/Documentation/networking/bonding.txt>。



### 提示：绑定和 Xen

绑定设备只对于有多个真实网卡可用的计算机有效。这意味着在大多数配置中，您仅应在 Dom0 中使用绑定配置。换言之，只有当您将多个网卡指派给一个 VM Guest 系统时，在 VM Guest 中设置绑定才有效。



### 注意：IBM POWER: ibmveth 不支持绑定模式 5 和 6 (balance-tlb/balance-alb)

tlb/alb 绑定配置与 Power 固件之间存在冲突。简单而言，处于 tlb/alb 模式的绑定驱动程序会发送同时包含源和目标 MAC 地址（列作虚拟以太网 MAC 地址）的以太网回写包。Power 固件并不支持这些包。因此，ibmveth 不支持绑定模式 5 和 6。

要配置绑定设备，请使用以下过程：

1. 运行 YaST > 系统 > 网络设置。
2. 使用添加并将设备类型更改为绑定。单击下一步继续。



3. 选择如何为绑定设备指派 IP 地址。有三种方法可供选择：

- 无 IP 地址
- 动态地址（使用 DHCP 或 Zeroconf）
- 静态指派的 IP 地址

请使用最适合您环境的方法。

4. 在绑定端口选项卡中，选中相关复选框选择应加入到绑定中的以太网设备。
5. 编辑绑定驱动程序选项并选择绑定模式。
6. 确保将参数 miimon=100 添加到绑定驱动程序选项。如果没有此参数，则不会定期检查数据完整性。
7. 单击下一步，然后单击确定退出 YaST 以创建设备。

### 23.7.1 绑定端口的热插拔

在特定网络环境（如高可用性）中，有时需要将某个绑定端口接口替换为另一个。原因可能在于网络设备持续故障。解决方案是设置绑定端口的热插拔。

按常规配置绑定（按照 man 5 ifcfg-bonding），例如：

```
ifcfg-bond0
  STARTMODE='auto' # or 'onboot'
  BOOTPROTO='static'
  IPADDR='192.168.0.1/24'
  BONDING_MASTER='yes'
  BONDING_SLAVE_0='eth0'
  BONDING_SLAVE_1='eth1'
  BONDING_MODULE_OPTS='mode=active-backup miimon=100'
```

使用 STARTMODE=hotplug 和 BOOTPROTO=none 可指定绑定端口：

```
ifcfg-eth0
  STARTMODE='hotplug'
  BOOTPROTO='none'

ifcfg-eth1
  STARTMODE='hotplug'
  BOOTPROTO='none'
```

BOOTPROTO=none 使用 ethtool 选项（如果提供），但不会在 ifup eth0 上设置链路，这是因为绑定端口接口由绑定设备控制。

STARTMODE=hotplug 会使绑定端口接口在可用时自动加入绑定。

需要更改 /etc/udev/rules.d/70-persistent-net.rules 中的 udev 规则，以便按总线 ID (udev KERNELS 关键字等同于 hwinfo --netcard 中的“SysFS BusID” ) 而不是 MAC 地址匹配设备。这样将允许更换有缺陷的硬件（位于同一插槽但 MAC 不同的网卡），并避免在绑定更改其所有绑定端口的 MAC 地址时出现混淆。

例如：

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
KERNELS=="0000:00:19.0", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"
```

在引导时，systemd network.service 不会等待热插拔绑定端口就绪，而是等待绑定就绪，后者至少需要有一个绑定端口可用。当从系统中去除一个绑定端口接口（从 NIC 驱动程序解除绑定、执行 NIC 驱动程序的 rmmmod 命令或真正去除 PCI 热插拔）时，内核会自动从将其绑定中去除。当向系统添加新网卡时（更换插槽中的硬件），udev 会使用基于总线的永久名称规则将其重命名为绑定端口的名称，并为其调用 ifup。调用 ifup 会自动将新卡加入绑定。

## 23.8 设置组合设备以进行网络协作

“链路聚合”属于通用术语，指组合（或聚合）网络连接以提供逻辑层。有时，您还会看到“通道协作”、“以太网绑定”、“端口截断”等术语，这些同义词都是指同一个概念。这个概念通常被称为“绑定”，最初是集成到 Linux 内核中的（请参见[第 23.7 节 “设置绑定设备”](#)了解最初的实现方式）。**网络协作**一词用来指代这个概念的新实现方式。

绑定和网络协作之间的主要差别在于，协作提供一组小型内核模块，由它们负责提供用于 teamd 实例的接口。其他一切都在用户空间中处理。这一点与最初的绑定实现方式不同，后者是将自己的所有功能都专门包含在内核中。如需两者的比较，请参见[表 23.5 “绑定与组合的功能比较”](#)。

表 23.5：绑定与组合的功能比较

| 功能                  | 绑定 | 组合 |
|---------------------|----|----|
| 广播、循环 TX 策略         | 是  | 是  |
| 活动备份 TX 策略          | 是  | 是  |
| LACP (802.3ad) 支持   | 是  | 是  |
| 基于哈希的 TX 策略         | 是  | 是  |
| 用户可以设置哈希函数          | 否  | 是  |
| TX 负载平衡支持 (TLB)     | 是  | 是  |
| 针对 LACP 的 TX 负载平衡支持 | 否  | 是  |
| Ethtool 链接监视        | 是  | 是  |
| ARP 链接监视            | 是  | 是  |
| NS/NA (IPV6) 链接监视   | 否  | 是  |
| 针对 TX/RX 路径的 RCU 锁定 | 否  | 是  |
| 端口优先级和粘性            | 否  | 是  |
| 单独的按端口链接监视设置        | 否  | 是  |

| 功能      | 绑定  | 组合 |
|---------|-----|----|
| 多链接监视设置 | 有限制 | 是  |
| VLAN 支持 | 是   | 是  |
| 多设备堆叠   | 是   | 是  |

源: <http://libteam.org/files/teamdev.pp.pdf>

绑定和网络协作这两种实现方式可以并行使用。可将网络协作作为现有绑定实现方式的备选。它不会取代绑定。

网络协作可用于不同使用情况。稍后将会介绍其中两种最重要的使用情况，它们涉及：

- 不同网络设备之间的负载平衡。
- 从一个网络设备到另一个网络设备的故障转移（当其中一个设备出现故障时）。

目前没有用于支持创建协作设备的 YaST 模块。您需要手动配置网络协作。以下是适用于所有网络协作配置的一般过程：

#### 过程 23.1：一般过程

1. 确保已安装所有必需的软件包。安装 `libteam-tools`、`libteamdctl0` 和 `python-libteam` 软件包。
2. 在 `/etc/sysconfig/network/` 下创建一个配置文件，通常为 `ifcfg-team0`。如果您需要多个网络协作设备，请为它们指定依次递增的编号。  
此配置文件包含数个变量，手册页中有相关说明（请参见 `man ifcfg` 和 `man ifcfg-team`）。系统内的 `/etc/sysconfig/network/ifcfg.template` 文件中提供了示例配置。
3. 去除将用于协作设备的接口的配置文件（通常为 `ifcfg-eth0` 和 `ifcfg-eth1`）。  
建议您先备份这两个文件，然后再将其去除。Wicked 将会使用协作的必要参数重新创建配置文件。
4. (可选) 检查 Wicked 的配置文件中是否已包含所有内容：

```
> sudo wicked show-config
```

## 5. 启动网络协作设备 team0:

```
> sudo wicked ifup all team0
```

如果您需要其他调试信息, 请在 all 子命令后面使用 --debug all 选项。

## 6. 检查网络协作设备的状态。通过执行以下命令可以完成该操作:

- 从 Wicked 获取 teamd 实例的状态:

```
> sudo wicked ifstatus --verbose team0
```

- 获取整个实例的状态:

```
> sudo teamdctl team0 state
```

- 获取 teamd 实例的 systemd 状态:

```
> sudo systemctl status teamd@team0
```

以上各命令将根据您的需要分别显示稍有不同的视图。

## 7. 如果您之后需要对 ifcfg-team0 文件中的内容进行更改, 请使用以下命令重新装载其配置:

```
> sudo wicked ifreload team0
```

请勿使用 systemctl 来启动或停止协作设备! 而是使用如上所示的 wicked 命令。

要彻底去除组合设备, 请执行以下过程:

### 过程 23.2 : 去除组合设备

#### 1. 停止网络组合设备 team0:

```
> sudo wicked ifdown team0
```

#### 2. 将文件 /etc/sysconfig/network/ifcfg-team0 重命名为 /etc/sysconfig/network/.ifcfg-team0。在文件名前面插入一个点, 以使 wicked “看不到” 它。如果您确实不再需要该配置, 也可以去除该文件。

### 3. 重新装载配置：

```
> sudo wicked ifreload all
```

## 23.8.1 用例：网络协作的负载平衡

负载平衡用于提高带宽。使用下面的配置文件可创建具有负载平衡功能的网络协作设备。继续[过程 23.1 “一般过程”](#)以设置设备。使用 `teamdctl` 检查输出。

例 23.12：网络协作的负载平衡配置

```
STARTMODE=auto ①
BOOTPROTO=static ②
IPADDRESS="192.168.1.1/24" ②
IPADDR6="fd00:deca:fbad:50::1/64" ②

TEAM_RUNNER="loadbalance" ③
TEAM_LB_TX_HASH="ipv4,ipv6,eth,vlan"
TEAM_LB_TX_BALANCER_NAME="basic"
TEAM_LB_TX_BALANCER_INTERVAL="100"

TEAM_PORT_DEVICE_0="eth0" ④
TEAM_PORT_DEVICE_1="eth1" ④

TEAM_LW_NAME="ethtool" ⑤
TEAM_LW_ETHTOOL_DELAY_UP="10" ⑥
TEAM_LW_ETHTOOL_DELAY_DOWN="10" ⑥
```

① 控制协作设备的启动。`auto` 这个值表示接口将在网络服务可用时设置，并且会在每次重引导时自动启动。

如果您需要自行控制设备（并阻止其自动启动），请将 `STARTMODE` 设置为 `manual`。

② 设置静态 IP 地址（此处对于 IPv4 指定 `192.168.1.1`，对于 IPv6 指定 `fd00:deca:fbad:50::1`）。

如果网络协作设备应该使用动态 IP 地址，请设置 `BOOTPROTO="dhcp"` 并去除（或注释掉）包含 `IPADDRESS` 和 `IPADDR6` 的行。

③ 将 `TEAM_RUNNER` 设置为 `loadbalance` 以激活负载平衡模式。

- ④ 指定应聚合以创建网络协作设备的一个或多个设备。
- ⑤ 定义链路监视器，以监视从属设备的状态。只有当设备已启动并可访问时，ethtool 才会执行检查。因此，检查速度将会足够快。但是，它不会检查设备实际上是否可以发送或接收包。  
如果您需要确保连接的可信度更高，请使用 arp\_ping 选项。这样会向任意主机发送 ping（在 TEAM\_LW\_ARP\_PING\_TARGET\_HOST 变量中配置）。仅当接收到应答时，才将网络协作设备视为正在运行。
- ⑥ 定义链路启动（或关闭）与运行程序收到通知之间的延迟（以毫秒为单位）。

## 23.8.2 用例：使用网络协作实现故障转移

故障转移用于确保关键网络协作设备的高可用性，方法是纳入并行的备用网络设备。备用网络设备时刻都在运行，并在主设备出现故障时接替主设备。

使用以下配置文件可创建具有故障转移功能的网络协作设备。继续[过程 23.1 “一般过程”](#)以设置设备。使用 teamdctl 检查输出。

### 例 23.13：DHCP 网络协作设备的配置

```
STARTMODE=auto ①
BOOTPROTO=static ②
IPADDR="192.168.1.2/24" ②
IPADDR6="fd00:deca:fbad:50::2/64" ②

TEAM_RUNNER=activebackup ③
TEAM_PORT_DEVICE_0="eth0" ④
TEAM_PORT_DEVICE_1="eth1" ④

TEAM_LW_NAME=ethtool ⑤
TEAM_LW_ETHTOOL_DELAY_UP="10" ⑥
TEAM_LW_ETHTOOL_DELAY_DOWN="10" ⑥
```

- ① 控制协作设备的启动。auto 一值表示接口将在网络服务可用时设置，并且会在每次重引导时自动启动。  
如果您需要自行控制设备（并阻止其自动启动），请将 STARTMODE 设置为 manual。

- ② 设置静态 IP 地址（此处对于 IPv4 指定 192.168.1.2，对于 IPv6 指定 fd00:deca:fbad:50::2）。  
如果网络协作设备应该使用动态 IP 地址，请设置 BOOTPROTO="dhcp" 并去除（或注释掉）包含 IPADDRESS 和 IPADDR6 的行。
- ③ 将 TEAM\_RUNNER 设置为 activebackup 以激活故障转移模式。
- ④ 指定应聚合以创建网络协作设备的一个或多个设备。
- ⑤ 定义链路监视器，以监视从属设备的状态。只有当设备已启动并可访问时，ethtool 才会执行检查。因此，检查速度将会足够快。但是，它不会检查设备实际上是否可以发送或接收包。  
如果您需要确保连接的可信度更高，请使用 arp\_ping 选项。这样会向任意主机发送 ping（在 TEAM\_LW\_ARP\_PING\_TARGET\_HOST 变量中配置）。仅当收到答复时，网络协作设备才会被视为已启动。
- ⑥ 定义链路启动（或关闭）与运行程序收到通知之间的延迟（以毫秒为单位）。

### 23.8.3 用例：组合设备上的 VLAN

VLAN 是**虚拟局域网 (Virtual Local Area Network)** 的缩写。它允许在单个物理以太网上运行多个**逻辑 (虚拟)** 以太网。它以逻辑方式将网络分为不同的广播域，以便数据包仅在为同一 VLAN 指定的端口之间交换。

下面的用例会在组合设备的基础上创建两个静态 VLAN：

- vlan0，绑定到 IP 地址 192.168.10.1
- vlan1，绑定到 IP 地址 192.168.20.1

按如下所示继续：

1. 在交换机上启用 VLAN 标记。要针对组合设备使用负载平衡，交换机需要支持**链接聚合控制协议 (LACP) (802.3ad)**。有关细节，请查阅硬件手册。
2. 确定是否要针对组合设备使用负载平衡或故障转移。按**第 23.8.1 节 “用例：网络协作的负载平衡”** 或**第 23.8.2 节 “用例：使用网络协作实现故障转移”** 中所述设置组合设备。
3. 在 /etc/sysconfig/network 中，创建包含以下内容的 ifcfg-vlan0 文件：

```
STARTMODE="auto"
BOOTPROTO="static" ①
IPADDR='192.168.10.1/24' ②
ETHERDEVICE="team0" ③
VLAN_ID="0" ④
VLAN='yes'
```

- ① 定义固定的 IP 地址（在 IPADDR 中指定）。
- ② 定义 IP 地址，这里包含其网络掩码。
- ③ 包含要用于 VLAN 接口的实际接口，这里是我们的组合设备 (team0)。
- ④ 为 VLAN 指定唯一的 ID。文件名和 VLAN\_ID 最好与名称 ifcfg-vlanVLAN\_ID 对应。在本例中，VLAN\_ID 为 0，因而文件名为 ifcfg-vlan0。

4. 将 /etc/sysconfig/network/ifcfg-vlan0 文件复制到 /etc/sysconfig/network/ifcfg-vlan1，并更改以下值：

- 将 IPADDR 从 192.168.10.1/24 更改为 192.168.20.1/24。
- 将 VLAN\_ID 从 0 更改为 1。

5. 启动两个 VLAN：

```
# wicked ifup vlan0 vlan1
```

6. 检查 ifconfig 的输出：

```
# ifconfig -a
[...]
vlan0      Link encap:Ethernet  HWaddr 08:00:27:DC:43:98
           inet  addr:192.168.10.1  Bcast:192.168.10.255  Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fedc:4398/64  Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:0  errors:0  dropped:0  overruns:0  frame:0
           TX packets:12  errors:0  dropped:0  overruns:0  carrier:0
           collisions:0  txqueuelen:1000
           RX bytes:0 (0.0 b)  TX bytes:816 (816.0 b)
```

```
vlan1      Link encap:Ethernet  HWaddr 08:00:27:DC:43:98
           inet  addr:192.168.20.1  Bcast:192.168.20.255  Mask:255.255.255.0
           inet6 addr: fe80::a00:27ff:fedc:4398/64  Scope:Link
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                  RX packets:0  errors:0  dropped:0  overruns:0  frame:0
                  TX packets:12  errors:0  dropped:0  overruns:0  carrier:0
                  collisions:0  txqueuelen:1000
                  RX bytes:0 (0.0 b)  TX bytes:816 (816.0 b)
```

# 24 打印机操作

SUSE® Linux Enterprise Desktop 支持使用多种类型的打印机进行打印，其中包括远程网络打印机。可以手动或使用 YaST 配置打印机。有关配置描述，请参见[第 34 章 “设置打印机”](#)。启动和管理打印任务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 24.8 节 “查错”](#)。

CUPS（通用 Unix 打印系统）是 SUSE Linux Enterprise Desktop 中的标准打印系统。

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，确保打印机配有支持的接口（USB、以太网或 Wi-Fi）并使用适合的打印机语言。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任务。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。

PDF 正在逐渐取代 PostScript，成为标准打印作业格式。可直接打印 PDF（而不仅仅是 PostScript）的 PostScript+PDF 打印机已经面世。传统的 PostScript 打印机需要在打印工作流程中将 PDF 转换为 PostScript。

## 标准打印机（PCL 和 ESC/P 等语言）

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。此处理阶段称为解释。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，可以生成符合需要的打印效果。Linux 可能无法提供某些特殊的打印机功能。除了 HP 和 Epson 之外，当前尚没有其他打印机制造商开发 Linux 驱动程序，并通过开放源代码许可证将这些驱动程序提供给 Linux 发行套件供应商。

## 专有打印机（也称为 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。通常只会为这些打印机提供 Windows 驱动程序。有关更多信息，请参见[第 24.8.1 节 “不支持标准打印机语言的打印机”](#)。

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.openprinting.org/printers>

包含打印机数据库的 OpenPrinting 主页。数据库显示最新的 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，当前评定为“完全支持”的打印机在最新的 SUSE Linux Enterprise Desktop 版本发布后，不一定还具有此称号。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。

<http://pages.cs.wisc.edu/~ghost/>

Ghostscript 网页

/usr/share/doc/packages/ghostscript/catalog.devices

内置 Ghostscript 驱动程序列表。

## 24.1 CUPS 工作流程

用户创建一个打印任务。打印作业由要打印的数据和有关假脱机程序的信息组成。其中包括打印机的名称或打印队列的名称，还有可能包括有关过滤器（例如特定于打印机的选项）的信息。

每台打印机至少有一个专用打印队列。假脱机程序存储着队列中的打印任务，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。

过滤器会将正在执行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）转换为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。

## 24.2 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 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。有关打印机连接的更多信息，请阅读 [https://en.opensuse.org/SDB:CUPS\\_in\\_a\\_Nutshell](https://en.opensuse.org/SDB:CUPS_in_a_Nutshell) 上的文章 **CUPS in a Nutshell** (CUPS 概述)。



### 警告：更改正在运行的系统中的电缆连接

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。

## 24.3 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 中的打印机选项是必需的。如果没有 PPD 文件，打印数据将会以“原始”状态转发到打印机，而这并不是用户所希望的。

要配置 PostScript 打印机，最佳的方法是获得一个合适的 PPD 文件。[manufacturer-PPDs](#) 和 [OpenPrintingPPDs-postscript](#) 软件包中提供了许多 PPD 文件。请参见[第 24.7.3 节“各个软件包中的 PPD 文件”](#) 和 [第 24.8.2 节“没有合适的 PPD 文件可用于 PostScript 打印机”](#)。

可以将新 PPD 文件存储在目录 `/usr/share/cups/model/` 中，或如[第 34.1.1 节“使用 YaST 添加驱动程序”](#) 中所述使用 YaST 将其添加到打印系统中。随后，可以在打印机设置过程中选择 PPD 文件。

如果打印机制造商希望您安装整个软件包，请务必小心。这种安装类型可能导致 SUSE Linux Enterprise Desktop 提供的支持失效。另外，打印命令可能会以不同的方式工作，并且系统可能不再能够对其他制造商的设备寻址。出于此原因，不建议安装制造商软件。

## 24.4 网络打印机

网络打印机可以支持多种协议，有些甚至同时支持不同的协议。尽管大部分支持的协议都已标准化，但有些制造商可能会修改标准。他们仅提供适用于少数操作系统的驱动程序。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，您可能需要试验几个选项，以找出起作用的配置。

CUPS 支持 socket、LPD、IPP 和 smb 协议。

### **socket**

**套接字**是指将纯文本打印数据直接发送到 TCP 套接字的连接。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://IP.OF.THE.PRINTER:PORT，例如：socket://192.168.2.202:9100/。

### **LPD（行式打印机守护程序）**

LPD 协议如 RFC 1179 中所述。使用此协议，打印队列 ID 等作业相关数据将先于实际打印数据发送。因此，配置 LPD 协议时必须指定打印队列。各打印机制造商的实施非常灵活，可以接受为打印队列指定任何名称。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。LPD 服务的端口号是 515。设备 URI 示例：lpd://192.168.2.202/LPT1。

### **IPP（互联网打印协议）**

IPP 基于 HTTP 协议。使用 IPP，所传送的与任务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。要正确配置 IPP，必须提供打印队列的名称。IPP 的端口号是 631。设备 URI 示例：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 使用的端口号为 137、138 和 139。设备 URI 示例：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（随 nmap 软件包提供）来确定协议。nmap 检查主机端口是否打开。例如：

```
> nmap -p 35,137-139,515,631,9100-10000 IP.OF.THE.PRINTER
```

## 24.5 使用命令行工具配置 CUPS

CUPS 可使用 [lpinfo](#)、[lpadmin](#) 和 [lpoptions](#) 之类的命令行工具配置。您需要一个包含一个后端（例如 USB）和多个参数的设备 URI。要确定系统上的有效设备 URI，请使用命令 [lpinfo -v | grep "://" ":"](#)：

```
> sudo lpinfo -v | grep "://" 
direct usb://ACME/FunPrinter%20XL
network socket://192.168.2.253
```

使用 [lpadmin](#)，CUPS 服务器管理员可以添加、删除或管理打印队列。要添加打印队列，请使用以下语法：

```
> sudo lpadmin -p QUEUE -v DEVICE-URI -P PPD-FILE -E
```

设备 (-v) 便会以 QUEUE (-p) 的形式供您使用，并使用指定的 PPD 文件 (-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。

请勿使用 -E 作为第一个选项。对于所有 CUPS 命令，第一个参数 -E 都会设置使用加密连接。要启用打印机，必须使用 -E，如下面的示例所示：

```
> sudo lpadmin -p ps -v usb://ACME/FunPrinter%20XL -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

以下示例配置了网络打印机：

```
> sudo lpadmin -p ps -v socket://192.168.2.202:9100/ -P \
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

有关 [lpadmin](#) 的更多选项，请参考 [lpadmin\(8\)](#) 的手册页。

在系统安装期间，某些选项被设置为默认值。可以为每个打印任务修改这些选项（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：

```
> sudo lpoptions -p QUEUE -l
```

示例：

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

激活的默认选项前面会加上星号 (\*)。

2. 使用 **lpadmin** 更改选项：

```
> sudo lpadmin -p QUEUE -o Resolution=600dpi
```

3. 检查新设置：

```
> sudo lpoptions -p QUEUE -l
```

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

普通用户运行 **lpoptions** 时，设置会写入 `~/.cups/lpoptions`。但是，`root` 设置会写入 `/etc/cups/lpoptions`。

## 24.6 从命令行打印

要从命令行打印，请输入 `lp -d QUEUENAME FILENAME`，并用相应的名称替换 `QUEUENAME` 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 QUEUENAME`。

## 24.7 SUSE Linux Enterprise Desktop 中的特殊功能

某些 CUPS 功能已针对 SUSE Linux Enterprise Desktop 进行调整。本节将介绍最重要的更改。

### 24.7.1 CUPS 和防火墙

完成 SUSE Linux Enterprise Desktop 的默认安装后, `firewalld` 将处于活动状态, 并且网络接口配置为位于 `public` 区域中, 这会阻止传入通讯。

当 `firewalld` 处于活动状态时, 您可能需要将 `firewalld` 配置为允许客户端浏览网络打印机, 方法是允许 `mdns` 和 `ipp` 通过内部网络区域。公共区域应该永不公开打印机队列。

(《安全和强化指南》, 第 23 章 “掩蔽和防火墙”, 第 23.4 节 “`firewalld`” 中和 [https://en.opensuse.org/SDB:CUPS\\_and\\_SANE\\_Firewall\\_settings](https://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings) 上提供了有关 `firewalld` 配置的详细信息。)

#### 24.7.1.1 CUPS 客户端

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处于 `Internal Zone` 中, 以便可以从网络内部访问工作站。

#### 24.7.1.2 CUPS 服务器

如果 CUPS 服务器位于受防火墙保护的可信网络环境中, 则应将网络接口配置为处于防火墙的 `Internal Zone` 中。建议不要在不可信网络环境中安装 CUPS 服务器, 除非您确定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。

### 24.7.2 浏览网络打印机

CUPS 服务器会定期在网络上公告共享打印机的可用性及状态信息。客户端可以访问此信息, 以便在打印对话框中显示可用打印机列表。此过程称为 “浏览”。

CUPS 服务器通过传统的 CUPS 浏览协议或 Bonjour/DNS-SD 在网络上通告它们的打印队列。为了能够浏览网络打印队列，需要在通过 CUPS 服务器打印的所有客户端上运行 `cups-browsed` 服务。默认情况下，`cups-browsed` 不会启动。要为活动会话启动该服务，请使用 `sudo systemctl start cups-browsed`。要确保它在系统引导后会自动启动，请在所有客户端上使用 `sudo systemctl enable cups-browsed` 启用该服务。

如果在启动 `cups-browsed` 之后浏览不起作用，则表明 CUPS 服务器是通过 Bonjour/DNS-SD 通告网络打印队列的。在此情况下，您需要另外安装 `avahi` 软件包，并在所有客户端上使用 `sudo systemctl start avahi-daemon` 启动关联的服务。

有关如何通过 `firewalld` 允许浏览打印机的信息，请参见第 24.7.1 节 “CUPS 和防火墙”。

### 24.7.3 各个软件包中的 PPD 文件

YaST 打印机配置使用 `/usr/share/cups/model` 中安装的 PPD 文件为 CUPS 设置队列。为查找适用于打印机型号的 PPD 文件，YaST 将对照硬件检测过程中确定的供应商和型号比较所有 PPD 文件中的供应商和型号。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。

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。例如，如果您有 PostScript 打印机，可直接将 PPD 文件复制到 `/usr/share/cups/model`（如果这些文件尚不存在于 `manufacturer-PPDs` 或 `OpenPrintingPPDs-postscript` 软件包中），以实现打印机的最佳配置。

其他 PPD 文件由下列软件包提供：

- `gutenprint`: Gutenprint 驱动程序及其匹配的 PPD
- `splix`: SpliX 驱动程序及其匹配的 PPD
- `OpenPrintingPPDs-ghostscript`: Ghostscript 内置驱动程序的 PPD
- `OpenPrintingPPDs-hpijs`: 适用于非 HP 打印机的 HPIJS 驱动程序的 PPD

## 24.8 查错

下面几节介绍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。

### 24.8.1 不支持标准打印机语言的打印机

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，所以这些打印机也称作 **GDI 打印机**。问题实际并不是出在编程接口上，而是因这些打印机只能通过相应打印机型号的专用打印机语言来寻址所造成。

某些 GDI 打印机可切换成以 GDI 方式或一种标准打印机语言进行操作。请参见打印机手册以了解这是否可行。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安装的打印系统，也不能保证它们适合多种硬件平台。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。

与其花时间尝试使专有 Linux 驱动程序运行，购买支持标准打印机语言（最好是 PostScript）的打印机可能更经济高效。这可以一次性彻底解决驱动程序问题，您不再需要安装并配置特殊驱动程序软件，以及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。

### 24.8.2 没有合适的 PPD 文件可用于 PostScript 打印机

如果 manufacturer-PPDs 或 OpenPrintingPPDs-postscript 软件包不包含适用于 PostScript 打印机的 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，或从打印机制造商网页下载合适的 PPD 文件。

如果 PPD 文件以 zip 存档 (.zip) 或自解压缩 zip 存档 (.exe) 的形式提供, 请使用 **unzip** 命令将其解压缩。首先, 查看 PPD 文件的许可证协议条款。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, 如果应用程序返回 “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。

## 24.8.3 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。

如果相应的 lpd 处于活动状态并且主机接受查询, 请以 root 身份运行以下命令, 以查询远程 HOST 上 QUEUE 的状态报告:

```
# 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, 响应应该显示为什么在 queue 的 host 上不能进行打印。如果您接收到类似例 24.1 “来自 lpd 的错误消息” 中的响应, 则问题是由于远程 lpd 引起的。

#### 例 24.1：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`

CUPS 网络服务器可以在 UDP 端口 631 上广播其队列，默认每 30 秒广播一次。因此，以下命令可用于测试网络中是否存在广播 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 24.2 “来自 CUPS 网络服务器的广播” 所示。

### 例 24.2：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。`lpstat -h HOST -l -t` 会返回 HOST 上所有队列的状态报告，前提是相应的 `cupsd` 处于活动状态并且主机接受查询。

下面的命令可用于测试 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。可能会退出一页空白纸。

```
> 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

有时，当在打印服务器计算机中运行的假脱机程序需要处理多个打印作业时会产生问题。这是打印服务器计算机中的假脱机程序导致的，目前尚无解决此问题的方法。变通方法是，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 24.4 节 “网络打印机”。

这样，打印服务器计算机仅充当各种不同数据传送方式之间 (TCP/IP 网络和本地打印机连接) 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。如果打印机连接到打印服务器计算机并且已启动，则通常可以在打开打印服务器计算机电源一段时间后，使用 `nmap` 软件包中的 `nmap` 实用程序确定此 TCP 端口。例如，`nmap IP-address` 可能会返回打印服务器计算机的以下输出：

| 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寻址。默认情况下，`nmap` 只检查在 `/usr/share/nmap/nmap-services` 中所列的几个常见端口。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。

#### 24.8.4 打印输出有缺陷但不显示错误消息

对于打印系统，打印任务完成的标志是 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特定于打印机的数据），则打印系统不会对此进行通知。如果打印机无法打印特定于打印机的数据，请选择一个更适合该打印机的 PPD 文件。

#### 24.8.5 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。后端用于确定在将数据传送报告为不可行前应执行的失败尝试次数。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` 命令重新启用打印。

## 24.8.6 CUPS 浏览：删除打印作业

如果 CUPS 网络服务器通过浏览向客户端主机广播其队列并且客户端主机上合适的本地 cupsd 处于活动状态，则客户端 cupsd 接受来自应用程序的打印任务并将它们转发到服务器上的 cupsd。当服务器上的 cupsd 接受打印任务后，会为该任务指派一个新的任务号。因此，客户端主机上的任务号与服务器上的任务号不同。因为打印作业通常都会立即转发出去，所以不能用客户端主机上的作业号将其删除，原因是当打印作业已转发到服务器 cupsd 后，客户端 cupsd 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 lpstat -h cups.example.com -o 之类的命令来确定服务器上的作业编号。此情况假设服务器尚未完成该打印作业（即尚未完全将它发送到打印机）。按如下方式使用获得的作业编号来删除服务器上的打印作业：

```
> cancel -h cups.example.com QUEUE-JOBNUMBER
```

## 24.8.7 有问题的打印作业和数据传输错误

如果在打印过程中关闭打印机或计算机，则打印任务将保留在队列中。再次打开计算机（或打印机）后，打印将继续。必须使用 cancel 从队列中删除有问题的打印任务。

如果打印作业损坏，或主机与打印机之间的通讯出现错误，打印机将无法正确处理数据，并会打印出很多张有乱码的纸。要修复该问题，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. 打印任务可能仍在队列中，因为只有在将任务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 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 删除打印作业。
3. 即使已将打印任务从队列中删除，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。
4. 通过关闭打印机一段时间完全重置打印机。然后插入纸张并打开打印机。

## 24.8.8 调试 CUPS

使用以下通用过程确定 CUPS 中的问题：

1. 在 /etc/cups/cupsd.conf 中设置 **LogLevel debug**。
2. 停止 cupsd。
3. 去除 /var/log/cups/error\_log\*，以免搜索非常长的日志文件。
4. 开始 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起因。

## 24.8.9 更多信息

有关在 SUSE Linux Enterprise Desktop 上执行打印操作的详细信息，请参见 openSUSE 支持数据库，网址为 <https://en.opensuse.org/Portal:Printing>。SUSE 知识库 (<https://www.suse.com/support/>) 中提供了对许多特定问题的解决方案。搜索 CUPS 文本找到相关文章。

## 25 图形用户界面

SUSE Linux Enterprise Desktop 包含 X.org 服务器、Wayland 和 GNOME 桌面。本章介绍用于所有用户的图形用户界面配置。

### 25.1 X Window 系统

X.org 服务器是实施 X11 协议的事实上的标准。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 或互联网）连接的另一个主机上。

在大多数情况下，X Window 系统不需要进行任何配置。X 启动期间会动态检测硬件。因此，`xorg.conf` 已被弃用。如果您仍然需要指定自定义选项来更改 X 的行为方式，还是可以通过修改 `/etc/X11/xorg.conf.d/` 下的配置文件来实现。

SUSE Linux Enterprise Desktop 15 SP5 中包含 Wayland，作为 X.org 服务器的替代选择。安装期间可选择安装它。

您可安装 `xorg-docs` 软件包以更深入地了解 X11。`man 5 xorg.conf` 提供了有关手动配置（如果需要）的格式的详细信息。有关 X11 开发的更多信息，请参见该项目的主页：<http://www.x.org>。

驱动程序位于 `xf86-video-*` 软件包中，例如 `xf86-video-ati`。相关手册页中详细说明了这些软件包附带的很多驱动程序。例如，如果您使用 `ati` 驱动程序，在 `man 4 ati` 中可以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-nvidiaG03` 的文档在安装软件包之后位于 `/usr/share/doc/packages/x11-video-nvidiaG04` 中。

在服务器上安装软件包 `xrdp`，并使用 RDP 客户端软件通过远程桌面协议访问服务器。

### 25.2 安装和配置字体

Linux 中的字体可分为两大类：

## 轮廓或矢量字体

包含作为字形组成相关绘图指导的数学描述。因此，每个字形都可以缩放为任意大小而无损质量。在可以使用此类字体（或字形）之前，需要将数学描述转换为光栅（网格）。此过程称为**字体光栅化**。**字体微调**（嵌入在字体中）可改进和优化特定大小的渲染效果。光栅化和微调通过FreeType库实现。

Linux下的常用格式为PostScript Type 1和Type 2、TrueType及OpenType。

## 位图或光栅字体

包含一个为特定字号设计的像素阵列。位图字体渲染速度极快，而且非常简单。然而，与矢量字体相比，位图字体无法在不损质量的情况下进行缩放。因此，这些字体通常以不同的大小发布。现在，Linux控制台中仍然使用位图字体，有时终端中也会使用这些字体。在Linux下，便携式编译格式(PCF)或字形位图分布格式(BDF)是最常用的格式。

这些字体的外观主要受两个方面的影响：

- 选择合适的字体系列，
- 采用某种算法渲染字体，达到接收者眼睛最舒服的效果。

最后一点只与矢量字体相关。虽然上面两点都需要根据个人情况而定，但仍有一些默认值需要创建。

Linux字体渲染系统由具有不同关系的几个库组成。基本字体渲染库是FreeType(<http://www.freetype.org/>)，它会将支持的格式的字体字形转换为优化的位图字形。渲染过程由算法及其参数（可能受专利问题影响）控制。

使用FreeType的每个程序或库都应该参考Fontconfig(<http://www.fontconfig.org/>)库。此库会从用户及系统那里收集字体配置。用户修改其Fontconfig设置后，此更改将导致发生Fontconfig感知的应用。

Arabic、Han或Phags-Pa等脚本所需的更复杂的OpenType成型以及其他更高级别的文本处理使用Harfbuzz(<http://www.harfbuzz.org/>)或Pango(<http://www.pango.org/>)进行。

### 25.2.1 显示安装的字体

要获得系统上安装了哪些字体的概观，请运行`rpm`或`fc-list`命令。这两个命令都可为您提供不错的答案，但有可能会因系统和用户配置不同而返回不同的列表。

## rpm

调用 rpm 可查看系统上安装了哪些包含字体的软件包：

```
> rpm -qa '*fonts*'
```

每个字体软件包都应该满足此表达式。不过，命令可能会返回误报，例如 fonts-config（它即不是字体，也不包含字体）。

## fc-list

调用 fc-list 可获得哪些字体系列可以访问、系统上或主目录中是否已安装这些字体的概观：

```
> fc-list ':' family
```



### 注意：命令 fc-list

命令 fc-list 是 Fontconfig 库的封装程序。从 Fontconfig（更确切地说，从它的缓存）可以查询许多有趣的信息。有关详细信息，请参见man 1 fc-list。

## 25.2.2 查看字体

如果想了解已安装字体系列的外观，请使用命令 ftview (ft2demos 软件包) 或访问 <http://fontinfo.opensuse.org/>。例如，要以 14 号字显示 FreeMono 字体，请按如下所示使用 ftview：

```
> ftview 14 /usr/share/fonts/truetype/FreeMono.ttf
```

如果需要更多信息，请访问 <http://fontinfo.opensuse.org/> 了解支持的样式（标准、粗体、斜体等）和语言。

## 25.2.3 查询字体

要查询指定了某种模式时使用哪种字体，请使用 fc-match 命令。

例如，如果您的模式包含已安装字体，**fc-match** 会返回文件名、字体系列和样式：

```
> fc-match 'Liberation Serif'  
LiberationSerif-Regular.ttf: "Liberation Serif" "Regular"
```

如果所需字体在您的系统上不存在，Fontconfig 的匹配规则将会生效，并尝试找到最接近的可用字体。这意味着用另一种字体来替代了您要求的字体。

```
> fc-match 'Foo Family'  
DejaVuSans.ttf: "DejaVu Sans" "Book"
```

Fontconfig 支持**别名**：用另一个系列名称替代某个名称。通用名称就是一个典型的例子，例如“sans-serif”、“serif” 和“monospace”。这些别名可由实际的系列名称，甚至是系列名称的首选设置列表替代：

```
> for font in serif sans mono; do fc-match "$font" ; done  
DejaVuSerif.ttf: "DejaVu Serif" "Book"  
DejaVuSans.ttf: "DejaVu Sans" "Book"  
DejaVuSansMono.ttf: "DejaVu Sans Mono" "Book"
```

结果可能因系统而异，具体视当前安装的字体而定。



### 注意：根据 fontconfig 而定的相似度规则

Fontconfig 总是根据具体请求返回最相似的实际系列（如果至少安装了一个系列）。“相似度”根据 Fontconfig 的内部度量以及用户或管理员的 Fontconfig 设置而定。

## 25.2.4 安装字体

安装新字体的方法主要有以下几种：

1. 将 `*.ttf` 或 `*.otf` 等字体文件手动安装到您知道的一个字体目录。如果要将字体用于整个系统，请使用标准目录 `/usr/share/fonts`。如果要安装在主目录中，请使用 `~/.config/fonts`。

如果不想使用标准目录，Fontconfig 允许您选择其他目录。使用 `<dir>` 元素告知 Fontconfig 所用目录，有关细节，请参见第 25.2.5.2 节 “深入了解 fontconfig XML”。

2. 使用 `zypper` 安装字体。大量字体已通过软件包提供，随附在 SUSE 分发包中或包含在 M17N:fonts (<http://download.opensuse.org/repositories/M17N:/fonts/>) 储存库中。请使用以下命令将储存库添加到您的列表中。例如，要为 SUSE Linux Enterprise Desktop 15 SP5 添加储存库，请使用以下命令：

```
> sudo zypper ar  
https://download.opensuse.org/repositories/M17N:/fonts/SLE_15/
```

要搜索您的 `FONT_FAMILY_NAME`，请使用以下命令：

```
> zypper se 'FONT_FAMILY_NAME*fonts'
```

## 25.2.5 配置字体的外观

根据渲染媒体和字号的不同，结果可能并不令人满意。例如，现在的显示器分辨率一般为 100dpi，这导致像素太大，字形显得粗陋难看。

有几种算法可用来处理低分辨率，例如消除锯齿（灰度平滑）、微调（适合网格）或子像素渲染（在一个方向将分辨率增至三倍）。这些算法还可能因字体格式而异。

通过 Fontconfig，可单独为每种字体选择渲染算法，也可为一组字体选择渲染算法。

### 25.2.5.1 通过 `sysconfig` 配置字体

SUSE Linux Enterprise Desktop 在 Fontconfig 上提供了一个 `sysconfig` 层。要尝试进行字体配置，这是一个不错的着手点。要更改默认设置，请编辑配置文件 `/etc/sysconfig/fonts-config`。（或使用 YaST sysconfig 模块）。编辑该文件之后，请运行 `fonts-config`。

```
> sudo /usr/sbin/fonts-config
```

重启该应用程序以显示成效。请牢记以下要点：

- 一些应用程序不需要重启。例如，Firefox 会不时重新读取 Fontconfig 配置。新创建或重新装载的标签可在稍后获得新的字体配置。
- 每次发生包安装或去除操作之后，系统会自动调用 **`fonts-config`** 脚本（如未调用，则表示字体软件包有错误）。
- 使用 **`fonts-config`** 命令行选项可以暂时覆盖每个 sysconfig 变量。有关详细信息，请参见 **`fonts-config --help`**。

有几个 sysconfig 变量可以更改。请参见 **`man 1 fonts-config`** 或 YaST sysconfig 模块的帮助页。以下是一些变量示例：

### 渲染算法的使用

可以使用

FORCE\_HINTSTYLE、FORCE\_AUTOHINT、FORCE\_BW、FORCE\_BW\_MONOSPACE、USE\_EMBEDDED\_BITMAPS 和 EMBEDDED\_BITMAP\_LANGAGES

### 通用别名的首选项列表

使用

PREFER\_SANS\_FAMILIES、PREFER\_SERIF\_FAMILIES、PREFER\_MONO\_FAMILIES 和 SEARCH\_METRIC\_COMPATIBLE

下面的列表提供了一些配置示例，按从“最清晰”字体（对比度较高）到“最漂亮”（较平滑）的顺序显示。

### 位图字体

通过 PREFER\_\*\_FAMILIES 变量设置首选位图字体。请按照帮助部分的示例使用这些变量。请注意，这些字体渲染为黑白色，不进行平滑处理，并且位图字体只有几种字号。考虑使用

`SEARCH_METRIC_COMPATIBLE="no"`

来禁用基于度量兼容性的系列名称替代。

### 渲染为黑白色的可缩放字体

渲染时未消除锯齿的可缩放字体与位图字体的显示效果相似，同时又可保持字体可缩放性。请使用经过精细微调的字体，如 Liberation 系列。遗憾的是，系统中经过精细微调的字体并不充足。设置下面的变量可强制采用此方法：

```
FORCE_BW="yes"
```

## 渲染为黑白色的等宽字体

仅采用不消除锯齿的方式渲染等宽字体，否则，请使用默认设置：

```
FORCE_BW_MONOSPACE="yes"
```

## 默认设置

渲染所有字体时都消除锯齿。经过精细微调的字体将通过**字节码解释器 (BCI)** 渲染，其余字体将使用自动微调器 (hintstyle=hintslight) 渲染。让所有相关的 sysconfig 变量保持默认设置。

## CFF 字体

以 CFF 格式使用字体。FreeType2 中有了当前的改进后，该字体应该也会比默认的 TrueType 字体更清晰。请按照 PREFER\_\*\_FAMILIES 示例尝试一下。可以使用以下选项将字体调得更黑更粗：

```
SEARCH_METRIC_COMPATIBLE="no"
```

因为它们默认是通过 hintstyle=hintslight 来渲染的。还可以考虑使用：

```
SEARCH_METRIC_COMPATIBLE="no"
```

## 专用自动微调器

即使对于精细微调的字体，也可以使用 FreeType2 的自动微调器。这可能会导致字形变得更粗，有时还会变得更模糊、对比度更低。设置下面的变量可激活此功能：

```
FORCE_AUTOHINTER="yes"
```

使用 FORCE\_HINTSTYLE 可控制微调级别。

## 25.2.5.2 深入了解 fontconfig XML

Fontconfig 的配置格式是**可扩展标记语言 (XML)**。下面的几个示例不是完整参考，只是简要概述。细节及其他启示可在 man 5 fonts-conf 或 /etc/fonts/conf.d/ 中找到。

中心 Fontconfig 配置文件是 /etc/fonts/fonts.conf，它及其他作品包括整个 /etc/fonts/conf.d/ 目录。要自定义 Fontconfig，可在两个位置插入您的更改：

## FONTCONFIG 配置文件

**1. 系统范围的更改：** 编辑文件 /etc/fonts/local.conf（默认情况下，该文件包含一个空的 fontconfig 元素）。

**2. 用户特定的更改：** 编辑文件 ~/.config/fontconfig/fonts.conf。将 Fontconfig 配置文件置于 ~/.config/fontconfig/conf.d/ 目录中。

用户特定的更改会覆盖任何系统范围的设置。



### 注意：弃用的用户配置文件

文件 ~/.fonts.conf 标记为已弃用，不应再使用。请改用 ~/.config/fontconfig/fonts.conf。

每个配置文件都需要有一个 fontconfig 元素。因此，最精简的文件应如下所示：

```
<?xml version="1.0"?>
  <!DOCTYPE fontconfig SYSTEM "fonts.dtd">
  <fontconfig>
    <!-- Insert your changes here -->
  </fontconfig>
```

如果默认目录不够用，请插入 dir 元素及相应的目录：

```
<dir>/usr/share/fonts2</dir>
```

Fontconfig 会以递归方式搜索字体。

使用下面的 Fontconfig 代码段可以选择字体渲染算法（请参见 [例 25.1 “指定渲染算法”](#)）：

#### 例 25.1：指定渲染算法

```
<match target="font">
  <test name="family">
    <string>FAMILY_NAME</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>true</bool>
  </edit>
```

```
<edit name="hinting" mode="assign">
  <bool>true</bool>
</edit>
<edit name="autohint" mode="assign">
  <bool>false</bool>
</edit>
<edit name="hintstyle" mode="assign">
  <const>hintfull</const>
</edit>
</match>
```

字体的各种属性都可以测试。例如，<test>元素可测试字体系列（如该例中所示）、字体间隔、间距、字体格式以及其他。如果完全不使用<test>，所有<edit>元素都会应用于每个字体（全局更改）。

#### 例 25.2：别名和系列名称替换项

##### 规则 1

```
<alias>
  <family>Alegreya SC</family>
  <default>
    <family>serif</family>
  </default>
</alias>
```

##### 规则 2

```
<alias>
  <family>serif</family>
  <prefer>
    <family>Droid Serif</family>
  </prefer>
</alias>
```

##### 规则 3

```
<alias>
  <family>serif</family>
```

```
<accept>
  <family>STIXGeneral</family>
</accept>
</alias>
```

例 25.2 “别名和系列名称替换项” 中的规则会创建一份排定了优先级的系列列表 (PFL)。根据元素的不同，执行的操作也不同：

#### 规则 1 中的 <default>

此规则会在 PFL 末尾添加一个 serif 系列名称。

#### 规则 2 中的 <prefer>

只要 PFL 中存在 Alegreya SC，此规则就会在 PFL 中的第一个 serif 之前添加 “Droid Serif”。

#### 规则 3 中的 <accept>

此规则会在 PFL 中第一个 serif 系列名称之后添加 “STIXGeneral” 系列名称。

如果将这些代码段按规则 1 - 规则 2 - 规则 3 的顺序组合起来，当用户请求 “Alegreya SC” 时，系统便会如创建如表 25.1 “基于 fontconfig 规则生成 PFL” 中所述的 PFL。

表 25.1：基于 FONTCONFIG 规则生成 PFL

| 顺序   | 当前的 PFL   |
|------|---|
| 请求   | <u>Alegreya SC</u>  |
| 规则 1 | <u>Alegreya SC</u> , <u>serif</u>   |
| 规则 2 | <u>Alegreya SC</u> , <u>Droid Serif</u> , <u>serif</u>                      |
| 规则 3 | <u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u> 、 <u>STIXGeneral</u> |

在 Fontconfig 的度量中，系列名称具有最高优先级，高于样式、大小等其他模式。Fontconfig 会检查系统上当前安装了哪个系列。如果安装了 “Alegreya SC”，Fontconfig 将会返回它。如果未安装，则会查找 “Droid Serif”，以此类推。

请小心。如果 Fontconfig 代码段的顺序发生变化，Fontconfig 可能会返回不同的结果，如表 25.2 “基于更改了顺序的 fontconfig 规则生成 PFL 的结果” 中所述。

表 25.2：基于更改了顺序的 FONTCONFIG 规则生成 PFL 的结果

| 顺序   | 当前的 PFL                   | 注释                                |
|------|---------------------------|-----------------------------------|
| 请求   | Alegreya SC               | 执行相同的请求。                          |
| 规则 2 | Alegreya SC               | PFL 中没有 <u>serif</u> , 未替代任何内容    |
| 规则 3 | Alegreya SC               | PFL 中没有 <u>serif</u> , 未替代任何内容    |
| 规则 1 | Alegreya SC, <u>serif</u> | PFL 中存在 <u>Alegreya SC</u> , 执行替代 |



## 注意：隐含意义

将 <default> 别名视为此组的分类或内含项（如果未安装）。如该例所示，<default> 应该一律优先于该组的 <prefer> 和 <accept> 别名。

<default> 分类不限于通用别名 serif、sans-serif 和 monospace。有关复杂示例，请参见 /usr/share/fontconfig/conf.avail/30-metric-aliases.conf。

例 25.3 “别名和系列名称替换项” 中的以下 Fontconfig 代码段会创建一个 serif 组。如果前一种字体未安装，此组中的每个系列可替代其他系列。

### 例 25.3：别名和系列名称替换项

```
<alias>
  <family>Alegreya SC</family>
<default>
  <family>serif</family>
</default>
</alias>
<alias>
  <family>Droid Serif</family>
<default>
  <family>serif</family>
```

```
</default>
</alias>
<alias>
  <family>STIXGeneral</family>
  <default>
    <family>serif</family>
  </default>
</alias>
<alias>
  <family>serif</family>
  <accept>
    <family>Droid Serif</family>
    <family>STIXGeneral</family>
    <family>Alegreya SC</family>
  </accept>
</alias>
```

优先级按 <accept> 别名中的顺序确定。类似地，可以使用更强的 <prefer> 别名。

例 25.4 “别名和系列名称替换项” 扩展了例 25.2 “别名和系列名称替换项”。

例 25.4：别名和系列名称替换项

#### 规则 4

```
<alias>
  <family>serif</family>
  <accept>
    <family>Liberation Serif</family>
  </accept>
</alias>
```

#### 规则 5

```
<alias>
  <family>serif</family>
  <prefer>
    <family>DejaVu Serif</family>
  </prefer>
```

```
</alias>
```

例 25.4 “别名和系列名称替换项” 中的扩展配置将导致下列 PFL 变化：

表 25.3：基于 FONTCONFIG 规则生成 PFL 的结果

| 顺序   | 当前的 PFL  |
|------|--|
| 请求   | <u>Alegreya SC</u>   |
| 规则 1 | <u>Alegreya SC</u> , <u>serif</u>  |
| 规则 2 | <u>Alegreya SC</u> , <u>Droid Serif</u> , <u>serif</u>   |
| 规则 3 | <u>Alegreya SC</u> 、 <u>Droid Serif</u> 、 <u>serif</u> 、 <u>STIXGeneral</u>  |
| 规则 4 | <u>Alegreya SC</u> , <u>Droid Serif</u> , <u>serif</u> , <u>Liberation Serif</u> ,<br><u>STIXGeneral</u>                       |
| 规则 5 | <u>Alegreya SC</u> , <u>Droid Serif</u> , <u>DejaVu Serif</u> , <u>serif</u> ,<br><u>Liberation Serif</u> , <u>STIXGeneral</u> |



## 注意：隐含意义。

- 如果同一个通用名称存在多个 <accept> 声明，则最后分析的声明“胜出”。如有可能，在创建系统范围的配置时，不要在用户 (/etc/fonts/conf.d/\*-user.conf) 之后使用 <accept>。
- 如果同一个通用名称存在多个 <prefer> 声明，则最后分析的声明“胜出”。如有可能，在系统范围的配置中，不要在用户之前使用 <prefer>。
- 对与相同的通用名称，每个 <prefer> 声明都会覆盖 <accept> 声明。如果管理员不仅希望用户可使用 <prefer>，还想允许其使用 <accept>，就不应该在系统范围的配置中使用 <prefer>。另一方面，因为用户通常都是使用 <prefer>，这种做法应该不会产生任何不利影响。我们还看到过在系统范围的配置中使用 <prefer> 的情况。

## 25.3 适用于管理员的 GNOME 配置

### 25.3.1 dconf 系统

GNOME 桌面的配置是使用 dconf 管理的。dconf 是按层次结构化的数据库或注册表，可让用户修改其个人设置，并可让系统管理员为所有用户设置默认值或强制值。dconf 取代了 GNOME 2 的 gconf 系统。

使用 dconf-editor 可通过图形用户界面查看 dconf 选项。使用 dconf 可通过命令行访问和修改配置选项。

GNOME Tweaks 工具提供易于使用的用户界面，其中包含不属于标准 GNOME 配置的其他配置选项。从 GNOME 应用程序菜单或在命令行中使用 gnome-tweak-tool 可启动该工具。

### 25.3.2 系统范围的配置

全局 dconf 配置参数可在 /etc/dconf/db/ 目录中设置，其中包括 GDM 的配置或针对用户锁定特定的配置选项。

我们以下面的过程举例说明如何创建系统范围的配置：

1. 在 /etc/dconf/db/ 中创建以 .d 结尾的新目录。此目录可包含任意数量的配置选项文本文件。在此例中，创建包含以下内容的文件 /etc/dconf/db/network.d/00-proxy：

```
# This is a comment
[system/proxy/http]
host='10.0.0.1'
enabled=true
```

2. 将新的配置指令分析为 dconf 数据库格式：

```
> sudo dconf update
```

3. 创建 /etc/dconf/profiles/user 文件，以将新 network 配置数据库添加至默认用户配置文件。然后添加以下内容：

system-db:network

文件 `/etc/dconf/profiles/user` 是 GNOME 的默认配置文件。其他配置文件可在环境变量 `DCONF_PROFILE` 中定义。

4. 可选：要锁定用户的代理配置，请创建文件 `/etc/dconf/db/network/locks/proxy`。然后将含有不能更改的关键字的一行添加至此文件：

```
/system/proxy/http/host  
/system/proxy/http/enabled
```

您可以使用图形 `dconf-editor` 创建包含一个用户的配置文件，然后使用 `dconf dump` / 列出所有配置选项。之后便可将配置选项存储在全局配置文件中。

<https://wiki.gnome.org/Projects/dconf/SystemAdministrators> 中提供了全局配置的详细说明。

### 25.3.3 在 Wayland 上管理平铺

如果您在 Wayland 上运行 GNOME 桌面，想让窗口并排排列，建议使用 `gTile` GNOME 扩展。`gTile` 可让您根据可配置的网格方案重新排列窗口。例如，您可以启动六个终端，将它们排成 3 行 2 列，占满整个桌面区域。

#### 25.3.3.1 安装 `gTile`

##### ! 重要

要使用 `gTile` GNOME 扩展，您需要运行 GNOME 桌面并安装 `chrome-gnome-shell`（适用于 SLED 15 SP4 及更低版本）或 `gnome-browser-connector`（适用于 SLED 15 SP5 及更高版本）软件包。

1. 使用浏览器访问 `gTile` 扩展网页 <https://extensions.gnome.org/extension/28/gtile/>。如果您看到有关缺少浏览器扩展的警告，请单击该链接，然后按照 Web 浏览器的说明安装并启用 GNOME Shell 扩展。

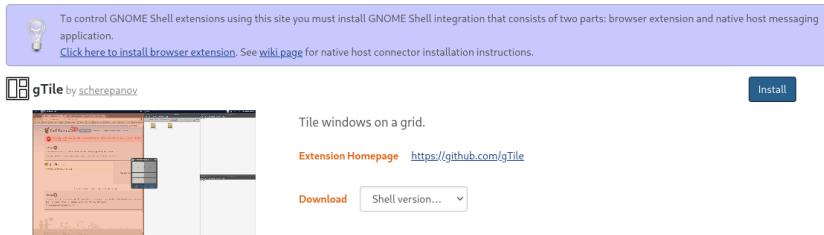


图 25.1：有关缺少 GNOME SHELL 扩展的警告

2. 在 Web 浏览器中安装 GNOME Shell 扩展后，您会在 gTile 网页右上角看到一个 OFF (关) 按钮。要激活 gTile 扩展，请单击该按钮将其切换为 ON (开) 。

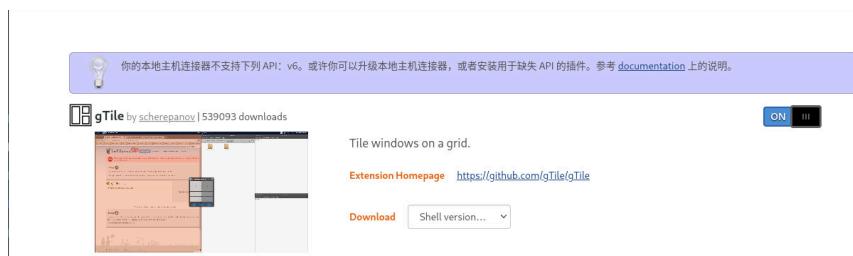


图 25.2：激活的 GNOME GTILE 扩展

3. GNOME 面板右侧的 GNOME 盘中会显示 gTile 托盘图标。您可以使用它来激活 gTile 窗口布局。



图 25.3：GTILE 托盘图标

### 25.3.3.2 使用 gTile 扩展

#### 过程 25.1：排列窗口

1. 要对当前聚焦的窗口打开 gTile，请按 **Meta** + **Enter** 。
2. 要设置将窗口排成几列，请按相应的数字键。

3. 要根据 Autotile Main window sizes (自动平铺主窗口大小) 自动平铺窗口, 请按 **M** 键。默认情况下, 您首次按 **M** 键时, 当前窗口会占据一半屏幕, 所有其他窗口会在另一半屏幕中平铺成一列。您可以重复按 **M** 键在各种变化间循环切换。这样可以始终让一个主窗口占据较大区域, 让几个次窗口占据较小区域。
4. 要将当前窗口移到屏幕的预定义平铺序列中, 请使用方向键。要减少或增加窗口所占的平铺块数, 请使用 **Shift** 和方向键。
5. 要将窗口拖到相邻窗口和桌面的边缘, 请按 **Ctrl + Alt + S**。
6. 要关闭 gTile 窗口, 请按 **Esc**。

#### 过程 25.2：高级配置

1. 要打开 gTile 设置, 请运行以下命令: **gnome-extensions-app**。
2. 选择 gTile, 然后单击 Settings (设置)。
3. 定义预设的组合键, 可轻松在常用窗口布局间切换。  
有关如何配置 gTile 扩展的更多细节, 请参见 <https://github.com/gTile/gTile>。

#### 25.3.4 更多信息

有关详细信息, 请参见 <http://help.gnome.org/admin/>。

### 25.4 使用 SUSE Prime 在 Intel 和 NVIDIA Optimus GPU 之间切换

SUSE Prime 是一款用于在板载 Intel 图形处理器 (GPU) 和配备 NVIDIA “可切换图形” Optimus 技术的 NVIDIA GPU 之间切换的工具。Optimus 提供了一种在板载 Intel GPU 和独立 NVIDIA GPU 之间轻松切换的机制。专为在省电模式下或以最佳性能运行笔记本电脑而设计：使用 Intel GPU 可省电, 使用 NVIDIA GPU 可支持 3D 应用程序。

SUSE Prime 随附在 SUSE Linux Enterprise 15 SP5 的 SUSE Linux Enterprise Workstation Extension 中。

SUSE Prime 只能在运行 X11 (而不是 Wayland) 的系统上工作。如果您的系统运行 Wayland, 您必须将其禁用, 并回退到 X11 才能使用 SUSE Prime (请参见 [第 25.4.1 节 “先决条件”](#) ) 。

## 25.4.1 先决条件

您必须有一个已配置好可正常工作的 NVIDIA Optimus GPU, 并且该 GPU 使用 SUSE Linux Enterprise 15 SP5 中包含的 NVIDIA 驱动程序 (请参见 [第 25.4.3 节 “安装 NVIDIA 驱动程序”](#) ), 以及一个板载 Intel GPU。切勿安装 NVIDIA Optimus 的较旧切换工具 Bumblebee。  
`/etc/X11/xorg.conf.d` 目录中不能存在 `/etc/X11/xorg.conf` 文件, 也不能存在包含活动 “ServerLayout” 、 “Device” 或 “Screen” 部分的配置文件。

SUSE Prime 仅适用于 X11。请使用 `loginctl` 命令查看您的系统是使用 X11 还是 Wayland:

```
> loginctl
  SESSION      UID  USER      SEAT      TTY
      2          1000  tux      seat0
> loginctl show-session 2|grep Type
Type=x11
```

如果您的系统使用 Wayland, 请编辑 `/etc/gdm/custom.conf` 并取消注释 `WaylandEnable=false` 将其禁用。然后重引导。

## 25.4.2 安装和使用 SUSE Prime

您的 NVIDIA 图形卡应已安装且正常工作。如果没有, 请参见 [第 25.4.3 节 “安装 NVIDIA 驱动程序”](#) 。

安装 `suse-prime` 软件包:

```
> sudo zypper install suse-prime
```

要切换 GPU, 请运行以下命令之一, 然后注销并重新登录:

```
> sudo prime-select intel
> sudo prime-select intel2
```

```
> sudo prime-select nvidia
```

如果是 modesetting 驱动程序, 请使用 intel 驱动程序。intel2 用于使用 xf86-video-intel 驱动程序的系统。您可以通过安装和运行 inxi:

```
> inxi -G
Graphics: Device-1: Intel Xeon E3-1200 v3/4th Gen Core Processor Integrated
          Graphics Controller
          Display Server: x11(X.org 1.20.1 ) drivers: modesetting (unloaded:
          fbdev, vesa)
          Resolution: 1920x1080@60.00hz
          OpenGL: renderer: Mesa DRI Intel Haswell Desktop version: 4.5 Mesa
          18.2.8
```

哪个 GPU 当前处于活动状态?

```
> sudo /usr/sbin/prime-select get-current
Driver configured: intel
```

### 25.4.3 安装 NVIDIA 驱动程序

如果您需要识别 NVIDIA 卡以便了解要使用哪个驱动程序, 请运行以下命令:

```
> /sbin/lspci | grep VGA
```

按照以下步骤使用 Zypper 安装驱动程序。

列出可用的驱动程序软件包:

```
> sudo zypper se nvidia
```

然后为您的 NVIDIA 图形卡安装驱动程序:

```
> sudo zypper se packagename
```

# 26 使用 FUSE 访问文件系统

FUSE 是用户空间中的文件系统 (file system in user space) 的缩写。这表示您可以将文件系统作为非特权用户配置和挂载。通常，只有 `root` 用户才能执行此任务。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。

## 26.1 配置 FUSE

需要安装 `fuse` 软件包才能使用 FUSE。根据要使用的文件系统，您需要作为独立软件包提供的附加插件。

一般而言，您无需配置 FUSE。但是建议创建一个合并所有挂载点的目录。例如，可以创建目录 `~/mounts` 并在其中为不同的文件系统插入子目录。

## 26.2 挂载 NTFS 分区

NTFS（新技术文件系统）是 Windows 的默认文件系统。在一般情况下，由于非特权用户无法使用外部 FUSE 库挂载 NTFS 块设备，因此下文所述的 Windows 分区挂载过程需要 `root` 特权。只有已安装 SUSE Linux Enterprise Workstation Extension 的 SUSE Linux Enterprise Server 和 SUSE Linux Enterprise Desktop 支持挂载 NTFS 分区。

1. 切换为 `root` 用户，然后安装软件包 `ntfs-3g`。SUSE Linux Enterprise Workstation Extension 中提供了该软件包。
2. 创建一个要充当挂载点的目录，例如 `~/mounts/windows`。
3. 确定所需的 Windows 分区。使用 YaST 并启动分区程序模块查看哪些分区属于 Windows，但不要修改任何内容。或者，切换为 `root` 用户并执行 `/sbin/fdisk -l`。查找分区类型为 `HPFS/NTFS` 的分区。
4. 以读写模式挂载分区。使用相应的 Windows 分区替换占位符 `DEVICE`：

```
> ntfs-3g /dev/DEVICE MOUNT POINT
```

要在只读模式下使用 Windows 分区，请追加 -o ro:

```
> ntfs-3g /dev/DEVICE MOUNT POINT -o ro
```

**ntfs-3g** 命令使用当前用户 (UID) 和组 (GID) 挂载给定设备。如果要为其他用户设置写权限，请使用命令 id USER 获取 UID 和 GID 值的输出。设置方式：

```
# id tux
uid=1000(tux) gid=100(users) groups=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE MOUNT POINT -o uid=1000,gid=100
```

在手册页中查找其他选项。

要卸载该资源，请运行 **fusermount -u MOUNT POINT**。

## 26.3 更多信息

有关详细信息，请参见 FUSE 主页：<https://github.com/libfuse/libfuse>。

## 27 安装多个内核版本

SUSE Linux Enterprise Desktop 支持多个内核版本的并行安装。安装第二个内核时，将自动创建一个引导项和一个 initrd，因此无需做进一步的手动配置。重引导计算机时，新添加的内核可用作附加的引导参数。

使用此功能，您可以安全测试内核更新，同时，还始终能够回退到已经过验证的前一内核。执行此操作时，请不要使用更新工具（例如 YaST 联机更新或更新小程序），而应该遵照本章中所述的过程。



### 警告：支持权利

请注意，如果您安装自行编译的内核或第三方内核，则会失去所有的支持权利。仅支持 SUSE Linux Enterprise Desktop 随附的内核，以及通过官方更新渠道为 SUSE Linux Enterprise Desktop 提供的内核。



### 提示：检查引导加载程序配置内核

建议在安装其他内核后检查您的引导加载程序配置，以设置您选择的默认引导项。有关更多信息，请参见[第 18.3 节 “使用 YaST 配置引导加载程序”](#)。

### 27.1 启用和配置多版本支持

在 SUSE Linux Enterprise Server 12 和更高版本中，系统默认支持安装软件包的多个版本（多版本支持）。要验证此设置，请按如下所示继续：

1. 以 `root` 身份使用所选的编辑器打开 `/etc/zypp/zypp.conf`。
2. 搜索字符串 `multiversion`。如果为所有支持此功能的内核软件包启用多版本，以下行可能会取消注释：

```
multiversion = provides:multiversion(kernel)
```

3. 要对特定的内核类型限制多版本支持，请将逗号分隔的软件包名称列表添加到 `/etc/zypp/zypp.conf` 中的 `multiversion` 选项，例如

```
multiversion = kernel-default,kernel-default-base,kernel-source
```

4. 保存所做的更改。



## 警告：内核模块软件包 (KMP)

确保也为更新后的新内核安装了由供应商提供的必需的内核模块（内核模块软件包）。内核更新过程不会发出有关最终缺少内核模块的警告，因为系统上保留的旧内核仍能满足软件包要求。

### 27.1.1 自动删除未使用的内核

如果您经常测试已启用多版本支持的新内核，引导菜单很快就会变得无序。由于 `/boot` 分区的空间通常有限，因此，您可能会遇到 `/boot` 溢出的问题。您可以使用 YaST 或 Zypper 手动删除未使用的内核版本（如下文所述），也可以将 `libzypp` 配置为自动删除不再使用的内核。默认情况下不会删除内核。

1. 以 `root` 身份使用所选的编辑器打开 `/etc/zypp/zypp.conf`。
2. 搜索字符串 `multiversion.kernels`，并通过取消注释对应的行来激活此选项。此选项将会使用包含以下值的逗号分隔列表：
  - `5.3.18-53.3`: 保留具有指定版本号的内核
  - `latest`: 保留具有最高版本号的内核
  - `latest-N`: 保留版本号排在第 N 位的内核
  - `running`: 保留正在运行的内核
  - `oldest`: 保留具有最低版本号的内核（最初 SUSE Linux Enterprise Desktop 随附的内核）
  - `oldest+N`: 保留版本号排在倒数第 N 位的内核

以下是一些示例

multiversion.kernels = latest,running

保留最新内核以及当前正在运行的内核。这相当于不启用多版本功能，不过，旧内核将会在下一次重引导后去除，而不是在安装后立即去除。

multiversion.kernels = latest,latest-1,running

保留最新的两个内核，以及当前正在运行的内核。

multiversion.kernels = latest,running,5.3.18-53.3

保留最新的内核、当前正在运行的内核，以及 5.3.18-53.3。



### 提示：保留正在运行的内核

除非您使用的是特殊的设置，否则始终将内核标记为 running。

如果不保留正在运行的内核，系统在更新内核时会将其删除。而这意味着正在运行的内核的所有模块也会被删除，并且无法再装载。

如果您决定不保留正在运行的内核，则在内核升级后始终应立即重引导，以免模块发生问题。

## 27.1.2 用例：仅在重引导后删除旧内核

您想要确保只在系统成功用新内核重引导之后才删除旧内核。

更改 /etc/zypp/zypp.conf 中的下面一行：

multiversion.kernels = latest,running

上面的参数告知系统要保留最新内核以及正在运行的内核（前提是两者不同）。

## 27.1.3 用例：保留较旧内核作为后备

您想要保留一个或多个内核版本，以便有一个或多个“备用的”内核。

如果您需要使用内核进行测试，则这样做可能十分有用。如果出现了故障（例如，您的计算机不能引导），您仍可使用一个或多个已知正常的内核版本。

更改 `/etc/zypp/zypp.conf` 中的下面一行：

```
multiversion.kernels = latest,latest-1,latest-2,running
```

安装新内核之后，当您重引导系统时，系统将保留三个内核：当前内核（配置为 `latest,running`）以及它的前两任（配置为 `latest-1` 和 `latest-2`）。

#### 27.1.4 用例：保留特定的内核版本

您要执行定期系统更新和安装新的内核版本。但是，您还要编译自己的内核版本，并想要确保系统保留这些版本。

更改 `/etc/zypp/zypp.conf` 中的下面一行：

```
multiversion.kernels = latest,5.3.18-53.3,running
```

在安装新内核后重引导系统时，系统将保留两个内核：新的且正在运行的内核（配置为 `latest,running`），以及您自己编译的内核（配置为 `5.3.18-53.3`）。

## 27.2 使用 YaST 安装/去除多个内核版本

可以使用 YaST 安装或去除多个内核：

1. 启动 YaST，然后通过软件 > 软件管理打开软件管理器。
2. 选择视图 > 软件包分类 > 多版本软件包，以列出能够提供多个版本的所有软件包。

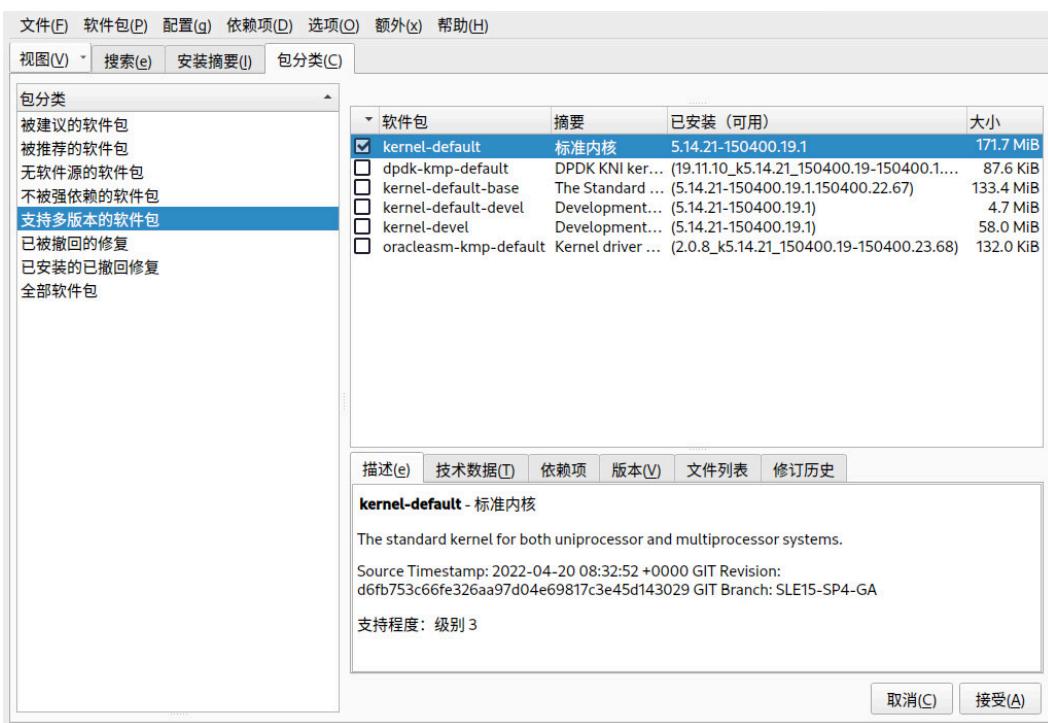


图 27.1：YAST 软件管理器：多版本视图

3. 在底部窗格的左侧选择一个软件包并打开其版本选项卡。
4. 要安装某个软件包，请单击它旁边的复选框。绿色选中标记表示已选择对应的软件包进行安装。  
要去除某个已安装的软件包（带有白色选中标记），请单击它旁边的复选框，直到出现红色的 X，这表示已选择去除该软件包。
5. 单击接受开始安装。

## 27.3 使用 Zypper 安装/去除多个内核版本

可以使用 zypper 安装或去除多个内核：

1. 使用命令 zypper se -s 'kernel\*' 显示所有可用内核软件包的列表：

| S | Name           | Type    | Version            | Arch   | Repository                       |
|---|----------------|---------|--------------------|--------|----------------------------------|
| - | -              | -       | -                  | -      | -                                |
| + | kernel-default | package | 5.14.21-150400.6.3 | x86_64 | SLE-Module-Basesystem15-SP4-Pool |

|                               |   |                      |
|-------------------------------|---|----------------------|
| kernel-default-base           | package   5.14.21-150400.6.3.150400.22.27 | x86_64   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| kernel-default-devel          | package   5.14.21-150400.6.3              | x86_64   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| kernel-devel                  | package   5.14.21-150400.6.4              | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-all       | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-amdgpu    | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-ath10k    | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-ath11k    | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-atheros   | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-bluetooth | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-bnx2      | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-brcm      | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-chelsio   | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-dpaa2     | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-i915      | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-intel     | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-iwlwifi   | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-liquidio  | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-marvell   | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-media     | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmwaremediatek   | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-mellanox  | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-mwififix  | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-network   | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-nfp       | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-nvidia    | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |
| i   kernel-firmware-platform  | package   20220119-150400.1.1             | noarch   SLE-Module- |
| Basesystem15-SP4-Pool         |   |                      |

|                                 |                               |                      |
|---------------------------------|-------------------------------|----------------------|
| i   kernel-firmware-prestera    | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-qcom        | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-qlogic      | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-radeon      | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-realtek     | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-serial      | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-sound       | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-ti          | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-ueagle      | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| i   kernel-firmware-usb-network | package   20220119-150400.1.1 | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |
| kernel-macros                   | package   5.14.21-150400.6.4  | noarch   SLE-Module- |
| Basesystem15-SP4-Pool           |                               |                      |

## 2. 安装时指定确切的版本：

```
> sudo zypper in kernel-default-5.3.18-53.3
```

## 3. 卸装内核时，使用命令 **zypper se -si 'kernel\*'** 可以列出安装的所有内核，使用 **zypper rm PACKAGE-NAME-VERSION** 可以去除指定软件包。

## 28 管理内核模块

虽然 Linux 属于单内核，但可通过内核模块加以扩展。这些特殊对象可以插入到内核中，并可按需去除。就实际角度而言，内核模块使添加和去除内核本身未包含的驱动程序和接口成为现实。Linux 提供了若干用于管理内核模块的命令。

### 28.1 使用 lsmod 和 modinfo 列出装载的模块

使用 **lsmod** 命令可查看目前装载了哪些内核模块。该命令的输出可能如下所示：

```
> lsmod
Module                  Size  Used by
snd_usb_audio           188416  2
snd_usbmidi_lib          36864  1 snd_usb_audio
hid_plantronics          16384  0
snd_rawmidi              36864  1 snd_usbmidi_lib
snd_seq_device            16384  1 snd_rawmidi
fuse                     106496  3
nfsv3                    45056  1
nfs_acl                  16384  1 nfsv3
```

输出内容分为三列：Module 列列出所装载模块的名称，Size 列显示各模块的大小。Used by 列显示引用模块的进程数及其名称。此列表可能不完整。

要查看有关特定内核模块的详细信息，请使用 **modinfo MODULE\_NAME** 命令。其中 MODULE\_NAME 为所需内核模块的名称。**modinfo** 二进制文件位于用户 PATH 环境变量中未包含的 /sbin 目录下。这意味着，当您以普通用户身份运行 **modinfo** 命令时，必须指定该二进制文件的完整路径：

```
> /sbin/modinfo kvm
filename:      /lib/modules/5.3.18-57-default/kernel/arch/x86/kvm/kvm.ko.xz
license:       GPL
author:        Qumranet
suserrelease:  SLE15-SP3
srcversion:    3D8FBA9060D4537359A06FC
depends:      irqbypass
```

```
supported: yes
retpoline: Y
intree: Y
name: kvm
vermagic: 5.3.18-57-default SMP mod_unload modversions
```

## 28.2 添加和去除内核模块

虽然可以使用 `insmod` 和 `rmmod` 分别添加和去除内核模块，但建议使用 `modprobe` 工具来执行这些操作。`modprobe` 具有多项重要优势，包括自动解析依赖项和将内核模块列入黑名单。如果不指定任何参数，使用 `modprobe` 命令会安装指定的内核模块。必须使用 `root` 特权来运行 `modprobe`：

```
> sudo modprobe acpi
```

要去除内核模块，请使用 `-r` 参数：

```
> sudo modprobe -r acpi
```

### 28.2.1 引导时自动装载内核模块

您可以选择不手动装载内核模块，而是使用 `systemd-modules-load.service` 服务在引导过程中自动装载这些模块。要启用内核模块，请将 `.conf` 文件添加到 `/etc/modules-load.d/` 目录。建议为配置文件指定与模块相同的名称，例如：

```
/etc/modules-load.d/rt2800usb.conf
```

配置文件中必须包含所需内核模块的名称（例如 `rt2800usb`）。

通过上述的这个技巧，无需指定任何参数即可装载内核模块。如果您需要使用特定选项装载内核模块，请将配置文件添加到 `/etc/modprobe.d/` 目录。该文件的扩展名必须为 `.conf`。文件名必须符合以下命名约定：`priority-modulename.conf`，例如：`50-thinkfan.conf`。配置文件中必须包含内核模块名称及所需参数。您可以使用以下示例命令来创建包含内核模块名称及其参数的配置文件：

```
> echo "options thinkpad_acpi fan_control=1" | sudo tee /etc/modprobe.d/  
thinkfan.conf
```



## 注意：装载内核模块

当检测到设备或用户空间请求特定功能时，系统会自动装载大多数内核模块。因此，很少需要手动将模块添加到 `/etc/modules-load.d/`。

### 28.2.2 使用 modprobe 将内核模块列入黑名单

将某个内核模块列入黑名单后，引导期间便不再会装载该模块。当要禁用您怀疑可能导致系统出现问题的某个模块时，此功能十分有用。您仍可使用 `insmod` 或 `modprobe` 工具手动装载列入黑名单的内核模块。

要将模块列入黑名单，请创建包含以下内容的 `/etc/modprobe.d/60-blacklist-MODULE_NAME.conf` 文件：

```
blacklist MODULE_NAME
```

以 root 身份运行 `dracut` 命令生成新的 `initrd` 映像，然后重引导计算机（将 `NAME` 替换为当前 `initrd` 的名称，将 `KERNELVERSION` 替换为当前运行的内核）：

```
> su  
echo "blacklist nouveau" >> /etc/modprobe.d/60-blacklist-nouveau.conf  
/usr/bin/dracut --logfile /var/log/YaST2/mkinitrd.log --force /boot/$initrd-NAME  
$KERNELVERSION  
reboot
```

如果只想临时禁用内核模块，可在引导期间即时将其列入黑名单。要实现此目的，请在引导屏幕显示时按 `E` 键。这样，您会进入一个可供您修改引导参数的小编辑器。找到如下所示的行：

```
linux /boot/vmlinuz...splash= silent quiet showopts
```

在该行末尾添加 `modprobe.blacklist=MODULE_NAME` 命令。例如：

```
linux /boot/vmlinuz...splash= silent quiet showopts modprobe.blacklist=nouveau
```

按 **F10** 或 **Ctrl + X** 以按照指定配置引导。

要通过 GRUB 将某个内核模块永久列入黑名单，请打开要编辑的 `/etc/default/grub` 文件，在 `GRUB_CMD_LINUX` 命令中添加 `modprobe.blacklist=MODULE_NAME` 选项。然后运行 `sudo grub2-mkconfig -o /boot/grub2/grub.cfg` 命令使更改生效。

# 29 使用 udev 进行动态内核设备管理

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（无论插入还是移除设备）需要传播给用户空间。插入及识别设备后需要对其进行配置。某个设备已识别状态的任何更改都需要通知给此设备的用户。`udev` 可提供所需的基础架构来动态维护 `/dev` 目录中的设备节点文件和符号链接。`udev` 规则提供了将外部工具插入内核设备事件处理的方式。因而，您可以通过添加在内核设备处理过程中执行的特定脚本，来自定义 `udev` 设备处理方式，或者可以在设备处理期间请求并导入其他数据进行评估。

## 29.1 `/dev` 目录

`/dev` 目录中的设备节点提供访问相应内核设备的途径。使用 `udev` 时，`/dev` 目录会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，所有文件都是在系统每次启动时出现。手动创建或修改的文件在重引导时是有意不保存的。使用 `systemd-tmpfiles` 可以创建无论相应内核设备的状态如何，都始终应位于 `/dev` 目录中的静态文件和目录。配置文件位于 `/usr/lib/tmpfiles.d/` 和 `/etc/tmpfiles.d/` 中。有关详细信息，请参见 `systemd-tmpfiles(8)` 手册页。

## 29.2 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，内核都会发送 `uevent` 来向 `udev` 通知更改。系统启动时，`udev` 守护程序会读取并分析 `/usr/lib/udev/rules.d/*.rules` 和 `/etc/udev/rules.d/*.rules` 文件中的所有规则，并将分析结果保留在内存中。如果更改、添加或去除了规则文件，守护程序可以使用 `udevadm control --reload` 命令重新装载这些规则在内存中的表示。有关 `udev` 规则及其语法的更多细节，请参见第 29.6 节 “使用 `udev` 规则影响内核设备事件处理”。

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键、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核心 uevents。

## 29.3 驱动程序、内核模块和设备

设备的内核总线驱动程序探测。对于每个检测到的设备，内核都会在驱动程序核心将 uevent 发送到 udev 守护程序时创建内部设备结构。总线设备通过特殊格式的 ID 来标识自己，该 ID 可表明其为何种设备。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 会读取 ID 列表，并在内核的 /lib/modules 目录中为当前可用的所有模块创建文件 modules.alias。使用这种基础架构，模块的装载就如为每个带有 modprobe 键的事件调用 MODALIAS 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。所有这些操作均由 udev 自动触发。

## 29.4 引导和启动设备设置

在 udev 守护程序运行之前的引导进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架构保存在 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 中所有 uevent 文件的简单循环将再次触发所有事件，以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。您无需手动搜索连接的设备，udev 会在根文件系统可用后向内核请求所有设备事件，这样 USB 鼠标设备的事件就会再次运行。现在它在挂载的 root 文件系统上找到内核模块，因此可以初始化 USB 鼠标。

在用户空间中，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。

## 29.5 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于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系统日志的默认系统日志优先级在 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级。

## 29.6 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 的任何信息匹配。规则还可以从外部程序请求其他信息。系统会将事件与目录 /usr/lib/udev/rules.d/（适用于默认规则）和 /etc/udev/rules.d（系统特定的配置）中提供的所有规则进行匹配。

规则文件中的每一行至少包含一个键值对。有两种类型的键，匹配键和赋值键。如果所有匹配键与它们的值匹配，则应用此规则并将赋值键指派给特定的值。匹配规则可以指定设备节点的名称、添加指向该节点的符号链接或者运行作为事件处理一部分的特定程序。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。udev 手册页中描述了有关规则语法和提供用来与数据匹配或导入数据的键的详细信息。以下示例规则提供了 udev 规则语法的基本介绍。这些示例规则全部取自 udev 默认规则集 /usr/lib/udev/rules.d/50-udev-default.rules。

### 例 29.1：udev 规则示例

```
# 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` 规则包含三个键：一个匹配键 (`KERNEL`) 和两个赋值键

(`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` 键为设备节点指派特殊权限，仅为此设备的拥有者指派读写权限。`OPTIONS` 键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，但该规则仍然值得考虑。该规则包含两个匹配键 (`KERNEL` 和 `ATTRS`) 和一个赋值键 (`SYMLINK`)。`KERNEL` 键会搜索 `ttyUSB` 类型的所有设备。如果使用 `*` 通配符，该键会匹配多个此类设备。第二个匹配键 `ATTRS` 会检查任何 `ttyUSB` 设备的 `sysfs` 中的 `product` 属性文件是否包含特定字符串。赋值键 (`SYMLINK`) 会触发为该设备添加名为 `/dev/pilot` 的符号链接。此键中使用的运算符 (`+=`) 告知 `udev` 额外执行此操作，即使前面或后面的规则添加了其他符号链接也是如此。由于此规则包含两个匹配键，因此仅当两个条件都满足时才会应用此规则。

`printer` 规则处理 USB 打印机，其中包含两个匹配键 (`SUBSYSTEM` 和 `KERNEL`)，并且必须同时应用这两个键，才能应用整个规则。三个赋值键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` 键中使用通配符 `*` 会使其匹配多个 `lp` 打印机设备。`NAME` 和 `SYMLINK` 键中都使用了替换项，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` 规则可使 `udev` 在运行期间通过外部助手脚本装载其他固件。`SUBSYSTEM` 匹配键会搜索 `firmware` 子系统。`ACTION` 键检查是否添加了属于 `firmware` 子系统的任何设备。`RUN+=` 键会触发 `firmware.sh` 脚本的执行，以查找应装载的固件。

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键值对构成。
- 键的运算由运算符确定。`udev` 规则支持多个运算符。
- 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。如果某个规则超过一行，请使用 `\` 合并不同的行，就像在外壳语法中一样。

- udev 规则支持与 `*`、`?` 和 `[]` 模式匹配的外壳风格模式。
- udev 规则支持替换项。

## 29.6.1 在 udev 规则中使用运算符

创建可以从多个运算符选择的键，具体取决于要创建的键的类型。匹配键通常用于查找匹配或明显不匹配搜索值的值。匹配键包含以下运算符之一：

`==`

比较等于性。如果键包含搜索模式，则匹配该模式的所有结果均有效。

`!=`

比较不等于性。如果键包含搜索模式，则匹配该模式的所有结果均有效。

赋值键可以使用下面的任何运算符：

`=`

为键指派值。如果键以前由一列值构成，键将重置，并且仅指派一个值。

`+=`

为包含一列项的键添加一个值。

`:=`

指派最终值。不允许后面的规则进行任何后续更改。

## 29.6.2 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`

元素 `DEVPATH`。

%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 (通过 ATTRIBUTE 指定)。

%E{VARIABLE}, \$env{VARIABLE}

环境变量的值 (通过 VARIABLE 指定)。

%c, \$result

PROGRAM 的输出。

%%

% 字符。

\$\$

\$ 字符。

### 29.6.3 使用 udev 匹配键

匹配键描述应用 udev 规则之前必须满足的条件。以下匹配键可用：

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，例如，DEVPATH=/bus/pci/drivers/ipw3945 用于搜索与 ipw3945 驱动程序相关的所有事件。

## 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，以查找与 sysfs 属性值匹配的设备。

## 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。RESULT 键可使用程序的输出（打印到 `stdout`）。

## RESULT

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

## 29.6.4 使用 udev 赋值键

与上述匹配键相比，赋值键未描述必须满足的条件。它们将值、名称和操作指派给由 udev 维护的设备节点。

### NAME

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 键的其他所有规则。

### 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，也将使用该键匹配 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，也将使用该键匹配环境变量。

### RUN

告知 udev 向程序列表添加要为该设备执行的程序。请注意，应将此类程序限制为短期执行的任务，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 跳过若干规则，继续执行 GOTO 键所引用标签对应的规则。

### IMPORT{TYPE}

将变量装载入外部程序输出之类的事件环境中。udev 可导入多种类型的变量。如果未指定任何类型，udev 将尝试根据文件许可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 从父设备导入存储的键。

## WAIT\_FOR\_SYSFS

告知 udev 等待为特定设备创建指定的 sysfs 文件。例如，WAIT\_FOR\_SYSFS="ioerr\_cnt" 会通知 udev 等待 ioerr\_cnt 文件成功创建。

## OPTIONS

OPTION 键可以有多个值：

- last\_rule 告知 udev 忽略后面的所有规则。
- ignore\_device 告知 udev 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。

## 29.7 永久设备命名

动态设备目录和 udev 规则基础架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 还维护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
```

## 29.8 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。udev 会使用这些信息在 /dev 中创建设备节点

### /dev/\*

动态创建的设备节点和使用 systemd-tmpfiles 创建的静态内容；有关详细信息，请参见 systemd-tmpfiles(8) 手册页。

以下文件和目录包含 udev 基础架构的关键元素：

### /etc/udev/udev.conf

udev 主配置文件。

### /etc/udev/rules.d/\*

系统专属的 udev 事件匹配规则。可在此处添加自定义规则，以修改或覆盖 /usr/lib/udev/rules.d/\* 中的默认规则。

文件将以字母数字顺序进行分析。文件中优先级较高的规则将会修改或覆盖优先级较低的规则。数字越小，优先级越高。

/usr/lib/udev/rules.d/\*

默认的 udev 事件匹配规则。此目录中的文件由软件包拥有，将在更新时重写。请勿在此处添加、去除或编辑文件，而是应使用 /etc/udev/rules.d。

/usr/lib/udev/\*

从 udev 规则调用的帮助程序。

/usr/lib/tmpfiles.d/ 和 /etc/tmpfiles.d/

存放静态 /dev 内容。

## 29.9 更多信息

有关 udev 基础架构的更多信息，请参见以下手册页：

udev

有关 udev、键、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。

udevd

有关 udev 事件管理守护程序的信息。

# 30 特殊系统功能

本章首先提供有关各种软件包、虚拟控制台和键盘布局的信息。我们将讨论 bash、cron 和 logrotate 等软件组件，因为它们在上一个发布周期中已进行了更改或增强。即使这些组件很小或者被认为不太重要，用户也应该更改它们的默认行为，因为这些组件往往与系统密切相关。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。

## 30.1 特殊软件包的相关信息

后续章节提供了有关以下工具的基本信息：bash、cron、logrotate、locate、ulimit 和 free。

### 30.1.1 bash 软件包和 /etc/profile

Bash 是默认的系统外壳。在用作登录外壳时，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。为了确保系统能正确处理这些文件，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。执行以下外壳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
```

然后将个人的调整从 `*.old` 文件复制回去。

### 30.1.2 cron 软件包

使用 `cron` 可在预定义的时间自动在后台运行命令。`cron` 使用格式经过特别设置的时间表，并且该工具随附了几个默认的时间表。用户也可以根据需要指定自定义的表。

`cron` 表位于 `/var/spool/cron/tabs` 中。`/etc/crontab` 作为整个系统的 `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 30.1 “/etc/crontab 中的项”](#) 中，输入的是 `root`。位于 `/etc/cron.d` 中的软件包特定的表具有相同的格式。请参见 `cron` 手册页 ([man cron](#))。

#### 例 30.1：/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`。必须直接将此文件装载到编辑器中，然后对其进行修改并保存。

有些软件包会将外壳脚本安装到 `/etc/cron.hourly`、`/etc/cron.daily`、`/etc/cron.weekly` 和 `/etc/cron.monthly` 目录中，这些脚本的执行由 `/usr/lib/cron/run-crons` 控制。`/usr/lib/cron/run-crons` 每隔 15 分钟会从主表 (`/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。

要按自定义时间运行 `hourly`、`daily` 或其他定期维护脚本，请使用 `/etc/crontab` 项目定期去除时戳文件（请参见 [例 30.2 “/etc/crontab：去除时戳文件”](#)，它会在每个整点之前去除 `hourly` 时戳文件，在每天的早上 2:14 去除 `daily` 时戳文件，等等）。

#### 例 30.2：/ETC/CRONTAB：去除时戳文件

```
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` 设置为应启动 `cron.daily` 的时间。`MAX_NOT_RUN` 的设置确保即使用户很长一段时间都未在指定的 `DAILY_TIME` 打开计算机，日常任务仍被触发运行。`MAX_NOT_RUN` 的最大值为 14 天。

为了清楚起见，将日常系统维护任务分布在多个脚本中。这些脚本包含在软件包 `aaa_base` 中。例如，`/etc/cron.daily` 中包含组件 `suse.de-backup-rpmdb`、`suse.de-clean-tmp` 或 `suse.de-cron-local`。

### 30.1.3 停止 Cron 状态消息

要避免 Cron 状态消息导致的邮件泛滥，在新安装的 `/etc/sysconfig/cron` 中，`SEND_MAIL_ON_NO_ERROR` 的默认值设为 “`no`” 。即使将此设置设为 “`no`” ， cron 数据输出仍然会发送到 `MAILTO` 地址，如 cron 手册页中所述。

在更新时，建议根据需要设置这些值。

### 30.1.4 日志文件：软件包 `logrotate`

多个系统服务（**守护程序**）以及内核本身会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。这些日志文件通常存储 `/var/log` 中（如 FHS 所指定），文件大小每天都会增长。`logrotate` 软件包可以帮助控制这些文件的增长。有关详细信息，请参见《系统分析和微调指南》，第 3 章“系统日志文件”，第 3.3 节“使用 `logrotate` 管理日志文件”。

### 30.1.5 `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 `mlocate` 软件包，它是 `findutils-locate` 软件包的后继者。`updatedb` 进程将在每天晚上或引导系统约 15 分钟后自动启动。

### 30.1.6 `ulimit` 命令

利用 `ulimit`（**用户限制**）命令，您可以限制系统资源的使用并显示这些限制。`ulimit` 对于限制应用程序的可用内存尤其有用。设置可用内存限制后，可以防止应用程序占用过多系统资源，而导致操作系统变慢甚至挂起。

**ulimit**可以对使用多个选项。要限制使用内存，请使用表 30.1 “**ulimit**：为用户设置资源”中列出的选项。

表 30.1：**ulimit**：为用户设置资源

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

系统范围的默认项在 `/etc/profile` 中设置。建议不要直接编辑此文件，因为系统升级期间会覆盖所做的更改。要自定义系统范围的配置文件设置，请使用 `/etc/profile.local`。各用户的设置应该在 `~USER/.profile` 中配置。

例 30.3：**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 为单位指定内存分配。有关详细信息，请参见 [man bash](#)。

## ! 重要：**ulimit** 支持

并非所有外壳都支持 **ulimit** 指令。PAM (例如 `pam_limits`) 作为 **ulimit** 的替代方法，提供了全面的调整功能。

### 30.1.7 `free` 命令

`free` 命令显示系统中总的可用内存、已用物理内存和交换空间，以及内核占用的缓冲区和缓存。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，不允许有剩余或未使用的内存。

内核对任何应用程序或用户数据都没有直接的了解。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，这样，使用 `mmap` 命令便可一开始就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，其中存储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）缓存。

但是如果您的目的是要确定当前所用的 RAM 量，请在 `/proc/meminfo` 中查找此信息。

### 30.1.8 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，这些页面中提供了更深入的说明。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。您可以输入 `emacs -f info` 使用 Emacs 查看信息页，也可以直接在控制台中使用 `info` 来查看。还可以使用 tkinfo、xinfo 或帮助系统来查看信息页。

### 30.1.9 使用 `man` 命令选择手册页

要阅读手册页，请输入 `man MAN_PAGE`。如果不同章节存在同名手册页，所有手册页都会带相应部分编号列出。选择要显示的一个手册页。如果在数秒内未输入部分编号，将显示第一个手册页。

要将此行为更改为默认系统行为，请在外壳初始化文件（如 `~/.bashrc`）中设置 `MAN_POSIXLY_CORRECT=1`。

### 30.1.10 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 会读取包含用户、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` 会从 `/etc/skel` 中安装到各个用户的主目录中。而 `.emacs` 会读取文件 `/etc/skel/.gnu-emacs`。要自定义程序，请将 `.gnu-emacs` 复制到主目录（通过 `cp /etc/skel/.gnu-emacs ~/.gnu-emacs`）并根据您的需求设置。

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

在 SUSE Linux Enterprise Desktop 中，`emacs` 软件包会在目录 `/usr/share/emacs/site-lisp` 中安装 `site-start.el` 文件。文件 `site-start.el` 会在初始化文件 `~/.emacs` 之前装载。除其他作用之外，`site-start.el` 确保自动装载通过 Emacs 附加产品软件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，并且一律以 `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 的部件被分成多个软件包：

- 基础软件包 `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，可安装众多附加产品软件包：`emacs-auctex` (LaTeX)、`psgml` (SGML 和 XML)、`gnuserv` (客户端和服务器操作) 等。

## 30.2 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。

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**。

## 30.3 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/.emacs
/etc/skel/.gnu-emacs
/etc/skel/.vimrc
/etc/csh.cshrc
/etc/termcap
/usr/share/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的应用程序（**vi**、**emacs** 等）。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，可以按照 **/etc/X11/Xmodmap** 中的说明启用组合键（多键）。

可以通过“X 键盘扩展”(XKB) 进行进一步的设置。



### 提示：更多信息

有关 XKB 的信息，请参见 **/usr/share/doc/packages/xkeyboard-config** (**xkeyboard-config** 软件包的一部分) 中所列的文档。

## 30.4 语言和国家/地区特定的设置

该系统在很大程度上实现了国际化，可修改以满足本地需要。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。

通过文件 `/etc/sysconfig/language` 中定义的 `LC_` 变量来进行这些设置。这不仅指**本地语言支持**，还指**消息**（语言）、**字符集**、**排序顺序**、**时间和日期**、**数字和货币**等类别。这些类别中的每一种都可以使用自己的变量直接定义，或使用 `language` 文件中的主变量间接定义（请参见 `locale` 手册页）。

### 变量列表

`RC_LC_MESSAGES`, `RC_LC_CTYPE`, `RC_LC_COLLATE`, `RC_LC_TIME`, `RC_LC_NUMERIC`,  
`RC_LC_MONETARY`

这些变量将不带 `RC_` 前缀传递到外壳，代表所列的类别。下面列出了相关外壳配置文件。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。默认只会设置 `RC_LANG`。这便于用户输入他们自己的值。

### `ROOT_USES_LANG`

此变量可设置为 `yes` 或 `ctype`（默认值）。如果设置为 `yes`，`root` 将使用特定于语言和国家/地区的设置，否则系统管理员将始终在 POSIX 环境中操作。

这些变量可通过 YaST sysconfig 编辑器进行设置。此类变量的值包含语言代码、国家/地区代码、编码和修饰符。各个组成部分之间通过特殊字符联接：

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

### 30.4.1 系统范围的区域设置

systemd 会在引导初期读取 /etc/locale.conf。除非存在单独的设置，否则此文件中配置的区域设置将由每个服务或用户继承。



#### 注意：SUSE Linux Enterprise Desktop 下旧配置文件的行为

早期版本的 SUSE Linux Enterprise Desktop 会从 /etc/sysconfig/language、/etc/sysconfig/keyboard 和 /etc/sysconfig/console 读取区域设置。从 SUSE Linux Enterprise Desktop 15 GA 开始，这些文件被视为已过时。systemd 不再从这些文件读取设置。systemd 改为从 /etc/locale.conf 读取设置，但仍会使用 /etc/sysconfig/language 中定义的变量。这些变量会覆盖系统范围的区域设置，并可用于定义用户外壳的不同区域设置（请参见第 30.4.2 节“一些示例”）。

要设置系统范围的区域设置，您可以采用以下其中一种方式：

- 在 /etc/locale.conf 中写入您的设置。每行指派一个环境类变量（有关变量列表，请参见 man 5 locale.conf）：

```
LANG=de_DE.UTF-8
```

要微调设置，可以添加更多的变量（每行一个变量）。

- 使用命令 localectl：

```
# localectl set-locale LANG=de_DE.UTF-8
```

同样，您也可以在 localectl set-locale 命令后面指定更多的变量。

在更新 systemd 软件包期间为了与旧系统保持向后兼容，上述所有变量都将从 sysconfig 迁移到其最终目标（如果尚未在这些目标中定义这些变量）。

## 30.4.2 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（参见 [http://en.wikipedia.org/wiki/ISO\\_3166](http://en.wikipedia.org/wiki/ISO_3166)）中列出。

仅当 `/usr/lib/locale` 中存在相应的可用描述文件时，设置的值才有意义。可以使用命令 `localeddef` 基于 `/usr/share/i18n` 中的文件创建其他说明文件。说明文件是 `glibc-i18n` 软件包的一部分。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。此设置现已过时，因为 `UTF-8` 也可涵盖欧元符号。仅当应用程序支持 `ISO-8859-15` 而不是 `UTF-8` 时，它才有用。

对 `/etc/sysconfig/language` 所做的更改将通过以下过程链来激活：

- 对于 Bash: `/etc/profile` 会读取 `/etc/profile.d/lang.sh`，后者会分析 `/etc/sysconfig/language`。
- 对于 tcsh: 登录时，`/etc/csh.login` 会读取 `/etc/profile.d/lang.csh`，后者会分析 `/etc/sysconfig/language`。

这样可确保对 `/etc/sysconfig/language` 所做的任何更改在下次登录到相应外壳时即会生效，而无需手动将其激活。

用户可以编辑他们的 `~/.bashrc` 来覆盖系统默认值。例如，如果不想对程序消息使用系统范围的 `en_US`，请加入 `LC_MESSAGES=es_ES`，这样消息将以西班牙语显示。

### 30.4.3 `~/.i18n` 中的区域设置

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` 中的项会覆盖来自 `/etc/sysconfig/language` 中的系统默认值。使用相同的变量名称，但不要包含 `RC_` 名称空间前缀。例如，使用 `LANG` 而非 `RC_LANG`：

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

### 30.4.4 语言支持的设置

消息类别中的文件通常只存储在对应的语言目录（例如 `en`）中以保留后备。如果您将 `LANG` 设置为 `en_US` 并且 `/usr/share/locale/en_US/LC_MESSAGES` 中的消息文件不存在，系统会改为使用 `/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"`

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` 设置为仅包含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`，就会发生这种情况。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。

### 30.4.5 更多信息

- 《**GNU C 库参考手册**》中的“区域设置和国际化”一章。它包含在软件包 `glibc-info`。
- Markus Kuhn 编写的 **Unix/Linux 的 UTF-8 和 Unicode 常见问题解答**，当前位于 <https://www.cl.cam.ac.uk/~mgk25/unicode.html>。

# 31 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。它支持网络连接的顶级加密类型和标准，包括 802.1x 保护的网络的连接。802.1X 是“基于端口的网络访问控制的本地和城域网 IEEE 标准”。使用 NetworkManager，您在外出时就不必顾虑网络接口的配置，以及在有线或无线网络之间切换的问题。NetworkManager 可自动连接到已知无线网络或并行管理多个网络连接 — 然后将最快的连接用作默认连接。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管理网络连接。

不只可激活一个连接，也可同时激活多个。这样您可以将便携式计算机从以太网连接拔出后仍通过无线连接保持连接状态。

## ! 重要：支持覆盖范围

只有安装了 SLED 扩展或 Workstation Extension 的 SUSE 桌面工作负载支持 NetworkManager。所有服务器认证都通过网络配置工具 **wicked** 来完成，使用 NetworkManager 可能会使它们失效。SUSE 服务器工作负载不支持 NetworkManager。

## 31.1 NetworkManager 的用例

NetworkManager 提供了完善且直观的用户界面，可使用户轻松地切换其网络环境。但是，在以下情况下，NetworkManager 解决方案不适用：

- 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。
- 您的计算机为 Xen 服务器或您的系统是 Xen 内的虚拟系统。

## 31.2 启用或禁用 NetworkManager

在台式机和便携式计算机上，NetworkManager 默认处于启用状态。您可以使用 YaST 中的“网络设置”模块随时将其禁用和启用。

1. 运行 YaST，然后转到系统 > 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. 要通过 NetworkManager 配置和管理您的网络连接, 请执行以下操作:
  - a. 在网络设置方法字段中选择通过 NetworkManager 的用户控制方法。
  - b. 单击确定并关闭 YaST。
  - c. 按照[第 31.3 节 “配置网络连接”](#) 中所述通过 NetworkManager 配置您的网络连接。
4. 停用 NetworkManager 并使用您自己的配置控制网络:
  - a. 在网络设置方法字段中选择通过 wicked 进行控制。
  - b. 单击确定。
  - c. 使用 YaST 设置您的网卡: 通过 DHCP 自动配置, 或使用静态外部 IP 地址。  
[第 23.4 节 “使用 YaST 配置网络连接”](#) 中提供了使用 YaST 进行网络配置的详细说明。

### 31.3 配置网络连接

在 YaST 中启用 NetworkManager 后, 使用 GNOME 中提供的 NetworkManager 前端配置网络连接。它会显示所有网络连接类型对应的选项卡, 例如有线、无线、移动宽带、DSL 和 VPN 连接。



#### 提示: NetworkManager 连接编辑器

在以前的 SUSE Linux Enterprise Desktop 版本中, 通过 **NetworkManager 连接编辑器** 应用程序来配置网络连接。现在默认已不再安装此应用程序, 因为 **GNOME 控制中心** 已完全取代了其配置功能。

如果您仍需使用 NetworkManager 连接编辑器来配置网络连接, 请手动安装 [NetworkManager-connection-editor](#) 软件包:

```
> sudo zypper install NetworkManager-connection-editor
```

要在 GNOME 中打开网络配置对话框, 请通过状态菜单打开设置菜单, 然后单击网络项。

## 注意: 选项的可用性

根据您的系统设置, 可能不允许您配置连接。在安全环境中, 某些选项可能会被锁定或需要 root 权限。请咨询系统管理员以了解细节。

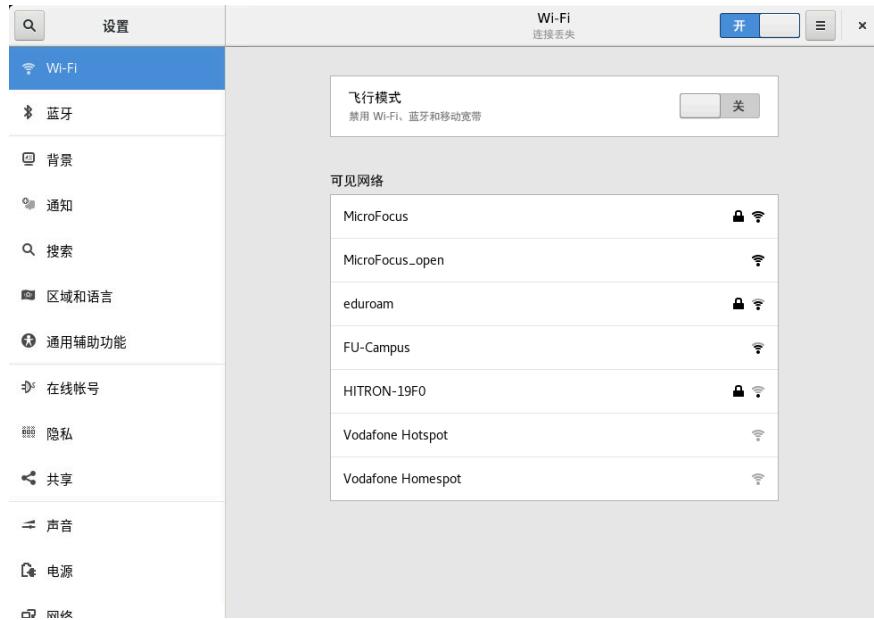


图 31.1: GNOME 网络连接对话框

### 过程 31.1: 添加和编辑连接

- 1. 打开 NetworkManager 配置对话框。**
- 2. 添加连接:**
  - a. 单击左下角的 + 图标。**
  - b. 选择首选连接类型并按照指示操作。**
  - c. 完成后, 单击添加。**
  - d. 确认您的更改后, 新配置的网络连接即会出现在“状态菜单”中的可用网络列表内。**
- 3. 编辑连接:**

- a.** 选择要编辑的项目。
- b.** 单击齿轮图标打开连接设置对话框。
- c.** 插入您的更改，然后单击应用保存更改。
- d.** 要让您的连接可用作系统连接，请转到身份选项卡，选中复选框可用于其他用户。有关用户和系统连接的详细信息，请参见[第 31.4.1 节 “用户和系统连接”](#)。

### 31.3.1 管理有线网络连接

如果您的计算机连接的是有线网络，请使用 NetworkManager 小程序管理连接。

- 1.** 打开“状态菜单”，然后单击有线以更改连接细节或将其关闭。
- 2.** 要更改设置，请单击有线设置，然后单击齿轮图标。
- 3.** 要关闭所有网络连接，请激活飞行模式设置。

### 31.3.2 管理无线网络连接

可见的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 31.2：连接到可见无线网络

- 1.** 要连接可见无线网络，请打开“状态菜单”然后单击 Wi-Fi。
- 2.** 单击开启将其启用。
- 3.** 单击选择网络，选择 Wi-Fi 网络然后单击连接。
- 4.** 如果网络已加密，一个配置对话框将会打开。其中会显示网络使用的加密类型以及用于输入登录凭证的文本框。

#### 过程 31.3：连接到不可见无线网络

- 1.** 要连接到未广播其服务集标识符 (SSID 或 ESSID) 因而无法自动检测到的网络，请打开“状态菜单”，然后单击 Wi-Fi。

2. 单击 Wi-Fi 设置打开详细设置菜单。
3. 确保您的 Wi-Fi 处于启用状态，然后单击连接到隐藏网络。
4. 在打开的对话框中的网络名称中输入 SSID 或 ESSID，并视需要设置加密参数。

显式选中的无线网络将尽可能始终保持连接。如果在此期间插入网线，则会连接任何设置为尽可能保持连接的连接，而无线连接也会保持连接状态。

### 31.3.3 启用无线强制网络门户检测

在连接建立之初，许多公共的无线热点会强制用户访问登录页（称为**强制网络门户**）。在您登录或同意条款与条件之前，所有的 HTTP 请求都会重定向到提供者的强制网络门户。

通过强制网络门户连接到无线网络后，NetworkManager 和 GNOME 会自动显示登录页面（这属于连接过程的一部分）。这样做可以确保一旦连接成功您便能知道，并帮助您在不使用浏览器登录系统的情况下，尽快完成设置。

要启用此功能，请安装 NetworkManager-branding-SLE 软件包并使用以下命令重启 NetworkManager：

```
> sudo systemctl restart network
```

一旦通过强制网络门户连接到网络，NetworkManager（或 GNOME）便会为您打开强制网络门户登录页面。使用您的身分凭证登录，以便访问互联网。

### 31.3.4 将 Wi-Fi/蓝牙网卡配置为接入点

如果您的 Wi-Fi/蓝牙网卡支持接入点模式，则可以使用 NetworkManager 进行配置。

1. 打开“状态菜单”，然后单击 Wi-Fi。
2. 单击 Wi-Fi 设置打开详细设置菜单。
3. 单击用作热点并按照指示操作。
4. 使用随后出现的对话框中显示的身份凭证来连接远程计算机的热点。

### 31.3.5 NetworkManager 和 VPN

NetworkManager 支持多种虚拟专用网 (VPN) 技术。对于每种技术，SUSE Linux Enterprise Desktop 都随附了提供 NetworkManager 常规支持的基础包。此外，还需要为您的小程序安装特定于桌面的软件包。

#### OpenVPN

要使用此 VPN 技术，请安装：

- [NetworkManager-openvpn](#)
- [NetworkManager-openvpn-gnome](#)

#### OpenConnect

要使用此 VPN 技术，请安装：

- [NetworkManager-openconnect](#)
- [NetworkManager-openconnect-gnome](#)

#### PPTP (点对点隧道协议)

要使用此 VPN 技术，请安装：

- [NetworkManager-pptp](#)
- [NetworkManager-pptp-gnome](#)

以下过程介绍如何使用 NetworkManager 将您的计算机设置为 OpenVPN 客户端。设置其他类型 VPN 的过程与此类似。

开始之前，请确保软件包 [NetworkManager-openvpn-gnome](#) 已安装，并且所有依赖项均已解析。

#### 过程 31.4：使用 NETWORKMANAGER 设置 OPENVPN

1. 依次单击位于面板右端的状态图标和扳手和螺丝刀图标，打开应用程序设置。在所有设置窗口中，选择网络。
2. 单击 + 图标。

3. 依次选择 VPN 和 OpenVPN。
4. 选择身份验证类型。根据 OpenVPN 服务器的设置，选择证书 (TLS) 或口令和证书 (TLS)。
5. 将所需的值插入到相应文本框中。对于我们的示例配置，值如下所示：

|       |                                 |
|-------|---------------------------------|
| 网关    | VPN 服务器的远程端点                    |
| 用户名   | 用户 (仅当选择了口令和证书 (TLS) 时才可用)      |
| 口令    | 用户口令 (仅当选择了口令和证书 (TLS) 时才可用)    |
| 用户证书  | <u>/etc/openvpn/client1.crt</u> |
| CA 证书 | <u>/etc/openvpn/ca.crt</u>      |
| 私用密钥  | <u>/etc/openvpn/client1.key</u> |

6. 单击添加完成配置。
7. 要启用连接，请在设置应用程序的网络面板中，单击切换按钮。也可以单击位于面板右端的状态图标，然后依次单击您的 VPN 名称和连接。

## 31.4 NetworkManager 和安全性

NetworkManager 将无线连接分为可信和不可信两种类型。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。可信连接用接入点的名称和 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接入点不可使用。

NetworkManager 会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，NetworkManager 将等待您做出选择。

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所有更新（如新密钥）。

如果您从使用无线连接切换到脱机模式，则 NetworkManager 会将 SSID 或 ESSID 设为空白。这可以确保断开网卡连接。

### 31.4.1 用户和系统连接

NetworkManager 可识别两种类型的连接：user 连接和 system 连接。

用户连接需要在 NetworkManager 中对每个用户进行身份验证，而 NetworkManager 会将用户的身份凭证存储在其本地 GNOME 密钥环中，这样用户便无需在每次连接时都重新输入其身份凭证。

系统连接自动对所有用户可用。创建该连接的第一个用户需输入所有必要的身份凭证，然后，其他所有用户无需知道身份凭证即可拥有访问权限。配置用户连接与配置系统连接的唯一区别在于是否选中了可供其他用户使用复选框。有关如何用 NetworkManager 配置用户连接或系统连接的信息，请参见[第 31.3 节 “配置网络连接”](#)。

### 31.4.2 存储口令和身份凭证

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 GNOME 密钥环管理器将身份凭证加密存储在磁盘上，并用主密码保护。

### 31.4.3 防火墙区域

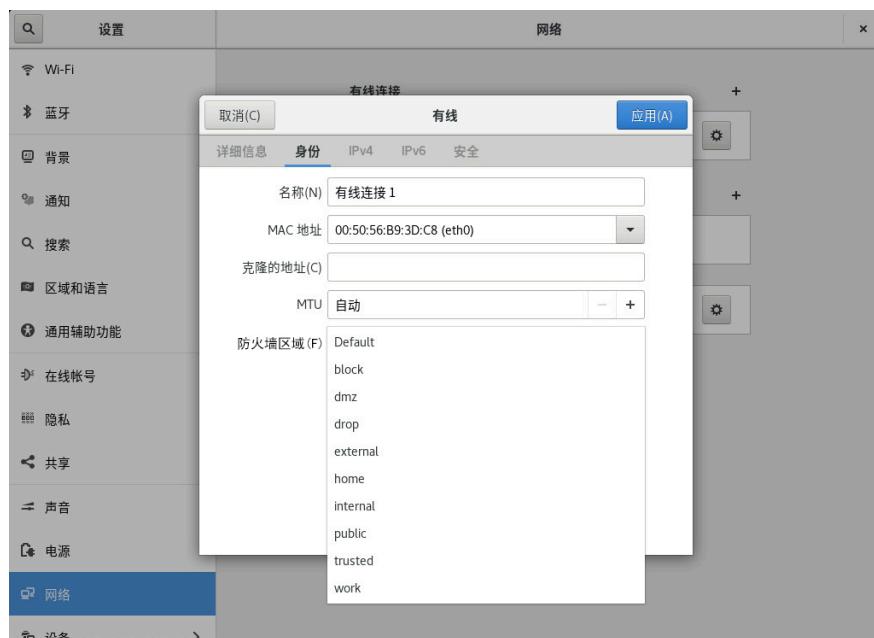


图 31.2：NETWORKMANAGER 中的 firewalld 区域

防火墙区域设置有关允许哪些网络连接的一般规则。要为有线连接配置 **firewalld** 的区域，请转至连接设置的身份选项卡。要为 Wi-Fi 连接配置 **firewalld** 的区域，请转至连接设置的安全选项卡。

如果您位于家庭网络中，请使用 home 区域。如果使用的是公共无线网络，请切换到 public。如果您位于安全的环境中并想要允许所有连接，请使用 trusted 区域。

有关 **firewalld** 的细节，请参见《安全和强化指南》，第 23 章 “掩蔽和防火墙”，第 23.4 节 “firewalld”。

## 31.5 常见问题解答

下面是关于使用 NetworkManager 配置特殊网络选项的一些常见问题。

### 5. 如何将连接绑定到特定设备？

默认情况下，NetworkManager 中的连接是特定于设备类型的：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台物理设备可用（例如您的计算机装有两块以太网卡），您可以将一个连接绑定到特定设备。

要在 GNOME 中执行此操作，请先查找设备的 MAC 地址（使用小程序中提供的连接信息，或者使用 nm-tool 或 wicked show all 等命令行工具的输出）。然后启动配置网络连接的对话框，选择您要修改的连接。在有线或无线选项卡上，输入设备的 MAC 地址，并确认更改。

### 6. 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。在基础架构基本服务集中，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）。

如第 31.3 节 “配置网络连接” 中所述启动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。

## 7. 如何与其他计算机共享网络连接?

主设备（连接到互联网的设备）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. 如第 31.3 节“配置网络连接”中所述启动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，并激活方法下拉框中的共享给其他计算机。这将启用 IP 通讯转发并运行该设备上的 DHCP 服务器。确认在 NetworkManager 中所做更改。
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，请从要允许的服务中选择 DHCP 服务器，然后单击添加。确认在 YaST 中所做更改。

## 8. 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。如第 31.3 节“配置网络连接”中所述启动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，并激活方法下拉框中的仅自动 (DHCP) 地址。在 DNS 服务器和搜索域字段中输入 DNS 信息。要忽略自动取得的路由，请单击路由，然后激活相应的复选框。确认更改。

## 9. 如何在用户登录之前将 NetworkManager 连接到受口令保护的网络？

定义可以用于此目的的 `system connection`。有关详细信息，请参见第 31.4.1 节“用户和系统连接”。

## 31.6 查错

可能出现连接问题。与 NetworkManager 相关的一些常见问题包括小程序不启动或缺少 VPN 选项。解决方法和预防这些问题的方法随使用的工具而定。

### NetworkManager 桌面小程序不启动

如果设置了 NetworkManager 控制的网络，小程序会自动启动。如果小程序未启动，请按照第 31.2 节“启用或禁用 NetworkManager”中所述检查是否在 YaST 中启用了 NetworkManager。然后，请确保 NetworkManager-gnome 软件包也已安装。

如果该桌面小程序已安装但没有运行，请使用命令 `nm-applet` 手动将其启动。

### NetworkManager 小程序未包含 VPN 选项

对 NetworkManager 的支持、小程序以及适用于 NetworkManager 的 VPN 已在不同的软件包中分发。如果 NetworkManager 小程序不包括 VPN 选项，请检查带有 NetworkManager 支持的 VPN 技术的软件包是否已经安装。有关详细信息，请参见第 31.3.5 节 “NetworkManager 和 VPN”。

### 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）也已启动并在正常运行，重启动计算机上的网络接口有时会有所帮助。要执行此操作，请以 `root` 身份登录到命令行，然后运行 `systemctl restart wicd`。

## 31.7 更多信息

可在以下网站和目录中找到有关 NetworkManager 的更多信息：

### NetworkManager 项目页

<https://gitlab.freedesktop.org/NetworkManager/NetworkManager> ↗

### 软件包文档

还可以在以下目录中找到有关 NetworkManager 和 GNOME 小程序的最新信息：

- `/usr/share/doc/packages/NetworkManager/`
- `/usr/share/doc/packages/NetworkManager-gnome/。`

## IV 硬件配置

- 32 设置系统键盘布局 439
- 33 设置声卡 440
- 34 设置打印机 443
- 35 设置扫描仪 449
- 36 电源管理 451
- 37 持久内存 457

## 32 设置系统键盘布局

您可以使用 YaST 系统键盘布局模块来定义系统（以及控制台）的默认键盘布局。用户可以在单个 X 会话中使用桌面工具修改键盘布局。

1. 在 YaST 中单击硬件 > 系统键盘布局启动 YaST 系统键盘配置对话框。或者在命令行中使用 `sudo yast2 keyboard` 启动该模块。
2. 从列表中选择所需的键盘布局。
3. 在测试文本框中尝试所选键盘布局。
4. 如果效果符合预期，请确认所做的更改并关闭对话框。
5. 结果存储在文件 `/etc/vconsole.conf`（用于文本控制台）和 `/etc/X11/xorg.conf.d/00-keyboard.conf`（用于 X11）中。
6. 可在系统 > Sysconfig 编辑器 > 硬件 > 键盘中配置高级键盘设置。您可以在此处指定键盘速度和延迟设置，并启用或禁用 NumLock、CapsLock 和 ScrollLock。这些设置存储在 `/etc/sysconfig/keyboard` 中。

## 33 设置声卡

YaST 可以自动检测大多数声卡，并使用相应的值配置它们。要更改默认设置，或者需要设置不能自动配置的声卡，可以使用 YaST 声音模块。在其中，还可以设置附加声卡或切换它们的顺序。

要启动声音模块，可以启动 YaST 并单击硬件 > 声音。或者，从命令行以 `root` 用户身份运行 `yast2 sound &`，直接启动声音配置对话框。如果声音模块不可用，请使用 `sudo zypper install yast2-sound` 命令加以安装。

### 过程 33.1：配置声卡

如果已添加新声卡或者 YaST 不能自动配置现有声卡，则执行以下步骤。要配置新声卡，需要知道声卡供应商和型号。如果有疑问，可以参考声卡文档以了解所需信息。有关 ALSA 所支持的声卡及其相应声音模块的参考列表，请参见 <http://www.alsa-project.org/main/index.php/Matrix:Main>。

在配置过程中，可以选择以下几个设置选项：

#### 快速自动设置

您不需要执行任何进一步的配置步骤 — 声卡自动进行配置。您可以设置音量或稍后要更改的任意选项。

#### 常规设置

允许您在配置过程中调节输出音量和播放测试声音。

#### 可更改选项的高级设置

仅针对专家。允许您自定义声卡的所有参数。



#### 重要：高级配置

只有在完全清楚自己所执行的操作时，才能使用该选项。否则，保留参数不变，并使用一般设置或自动设置选项。

1. 启动 YaST 声音模块。

2. 要配置检测到但未配置的声卡，请从列表中选择相应项，然后单击编辑。

要配置新声卡，请单击添加。选择声卡供应商和型号，并单击下一步。

3. 选择其中一个设置选项，并单击下一步。
4. 如果选择了常规设置，则现在可以测试声音配置，并对音量进行调节。您应从总音量的 10% 开始，以免损坏您的听力或扬声器。
5. 如果对所有选项设置感到满意，请单击下一步。  
声音配置对话框只会显示新配置或经过修改的声卡。
6. 要删除不再需要的声卡配置，可以选择相应条目并单击删除。
7. 单击确定保存更改，并退出 YaST 声音模块。

#### 过程 33.2：修改声卡配置

1. 要更改个别声卡的配置（仅针对专家！），可以在声音配置对话框中选择声卡条目并单击编辑。  
此时会转到声卡高级选项，在其中可以微调多个参数。有关更多信息，单击帮助。
2. 要调节已配置声卡的音量或测试声卡，可以在声音配置对话框中选择声卡条目，并单击其他。选择相应的菜单项。



#### 注意：YaST 混音器

YaST 混音器设置仅提供基本选项。这些选项旨在用于差错（例如，是否可以听见测试声音）。从其他 > 音量访问 YaST 混音器设置。要每天使用声音选项和对这些选项进行微调，请使用桌面或 **alsasound** 命令行工具提供的混音器小程序。

3. 要播放 MIDI 文件，可以选择其他 > 启动序列发生器。
4. 检测到支持的声卡时，您可以安装 SoundFonts 来播放 MIDI 文件：
  - a. 将原始驱动程序 CD-ROM 插入 CD 或 DVD 驱动器。
  - b. 选择其他 > 安装 SoundFonts 将 SF2 SoundFonts™ 安装到硬盘中。SoundFonts 保存在目录 `/usr/share/sfbank/creative/` 中。
5. 如果在系统中配置了多个声卡，则可以调整声卡的顺序。要将某个声卡设置为主设备，可以在声音配置中选择该声卡，并单击其他 > 设置为主卡。索引为 0 的声音设备是默认设备，由系统和应用程序使用。

6. 默认情况下，SUSE Linux Enterprise Desktop 使用 PulseAudio 声音系统。它是帮助混合多个音频流的抽象层，能够绕过硬件可能会有的任何限制。要启用或禁用 PulseAudio 声音系统，可以单击其他 > PulseAudio 配置。如果启用，PulseAudio 守护程序将用于播放声音。禁用 PulseAudio 支持可在系统范围使用其他设置。

单击确定并退出 YaST 声音模块时，将会保存所有声卡的音量和配置。混音器设置会保存到文件 /etc/asound.state 中。ALSA 配置数据追加到文件 /etc/modprobe.d/sound 的末尾，并写入 /etc/sysconfig/sound。

# 34 设置打印机

YaST 可用于配置区域设置和网络打印机。第 24 章 “打印机操作” 中提供了有关打印的进一步信息（一般信息、技术细节和查错）。

在 YaST 中，单击硬件 > 打印机以启动打印机模块。默认情况下，其会在打印机配置视图中打开，显示所有可用和已配置打印机的列表。这一点在通过网络访问大量打印机时尤其有用。在这里，您还可以打印测试页及配置打印机。



## 注意：启动 CUPS

要使用连接到计算机的打印机，您的系统上必须安装并运行 CUPS。如果 CUPS 未运行，系统会提示您将其启动。如果 CUPS 不会在引导时启动，系统还会提示您启用此设置（建议做法）。

## 34.1 配置打印机

系统通常可以自动检测到 USB 打印机。如果未检测到打印机，请检查打印机是否已打开并连接到计算机。

配置打印机分为三个步骤：指定连接类型、选择驱动程序，以及为此设置命名打印队列。

许多打印机型号可能有多个可用驱动程序。配置打印机时，YaST 默认使用标有 recommended 的驱动程序。通常不需要更改驱动程序。但是，如果您希望彩色打印机只打印黑白颜色，可以使用不支持彩色打印的驱动程序。如果使用 PostScript 打印机打印图形时遇到性能问题，请尝试从 PostScript 切换到 PCL 驱动程序（假设您的打印机支持 PCL）。

如果系统为列出适用于您的打印机的驱动程序，可以尝试从列表中选择相应标准语言的通用驱动程序。请参见您的打印机文档，以找出您的打印机支持的语言（控制打印机的命令集）。如果无法解决问题，请参见第 34.1.1 节 “使用 YaST 添加驱动程序” 了解备选解决方案。

打印机始终都是通过打印队列来使用的。这样可确保将同时执行的作业排入队列，然后逐个处理。每个打印队列都会被指派给一个特定驱动程序，而一台打印机可有多个队列。这样就能在彩色打印机上设置另一个队列，例如，专门用于打印黑白内容。请参见第 24.1 节 “CUPS 工作流程” 以获取有关打印队列的更多信息。

#### 过程 34.1：添加新打印机

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 在打印机配置屏幕中，单击添加。
3. 如果 Specify the Connection 下已列出打印机，请继续执行下一步。否则，请尝试检测更多或启动连接向导。
4. 在 Find and Assign a Driver 下的文本框中输入供应商名称和型号名称，然后单击搜索。
5. 选择与您的打印机匹配的驱动程序。建议您选择最先列出的驱动程序。如果未显示适合的驱动程序，请尝试以下操作。
  - a. 检查搜索词。
  - b. 单击查找更多扩大搜索范围。
  - c. 根据第 34.1.1 节 “使用 YaST 添加驱动程序” 中所述添加驱动程序。
6. 指定 Default paper size。
7. 在设置任意名称字段中，为打印队列输入一个唯一的名称。
8. 此时，打印机配置了默认设置并准备好可供使用。单击确定以返回到打印机配置视图。现在可以在打印机列表中看到新配置的打印机。

#### 34.1.1 使用 YaST 添加驱动程序

并非所有适用于 SUSE Linux Enterprise Desktop 的打印机驱动程序都会默认安装在系统上。在添加打印机时，如果查找并指派驱动程序对话框中没有合适的驱动程序可用，请安装包含适用打印机驱动程序的驱动程序软件包：

#### 过程 34.2：安装其他驱动程序软件包

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 在打印机配置屏幕中，单击添加。
3. 在 Find and Assign a Driver 部分中，单击驱动程序软件包。

4. 从列表中选择一个或多个合适的驱动程序软件包。不要指定打印机描述文件的路径。
5. 选择确定并确认安装软件包。
6. 要直接使用这些驱动程序, 请按照过程 34.1 “添加新打印机” 所述操作。

PostScript 打印机不需要打印机驱动程序软件。PostScript 打印机只需要与特定型号匹配的 PostScript 打印机描述 (PPD) 文件。PPD 文件由打印机制造商提供。

添加 PostScript 打印机时, 如果查找并指派驱动程序对话框中没有合适的 PPD 文件, 请安装适合您打印机的 PPD 文件:

PPD 文件来源有数个。建议先尝试 SUSE Linux Enterprise Desktop 随附但默认未安装的其他驱动程序软件包 (有关安装说明, 请参见下文)。如果这些软件包不包含适合您的打印机的驱动程序, 请直接从打印机供应商处或从 PostScript 打印机的驱动程序 CD 中取得 PPD 文件。有关详细信息, 请参见第 24.8.2 节 “没有合适的 PPD 文件可用于 PostScript 打印机” 。或者, 也可在 <http://www.linuxfoundation.org/collaborate/workgroups/openprinting/database/databaseintro> ( “OpenPrinting.org 打印机数据库” ) 中查找 PPD 文件。使用来自 OpenPrinting 的 PPD 文件时, 请注意它们可能不受 SUSE Linux Enterprise Desktop 支持。

#### 过程 34.3 : 添加 POSTSCRIPT 打印机的 PPD 文件

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 在打印机配置屏幕中, 单击添加。
3. 在 Find and Assign a Driver 部分中, 单击驱动程序软件包。
4. 在 Make a Printer Description File Available 下的文本框中输入 PPD 文件的完整路径。
5. 单击确定返回 Add New Printer Configuration 屏幕。
6. 要想直接使用该 PPD 文件, 请按照过程 34.1 “添加新打印机” 中所述操作。

## 34.1.2 编辑本地打印机配置

您可以编辑打印机的现有配置来更改连接类型和驱动程序等基本设置。还可以调整纸张大小、分辨率、媒体源等的默认设置。您可以通过更改打印机描述或位置来更改打印机的标识符。

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 在打印机配置屏幕的列表中选择本地打印机配置，然后单击编辑。
3. 按[过程 34.1 “添加新打印机”](#) 中所述更改连接类型或驱动程序。只有当前配置存在问题时，才需要执行该操作。
4. (选择性) 选中默认打印机将该打印机设为默认打印机。
5. 单击当前驱动程序的所有选项调整默认设置。要更改设置，可以单击 + 符号展开选项列表。通过单击选项更改默认值。单击确定应用您的更改。

## 34.2 使用 YaST 配置网络打印

无法自动检测到网络打印机。必须使用 YaST 打印机模块手动进行配置。根据网络设置的不同，您可以打印到打印服务器（CUPS、LPD、SMB 或 IPX），或直接打印到网络打印机（最好通过 TCP）。通过从 YaST 打印机模块的左窗格中选择[通过网络打印](#)来访问网络打印配置视图。

### 34.2.1 使用 CUPS

在 Linux 环境中，使用 CUPS 来通过网络打印。最简单的设置是仅通过所有客户端都可直接访问的一台 CUPS 服务器来打印。通过多个 CUPS 服务器进行打印时，需要运行与远程 CUPS 服务器通讯的本地 CUPS 守护程序。

#### ! 重要：浏览网络打印队列

CUPS 服务器通过传统的 CUPS 浏览协议或 Bonjour/DNS-SD 在网络上通告它们的打印队列。客户端需要浏览这些列表，以使用户能够选择要向其发送打印作业的特定打印机。要浏览网络打印队列，则必须在通过 CUPS 服务器打印的所有客户端上运行 `cups-filters-cups-browsed` 软件包提供的 `cups-browsed` 服务。使用 YaST 配置网络打印时，`cups-browsed` 会自动启动。

如果在启动 `cups-browsed` 之后无法浏览，则表明 CUPS 服务器可能是通过 Bonjour/DNS-SD 通告网络打印队列的。在此情况下，您需要另外安装 `avahi` 软件包，并在所有客户端上使用 `sudo systemctl start avahi-daemon` 启动关联的服务。

#### 过程 34.4：通过单台 CUPS 服务器打印

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 在左侧窗格中，选择通过网络打印。
3. 选中直接通过一台远程 CUPS 服务器来执行您的所有打印并指定该服务器的名称或 IP 地址。
4. 单击测试服务器以确保您已选择正确的名称或 IP 地址。
5. 单击确定返回打印机配置屏幕。现在会列出通过 CUPS 服务器提供的所有打印机。

#### 过程 34.5：通过多个 CUPS 服务器进行打印

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 在左侧窗格中，选择通过网络打印。
3. 选中接受来自 CUPS 服务器的打印机公告。
4. 在 General Settings 下，指定要使用的服务器。您可以接受来自所有可用网络或来自特定主机的连接。如果选择后一个选项，则需要指定主机名或 IP 地址。
5. 在系统提示您启动本地 CUPS 服务器时，依次单击确定和是。服务器启动之后，YaST 会返回打印机配置屏幕。单击刷新列表查看目前为止检测到的打印机。

### 34.2.2 使用 CUPS 之外的打印服务器

如果网络通过 CUPS 以外的其他打印服务器提供打印服务，可以单击硬件 > 打印机启动 YaST 打印机模块，然后从左侧窗格中选择通过网络打印。启动连接向导并选择相应的连接类型。关于在您的环境中配置网络打印机的细节，请咨询您的网络管理员。

### 34.3 通过网络共享打印机

您可以通过网络共享本地 CUPS 守护程序所管理的打印机，这实际上是将您的计算机变成了 CUPS 服务器。可通过启用 CUPS 中所谓的“浏览模式”来共享打印机。如果浏览已启用，就可将本地打印队列用于在网络上侦听远程 CUPS 守护程序。您也可以设置专用 CUPS 服务器来管理所有打印队列，让远程客户端可以直接访问该服务器。在这种情况下，不需要启用浏览。

### 过程 34.6：共享打印机

1. 通过硬件 > 打印机启动 YaST 打印机模块。
2. 从左侧窗格中选择共享打印机。
3. 选择允许远程访问。另外请选中对于本地网络中的计算机，并通过选中默认情况下，在本地网络中发布打印机启用浏览模式。
4. 单击确定重启动 CUPS 服务器，并返回到打印机配置屏幕。
5. 有关 CUPS 和防火墙设置，请参见 [https://en.opensuse.org/SDB:CUPS\\_and\\_SANE\\_Firewall\\_settings](https://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings)。

# 35 设置扫描仪

您可以使用 YaST 配置 USB 或 SCSI 扫描仪。[sane-backends](#) 包中包含使用扫描仪所需的硬件驱动程序和其他必要数据。如果您有 HP 多功能一体机，请参见[第 35.1 节 “配置 HP 多功能一体机”](#)，有关如何配置网络扫描仪的指导，可在[第 35.3 节 “通过网络进行扫描”](#) 中找到。

## 过程 35.1：配置 USB 或 SCSI 扫描仪

1. 将 USB 或 SCSI 扫描仪与计算机连接，并打开它。
2. 启动 YaST，然后选择硬件 > 扫描仪。YaST 构建了扫描仪数据库，它会尝试自动检测您的扫描仪型号。  
如果未正确检测到 USB 或 SCSI 扫描仪，则尝试其他 > 重启检测。
3. 要激活扫描仪，请从检测到的扫描仪列表中选择它，然后单击编辑。
4. 从列表中选择型号，然后单击下一步和完成。
5. 使用其他 > 测试可确保选择了正确的驱动程序。
6. 通过单击确定退出配置屏幕。

## 35.1 配置 HP 多功能一体机

HP 多功能一体机可使用 YaST 来配置，即使它是通过网络使用也是如此。如果您具有 USB HP 多功能一体机，可以按[过程 35.1 “配置 USB 或 SCSI 扫描仪”](#) 中所述开始配置。如果正确检测到它，并且测试成功，则它已可供使用。

如果未正确检测到 USB 设备，或者 HP 多功能一体机已联网，可以运行 HP 设备管理器：

1. 启动 YaST，然后选择硬件 > 扫描仪。YaST 会装载扫描仪数据库。
2. 通过其他 > 运行 hp-setup 启动 HP 设备管理器，并遵循屏幕指示信息执行操作。完成 HP 设备管理器操作之后，YaST 扫描仪模块会自动重启自动检测。
3. 选择其他 > 测试进行测试。
4. 通过单击确定退出配置屏幕。

## 35.2 通过网络共享扫描仪

SUSE Linux Enterprise Desktop 支持通过网络共享扫描仪。要实现该目的，可以如下配置扫描仪：

1. 按第 35 章 “设置扫描仪” 中所述配置扫描仪。
2. 选择其他 > 通过网络进行扫描。
3. 在服务器设置 > saned 允许的客户端下输入允许使用扫描仪的客户端的主机名（使用逗号分隔），然后单击确定退出配置对话框。

## 35.3 通过网络进行扫描

要使用通过网络进行共享的扫描仪，可以执行以下操作：

1. 启动 YaST，然后选择硬件 > 扫描仪。
2. 通过其他 > 通过网络进行扫描打开网络扫描仪配置菜单。
3. 在客户端设置 > 用于 Net 元驱动程序的服务器下输入扫描仪所连接的计算机的主机名。
4. 使用确定离开。网络扫描仪现在会在“扫描仪配置”窗口中列出，并可供使用。

# 36 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。ACPI（高级配置和电源接口）在所有现代计算机（便携式计算机、台式机和服务器）上都可用。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。

## 36.1 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。

### 暂停（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，计算机耗费的电量非常少。此状态的优点是无需引导和重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。

### 休眠（暂停到磁盘）

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。系统会恢复到暂停前的状态。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。



### 注意：通过 `mkswap` 格式化时更改了交换分区的 UUID

如果可能，请不要使用 `mkswap` 重新设置现有交换分区的格式。使用 `mkswap` 重新设置格式将会更改交换分区的 UUID 值。请通过 YaST 重新设置格式（这会更新 `/etc/fstab`），或者手动调整 `/etc/fstab`。

## 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）、限制和使处理器进入休眠 (C-state)。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。

# 36.2 高级配置和电源接口 (ACPI)

ACPI 旨在支持操作系统设置和控制各个硬件组件。ACPI 取代即插即用电源管理 (PnP) 和高级电源管理 (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停用部件等任务。因为操作系统执行 BIOS 中存储的命令，所以功能取决于 BIOS 实施。journald 中报告了 ACPI 能够检测并装载的表。有关查看日志消息的更多信息，请参见[第 21 章 “journalctl：查询 systemd 日志”](#)。有关对 ACPI 问题进行故障诊断的详细信息，请参见[第 36.2.2 节 “查错”](#)。

## 36.2.1 控制 CPU 性能

CPU 可以采用三种节能方法：

- 频率和电压调节
- 限制时钟频率 (T-state)
- 使处理器进入休眠 (C-state)

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。

仅当处理器忙时，才需要进行频率调节和限制，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-state。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。最佳方法是通过内核按需调节器来控制动态频率调节。

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，如果有些系统的节流程度过高，将无法顺畅运行。此外，如果 CPU 处理的任务量很少，则 CPU 节流就没什么作用。

有关详细信息，请参见《系统分析和微调指南》，第 11 章“电源管理”。

## 36.2.2 查错

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。更多情况下，问题是由于 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。如果计算机未引导，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，使用命令 `dmesg -T | grep -2i acpi` 来监视系统的引导消息（或所有消息，因为问题也可能是 ACPI 以外的因素所导致）。如果在分析 ACPI 表时出错，则最重要的表 DSDT（**区分系统描述表**）可替换为改进的版本。在这种情况下，将忽略 BIOS 中有问题的 DSDT。[第 36.4 节 “查错”](#) 中对这一过程进行了介绍。

在内核配置中，可以使用开关来激活 ACPI 调试消息。如果编译和安装的是带有 ACPI 调试功能的内核，则会显示详细信息。

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。

### 36.2.2.1 更多信息

- <https://tldp.org/HOWTO/ACPI-HOWTO/> (详细的 ACPI HOWTO 文档，包含 DSDT 补丁)
- <https://uefi.org/specifications> (高级配置和电源接口规范)

## 36.3 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，如果要最大程度地节能，请使用 `hdparm` 命令尝试以下方法。

该命令可用于修改硬盘设置。选项 `-y` 可立即将硬盘切换到待机模式。`-Y` 会使其进入休眠状态。`hdparm -S X` 会使硬盘在一段时间未活动后运行速度减慢。使用以下值替换 `X`: `0` 表示禁用此机制，会使硬盘持续运行。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。要减少硬盘噪音，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。此缓冲区由 `pdflush` 守护程序监视。当数据达到一定的有效期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

包含截至 `pdflush` 线程唤醒的延迟（以百分之一秒为单位）。

#### /proc/sys/vm/dirty\_expire\_centisecs

定义一个时间范围，最晚应在该时间之后写入未写入页。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。默认值为 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间隔期间写入未写入缓冲区，而不是继续写入。



### 警告：数据完整性风险

对 `pdflush` 守护程序设置进行更改可能会损害数据完整性。

除了这些进程之外，`Btrfs`、`Ext3`、`Ext4` 等日志文件系统以及其他系统会独立于 `pdflush` 写入它们的元数据，这也会阻止硬盘休眠。为了避免这种情况，已为移动设备开发了特殊的内核扩展。要使用该扩展，请安装 `laptop-mode-tools` 软件包；有关细节，请参见 /usr/src/linux/Documentation/laptops/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，这些技术由 `laptop-mode-tools` 控制。

## 36.4 查错

所有错误消息和警报都记录在系统日志中，您可使用 **`journalctl`** 命令来查询该日志（有关详细信息，请参见第 21 章 “**`journalctl`：查询 `systemd` 日志**”）。以下几个部分介绍最常见的问题。

### 36.4.1 CPU 频率功能不工作

请参见内核源以确认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。如果安装了 `kernel-source` 软件包，可在 `/usr/src/linux/Documentation/cpu-freq/*` 中找到此信息。

# 37 持久内存

本章包含有关使用配备非易失性主内存的 SUSE Linux Enterprise 的附加信息。此类内存称为**持久内存**，由一个或多个 NVDIMM 构成。

## 37.1 简介

持久内存是一种新型的计算机存储，其速度接近动态 RAM (DRAM)，同时兼具 RAM 的按字节寻址能力以及固态硬盘 (SSD) 的性能。

SUSE 目前支持在采用 AMD64/Intel 64 和 POWER 体系结构的计算机上，对 SUSE Linux Enterprise Server 使用持久性内存。

与传统的 RAM 一样，持久内存直接安装在主板上的内存插槽中。因此，它的物理外形规格与 RAM 相同，以 DIMM 的形式提供。这些内存称为 NVDIMM：非易失性双列直插式内存模块。

不过，与 RAM 不同，持久内存存在多个方面类似于基于闪存的 SSD。后两者采用固态内存电路的形式，但除此之外，两者都提供非易失性存储：系统断电或者重启动后，内存中的内容会得到保留。使用这两种媒体时，写入数据的速度比读取数据要慢；两者都支持有限的重新写入周期数。最后，与 SSD 一样，如果在特定的应用方案中更适合对持久内存进行扇区级别的访问，则也可以这样做。

不同的型号使用不同形式的电子存储媒体，例如 Intel 3D XPoint，或者将 NAND 闪存与 DRAM 结合使用。另外，行业正在开发新形式的非易失性 RAM。这意味着，不同的供应商和 NVDIMM 型号会提供不同的性能和持久性特征。

由于涉及的存储技术处于早期开发阶段，不同供应商的硬件可能会施加不同的限制。因此，以下叙述适用于一般性的场合。

持久内存的速度最多比 DRAM 要慢 10 倍，但比闪存要快大约 1000 倍。可在其中按字节重新写入数据，而不像在闪存中一样，需要擦除整个扇区，然后重新写入数据。最后，尽管重新写入周期数有限，但大部分形式的持久内存可以应对数百万次重新写入，相比之下，闪存只能应对数千个周期。

这会产生两种重要后果：

- 使用最新的技术无法运行仅包含持久内存的系统，因此无法实现非易失性的主内存。必须混合使用传统的 RAM 和 NVDIMM。操作系统和应用程序将在传统的 RAM 中执行，而 NVDIMM 可提供快速的补充性存储。
- 由于不同供应商的持久内存的性能特征不同，程序员可能需要考虑到特定服务器中 NVDIMM 的硬件规格，包括 NVDIMM 的数量，以及它们可以挂载到哪些内存插槽。这会影响超级管理程序的使用、不同主机之间的软件迁移等等。

ACPI 标准版本 6 中定义了此新型存储子系统。但是，libnvdimm 支持该标准颁布之前的 NVDIMM，可以相同的方式使用这些内存。



## 提示：Intel Optane DC 持久内存

Intel Optane DIMM 内存可在特定模式下使用：

- 在 **App Direct** 模式下，Intel Optane 内存可用作快速永久存储设备来替代 SSD 和 NVMe 设备。当系统断电时，此模式下的数据会保留。  
从 SLE 12 SP4 起已开始支持 App Direct 模式。
- 在 **内存模式**下，Intel Optane 内存可作为 DRAM 的一种经济高效的大容量替代品。在此模式下，单独的 DRAM DIMM 用于缓存最常访问的数据，而 Optane DIMM 内存则提供大内存容量。但是，与纯 DRAM 系统相比，这种模式在随机访问工作负载下速度较慢。如果您运行的应用程序不具有可利用此模式优势的 Optane 特定增强功能，内存性能可能会降低。此模式下的数据会在系统断电时丢失。  
从 SLE 15 SP1 起已开始支持内存模式。
- 在 **混合模式**下，Intel Optane 内存被分区，因此它可以同时在两种模式下使用。  
从 SLE 15 SP1 起已开始支持混合模式。

有关 Intel Optane DC 持久内存的更多详细信息，请参阅 <https://www.intel.com/content/dam/support/us/en/documents/memory-and-storage/data-center-persistent-mem/Intel-Optane-DC-Persistent-Memory-Quick-Start-Guide.pdf>。

## 37.2 术语

### 区域

**区域**是持久内存中可划分为一个或多个**名称空间**的块。如果不事先将某个区域分配到名称空间，则您无法访问该区域的持久内存。

### 名称空间

非易失性存储空间的单个连续寻址范围，可与 NVM Express SSD 名称空间或 SCSI 逻辑单元 (LUN) 相媲美。名称空间作为单独的块设备显示在服务器的 `/dev` 目录中。根据所需的访问方法，名称空间可将多个 NVDIMM 中的存储合并成较大的卷，或者允许将这些存储分区成较小的卷。

### 模式

每个名称空间还具有一种**模式**，该模式定义要为该名称空间启用哪些 NVDIMM 功能。同一父区域的同级名称空间一律具有相同的类型，但可将其配置为使用不同的模式。名称空间模式包括：

#### devdax

设备 DAX 模式。创建单字符设备文件 (`/dev/daxX.Y`)。不需要创建文件系统。

#### fsdax

文件系统 DAX 模式。如果未指定其他模式，则使用默认值。创建支持 `ext4` 或 `XFS` DAX 的块设备 (`/dev/pmemX [ .Y ]`)。

#### 扇区

适用于不执行元数据校验和计算的传统文件系统。适用于小型引导卷。与其他操作系统兼容。

#### raw

不包含卷标或元数据的内存磁盘。不支持 DAX。与其他操作系统兼容。



#### 注意

SUSE 不支持 `raw` 模式。无法在 `raw` 名称空间中挂载文件系统。

## 类型

每个名称空间和区域都有一种**类型**，该类型定义可采用何方式访问与该名称空间或区域关联的持久内存。名称空间的类型始终与其父区域的类型相同。有两种不同的类型：“持久内存”（可通过两种不同的方式进行配置），以及已弃用的“块模式”。

### 持久内存 (PMEM)

与 RAM 类似，PMEM 存储提供字节级别的访问。使用 PMEM 时，单个名称空间可以包含多个交错式的 NVDIMM，使这些 NVDIMM 都可用作单个设备。可通过两种方式来配置 PMEM 名称空间。

#### 将 PMEM 与 DAX 搭配使用

为 Direct Access (DAX) 配置 PMEM 名称空间后，访问内存时会绕过内核的页面缓存，并直接进入媒体。软件可以单独直接读取或写入该名称空间的每个字节。

#### 使用块转换表 (BTT) 的 PMEM

与在传统的磁盘驱动器中一样，将按扇区访问配置为以 BTT 模式运行的 PMEM 名称空间，而不是像在 RAM 中一样采用按字节寻址的模式。某个转换表机制会将访问活动批处理成扇区大小的单元。

BTT 的优点是具有数据保护功能。该存储子系统可确保完全写入底层媒体的每个扇区。如果某个扇区无法完全写入（即，如果写入操作由于某个原因失败），整个扇区将会回滚到其原先的状态。因此，无法在给定的扇区中进行部分写入。

此外，对 BTT 名称空间的访问会由内核缓存。

缺点在于，BTT 名称空间不支持 DAX。

### 块模式 (BLK)

块模式存储将每个 NVDIMM 作为独立的设备进行寻址。此模式已遭弃用，且不再受支持。

除 devdax 名称空间以外，其他所有类型也必须使用文件系统格式化，如同使用传统的驱动器一样。SUSE Linux Enterprise Desktop 支持将 ext2、ext4 和 XFS 文件系统用于此目的。

### 直接访问 (DAX)

DAX 允许将持久内存直接映射成进程的地址空间（例如，使用 mmap 系统调用进行映射）。

## DIMM 物理地址 (DPA)

在单个 DIMM 的内存中充当偏移量的内存地址；即，在该 DIMM 中充当最低可寻址字节的从零开始的地址。

## 标签

存储在 NVDIMM 中的元数据，例如名称空间定义。可以使用 DSM 访问这些数据。

## 设备特定的方法 (DSM)

用于访问 NVDIMM 中的固件的 ACPI 方法。

# 37.3 使用案例

## 37.3.1 将 PMEM 与 DAX 搭配使用

这种形式的内存访问不是事务性的。如果发生断电或其他系统故障，数据可能不会完全写入到存储中。仅当应用程序可以处理部分写入数据的情况时，PMEM 存储才适用。

### 37.3.1.1 可受益于较大字节可寻址存储量的应用程序

如果服务器托管的某个应用程序可按字节直接使用较大的快速存储量，则程序员可以使用 [mmap](#) 系统调用，将持久内存块直接放入该应用程序的地址空间，而无需使用任何附加的系统 RAM。

### 37.3.1.2 避免使用内核页面缓存

为了节省用于页面缓存的 RAM，请避免使用内核页面缓存，而是应将 RAM 分配给应用程序。例如，可以专门使用非易失性内存来保存虚拟机 (VM) 映像。由于这些映像不会缓存，因此可以减少主机上的缓存使用量，从而可在每台主机上配置更多的 VM。

### 37.3.2 将 PMEM 与 BTT 搭配使用

如果您想使用一组 NVDIMM 中的持久内存作为类似磁盘的快速存储池，可以使用此方法。例如，将文件系统日志置于使用 BTT 的 PMEM 上，可以提升在发生电源故障或其他突发性中断后所执行文件系统恢复的可靠性（请参见第 37.5.3 节“创建使用 BTT 的 PMEM 名称空间”）。

对应用程序而言，此类设备就像是快速 SSD 一样，能当成其他任何存储设备一样使用。例如，LVM 可以分布在持久内存的顶层，像平时一样正常工作。

BTT 的优点在于可以保证扇区写入的原子性，因此，即使是依赖于数据完整性的复杂应用程序也能保持正常工作。可通过标准的错误报告通道来运行媒体错误报告。

## 37.4 用于管理持久内存的工具

要管理持久内存，必须安装 `ndctl` 软件包。安装此软件包也会安装 `libndctl` 软件包，后者提供一组用户空间库，用于配置 NVDIMM。

这些工具通过 `libnvdimm` 库运行。该库支持三种类型的 NVDIMM：

- PMEM
- BLK
- 同步 PMEM 和 BLK。

`ndctl` 实用程序提供一系列有用的 `man` 页。可使用以下命令访问这些页面：

```
> ndctl help subcommand
```

要查看可用子命令的列表，请使用：

```
> ndctl --list-cmds
```

可用的子命令包括：

#### **version**

显示 NVDIMM 支持工具的当前版本。

#### **enable-namespace**

使指定的名称空间可供使用。

## **disable-namespace**

阻止使用指定的名称空间。

## **create-namespace**

从指定的存储设备创建新的名称空间。

## **destroy-namespace**

去除指定的名称空间。

## **enable-region**

使指定的区域可供使用。

## **disable-region**

阻止使用指定的区域。

## **zero-labels**

擦除设备中的元数据。

## **read-labels**

检索指定设备的元数据。

## **list**

显示可用的设备。

## **help**

显示有关工具用法的信息。

# 37.5 设置持久内存

## 37.5.1 查看可用的 NVDIMM 存储

可以使用 **ndctl list** 命令列出系统中所有可用的 NVDIMM。

在以下示例中，系统包含三个 NVDIMM，这些 NVDIMM 位于单个三通道交错集内。

```
# ndctl list --dimms  
[
```

```
{  
  "dev": "nmem2",  
  "id": "8089-00-0000-12325476"  
,  
  {  
    "dev": "nmem1",  
    "id": "8089-00-0000-11325476"  
,  
  {  
    "dev": "nmem0",  
    "id": "8089-00-0000-10325476"  
}  
]
```

如果结合不同的参数，[ndctl list](#) 还可以列出可用的区域。



## 注意

区域可能不会按数字顺序显示。

请注意，尽管只有三个 NVDIMM，但它们却显示为四个区域。

```
# ndctl list --regions  
  
[  
  {  
    "dev": "region1",  
    "size": 68182605824,  
    "available_size": 68182605824,  
    "type": "blk"  
,  
  {  
    "dev": "region3",  
    "size": 202937204736,  
    "available_size": 202937204736,  
    "type": "pmem",  
    "iset_id": 5903239628671731251  
,  
  {
```

```
  "dev": "region0",
  "size": 68182605824,
  "available_size": 68182605824,
  "type": "blk"
},
{
  "dev": "region2",
  "size": 68182605824,
  "available_size": 68182605824,
  "type": "blk"
}
]
```

空间以两种不同的形式显示：三个 BLK 类型的独立 64 GB 区域，或者一个 PMEM 类型的合并 189 GB 区域，后者将三个交错式 NVDIMM 中的所有空间表示为单个卷。

请注意，available\_size 的显示值与 size 的显示值相同。这意味着尚未分配任何空间。

### 37.5.2 将存储配置为使用 DAX 的单个 PMEM 名称空间

第一个示例将三个 NVDIMM 配置成使用 Direct Access (DAX) 的单个 PMEM 名称空间。

第一个步骤是创建新的名称空间。

```
# ndctl create-namespace --type=pmem --mode=fsdax --map=memory
{
  "dev": "namespace3.0",
  "mode": "memory",
  "size": 199764213760,
  "uuid": "dc8ebb84-c564-4248-9e8d-e18543c39b69",
  "blockdev": "pmem3"
}
```

这会创建支持 DAX 的块设备 /dev/pmem3。设备名称中的 3 继承自父区域编号（在本例中为 region3）。

--map=memory 选项会将 NVDIMM 上的 PMEM 存储空间预留一部分，用来分配给称作 struct pages 的内部内核数据结构。这样，便可以将新的 PMEM 名称空间与 0\_DIRECT I/O 和 RDMA 等功能搭配使用。

最终 PMEM 名称空间的容量之所以小于父 PMEM 区域，是因为有一部分持久内存预留给了内核数据结构。

接下来，我们校验新的块设备是否可用于操作系统：

```
# fdisk -l /dev/pmem3
Disk /dev/pmem3: 186 GiB, 199764213760 bytes, 390164480 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

与其他任何驱动器一样，在使用该设备之前，必须先将其格式化。在本示例中，我们使用 XFS 将其格式化：

```
# mkfs.xfs /dev/pmem3
meta-data=/dev/pmem3      isize=256    agcount=4, agsize=12192640 blks
                      =       sectsz=4096  attr=2, projid32bit=1
                      =       crc=0      finobt=0, sparse=0
data      =       bsize=4096   blocks=48770560, imaxpct=25
          =       sunit=0      swidth=0 blks
naming   =version 2      bsize=4096   ascii-ci=0 ftype=1
log       =internal log   bsize=4096   blocks=23813, version=2
          =       sectsz=4096  sunit=1 blks, lazy-count=1
realtime =none            extsz=4096   blocks=0, rtextents=0
```

接下来，可将新的驱动器挂载到某个目录：

```
# mount -o dax /dev/pmem3 /mnt/pmem3
```

然后，可以校验是否获得了一个支持 DAX 的设备：

```
# mount | grep dax
/dev/pmem3 on /mnt/pmem3 type xfs (rw,relatime,attr2,dax,inode64,noquota)
```

结果是，我们已获得一个使用 XFS 文件系统格式化的，且装有 DAX 的 PMEM 名称空间。

对该文件系统中的文件进行的任何 mmap() 调用都会返回直接映射到 NVDIMM 上的持久内存的虚拟地址，并且会绕过页面缓存。

对该文件系统中的文件进行任何 fsync 或 msync 调用仍可确保将修改后的数据完全写入到 NVDIMM。这些调用会刷新通过 mmap 映射在用户空间中修改的任何页面的关联处理器缓存行。

### 37.5.2.1 去除名称空间

在创建使用相同存储的其他任何类型的卷之前，我们必须卸载此 PMEM 卷，然后将其去除。

首先卸载该卷：

```
# umount /mnt/pmem3
```

然后禁用名称空间：

```
# ndctl disable-namespace namespace3.0
disabled 1 namespace
```

然后删除该卷：

```
# ndctl destroy-namespace namespace3.0
destroyed 1 namespace
```

### 37.5.3 创建使用 BTT 的 PMEM 名称空间

BTT 提供原子性扇区写入，因此，当您需要数据保护（例如，为 Ext4 和 XFS 日志提供数据保护）时，BTT 会是个不错的选择。如果发生电源故障，日志将受到保护且可恢复。以下示例演示如何创建使用 BTT 的扇区模式 PMEM 名称空间，以及如何将文件系统日志置于此名称空间。

```
# ndctl create-namespace --type=pmem --mode=sector
{
  "dev": "namespace3.0",
  "mode": "sector",
  "uuid": "51ab652d-7f20-44ea-b51d-5670454f8b9b",
  "sector_size": 4096,
  "blockdev": "pmem3s"
}
```

接下来，校验新设备是否存在：

```
# fdisk -l /dev/pmem3s
Disk /dev/pmem3s: 188.8 GiB, 202738135040 bytes, 49496615 sectors
Units: sectors of 1 * 4096 = 4096 bytes
Sector size (logical/physical): 4096 bytes / 4096 bytes
```

```
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
```

与前面配置的支持 DAX 的 PMEM 名称空间一样，这个支持 BTT 的 PMEM 名称空间也会占用 NVDIMM 中的所有可用存储。



## 注意

设备名称 (`/dev/pmem3s`) 末尾的 `s` 表示 `sector` 扇区，可用于轻松辨别配置为使用 BTT 的名称空间。

可按前一示例中所述格式化和挂载卷。

此处显示的 PMEM 名称空间不能使用 DAX。它会使用 BTT 来提供 **扇区写入原子性**。每次通过 PMEM 块驱动程序进行扇区写入时，BTT 都会分配一个新的扇区来接收新数据。完全写入新数据后，BTT 将以原子方式更新其内部映射结构，使新写入的数据可供应用程序使用。如果在此过程中的任意时刻发生电源故障，写入内容将会丢失，在这种情况下，应用程序可以访问其旧数据，而这些数据仍旧保持不变。这可以防止出现所谓“扇区撕裂”的情况。

与其他任何标准块设备一样，可以使用某个文件系统格式化这个支持 BTT 的 PMEM 名称空间，并在该文件系统中使用它。无法将该名称空间与 DAX 搭配使用。但是，此块设备中的文件的 `mmap` 映射将使用页面缓存。

### 37.5.4 将文件系统日志置于 PMEM/BTT 上

当您将文件系统日志置于一个单独的设备上时，该设备使用的文件系统块大小必须与该文件系统相同。此大小很有可能是 4096，您可以使用以下命令查找块大小：

```
# blockdev --getbsz /dev/sda3
```

下面的示例会在单独的 NVDIMM 设备上创建新的 Ext4 日志，并在 SATA 设备上创建文件系统，然后将新文件系统关联到该日志：

```
# mke2fs -b 4096 -0 journal_dev /dev/pmem3s
# mkfs.ext4 -J device=/dev/pmem3s /dev/sda3
```

下面的示例会在 SATA 驱动器上创建新的 XFS 文件系统，并在单独的 NVDIMM 设备上创建日志：

```
# mkfs.xfs -l logdev=/dev/pmem3s /dev/sda3
```

有关选项的详细信息，请参见 [man 8 mkfs.ext4](#) 和 [man 8 mkfs.xfs](#)。

## 37.6 更多信息

可以在以下列表中找到关于此主题的更多信息：

- [Persistent Memory Wiki \(<https://nvdimm.wiki.kernel.org/>\)](#)

包含有关配置 NVDIMM 系统的指导、有关测试的信息，以及有关启用 NVDIMM 的规范的链接。随着 Linux 中 NVDIMM 支持功能的不断发展，此站点的内容也会不断扩充。

- [Persistent Memory Programming \(<http://pmem.io/>\)](#)

有关在 Linux 和其他操作系统中配置、使用非易失性内存，以及为使用此类内存的系统编程的信息。其中介绍了 NVM 库 (NVML)。该库旨在提供有用的 API 用于在用户空间中进行持久性内存编程。

- [LIBNVDIMM: Non-Volatile Devices \(<https://www.kernel.org/doc/Documentation/nvdimm/nvdimm.txt>\)](#)

此文档面向内核开发人员，包含在当前 Linux 内核树的“文档”目录中。其中探讨了涉及启用 NVDIMM 的不同内核模块，列出了有关内核实现的技术细节，并介绍了 [ndctl](#) 工具使用的内核接口 [sysfs](#)。

- [GitHub: pmem/ndctl \(<https://github.com/pmem/ndctl>\)](#)

用于管理 Linux 内核中的 [libnvdimm](#) 子系统的实用程序库。另外还包含用户空间库，以及单元测试和文档。

## V 服务

38 使用 YaST 进行服务管理 471

39 使用 NTP 同步时间 473

## 38 使用 YaST 进行服务管理

YaST 提供了一个服务管理器，用于控制默认系统目标和服务、显示服务状态，以及读取日志文件。SUSE Linux Enterprise Desktop 15 SP5 中的一项新功能是，YaST 支持 `systemd` 基于套接字的服务激活功能，此功能会将服务配置为按需启动。

`systemd` 支持使用基于套接字的激活来启动服务，适用于按需启动服务。这些服务有两种单元类型：服务和套接字。例如，CUPS 由 `cups.service` 和 `cups.socket` 控制。YaST 允许您选择要使用的服务启动类型。

图 38.1 “YaST 服务管理器” 显示了“启动模式”下拉框中的选项：引导时、按需和手动。对于基于套接字的激活，请选择按需。如此会打开一个侦听网络套接字，并在收到请求时启动服务。

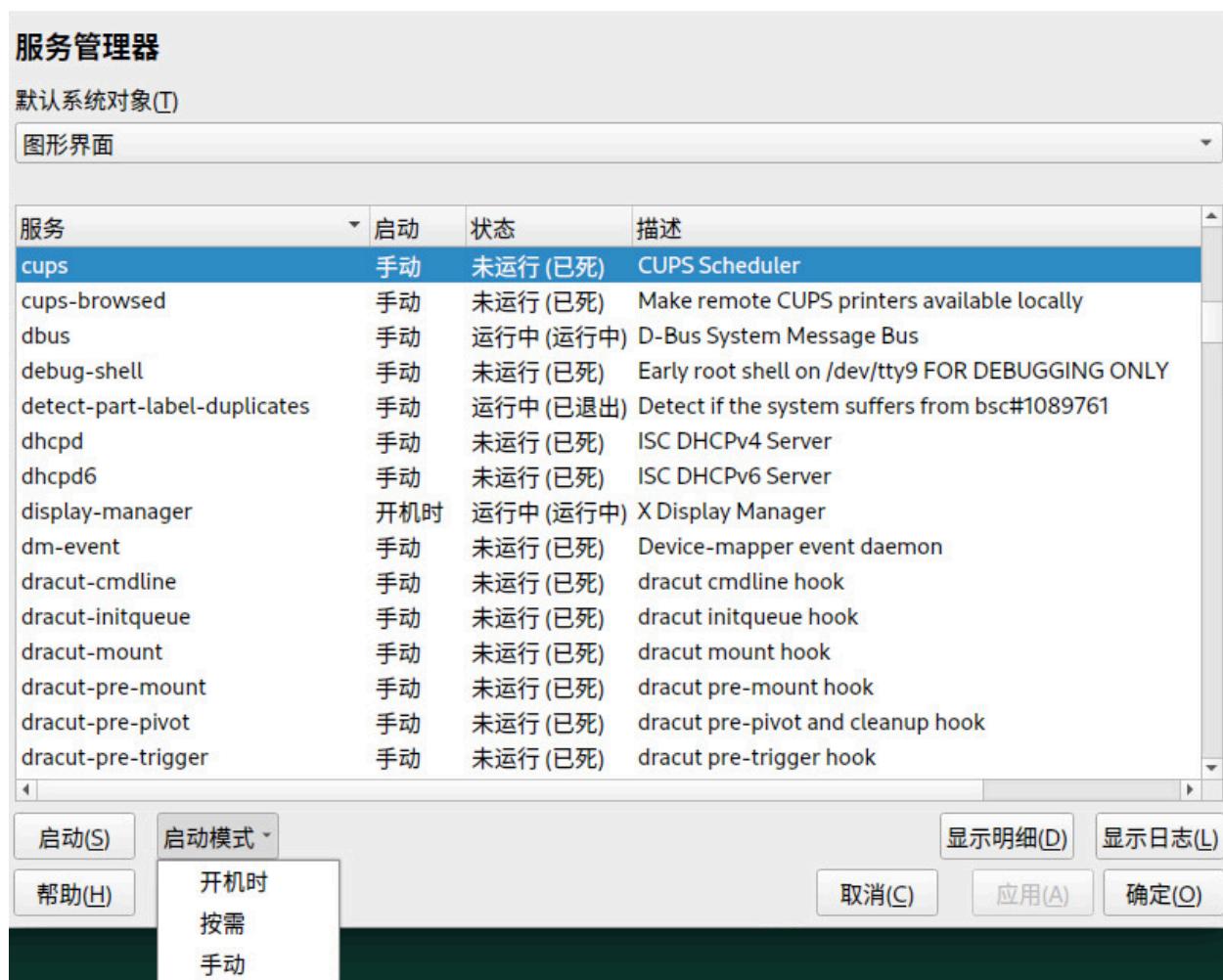


图 38.1：YAST 服务管理器

按需选项只会对支持它的服务显示。目前只有一小部分服务支持此选项，如 CUPS、dbus、iscsid、iscsiuio、multipathd、pcscd、rpcbind、TFTP、virtlockd、virtlogd。有关套接字激活工作原理的详细信息，请参见 [man 5 systemd.socket](#)。

## 39 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。内置硬件时钟往往不能满足数据库或群集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。NTP 服务通过网络中的可靠时间服务器持续调整系统时间。它还支持对本地参考时钟（如无线电控制的时钟）进行管理。

自 SUSE Linux Enterprise Desktop 15 起，采用 `chrony` 作为 NTP 的默认实施。`chrony` 包含两个部分：`chronyd` 守护程序可在引导时启动，`chronyc` 命令行界面程序用于监视 `chronyd` 的性能，以及在运行时更改各种操作参数。

从 SUSE Linux Enterprise Desktop 15.2 开始，如果不是配置为以守护程序的形式运行，适用于 NTP 客户端配置的 YaST 模块将配置 `systemd-timer` 而不是 `cron` 守护程序来执行 `chrony`。



### 注意

要通过 Active Directory 启用时间同步，请按照《安全和强化指南》，第 7 章“Active Directory 支持”，第 7.3.3 节“使用 Windows 域成员资格加入 Active Directory”，使用 Windows 域成员资格加入 Active Directory 域中的说明操作。

### 39.1 使用 YaST 配置 NTP 客户端

`chrony` 软件包中附带的 NTP 守护程序 (`chronyd`) 预先设置为使用本地计算机硬件时钟作为时间参考。硬件时钟的精确度极大程度上取决于它的时间源。例如，原子时钟或 GPS 接收器是非常精确的时间源，而一般 RTC 芯片则不是可靠的时间源。YaST 简化了 NTP 客户端的配置。

在 YaST NTP 客户端配置（网络服务，NTP 配置）窗口中，您可以指定启动 NTP 守护程序的时间、配置源的类型，以及添加自定义的时间服务器。



图 39.1：NTP 配置窗口

### 39.1.1 NTP 守护程序启动

您可以从下面三个选项中选择启动 NTP 守护程序的时间：

#### 仅手动

选择仅手动可手动启动 chrony 守护程序。

#### 不用守护进程同步

选择不用守护进程同步可定期设置系统时间，而不用永久运行 chrony。您可以设置同步间隔（以分钟计）。

#### 立即和在引导时

选择立即和在引导时可在引导系统时自动启动 chrony。建议使用此设置。

### 39.1.2 配置源的类型

在配置源下拉框中，选择动态或静态。如果服务器只使用固定的一组（公共）NTP 服务器，请设置静态；如果您的内部网络通过 DHCP 提供 NTP 服务器，则使用动态更合适。

### 39.1.3 配置时间服务器

供客户端查询的时间服务器列在 NTP 配置窗口的下半部分。使用添加、编辑和删除可按需修改此列表。

单击添加可添加新的时间服务器：



图 39.2：添加时间服务器

1. 在地址字段中，键入要与计算机时间同步的时间服务器或时间服务器池的 URL。填写好 URL 后，单击测试以校验该 URL 是否指向有效的时间源。
2. 激活快速初始同步，以便在 `chrony` 守护程序启动时通过发送更多请求来加速时间同步。
3. 激活离线启动，以使自动启动 `chrony` 守护程序且引导时可能没有互联网连接的系统加快引导速度。例如，对于由 NetworkManager 管理网络连接的笔记本电脑，此选项非常实用。
4. 单击确定进行确认。

## 39.2 手动配置网络中的 NTP

`chrony` 从 `/etc/chrony.conf` 文件读取其配置。要让计算机时钟保持同步，您需要告诉 `chrony` 使用什么时间服务器。您可以使用特定的服务器名称或 IP 地址，例如：

```
0.suse.pool.ntp.org
1.suse.pool.ntp.org
2.suse.pool.ntp.org
3.suse.pool.ntp.org
```

还可以指定池名称。池名称会解析为若干个 IP 地址：

```
pool pool.ntp.org
```



### 提示：同一网络中的多台计算机

要同步同一网络中的多台计算机的时间，建议不要通过一台外部服务器同步所有计算机。比较好的做法是将其中一台计算机作为时间服务器（它与外部时间服务器同步），其他计算机作为它的客户端。将 `local` 指令添加至服务器的 `/etc/chrony.conf`，以将其与权威时间服务器区分开：

```
local stratum 10
```

要启动 `chrony`，请运行：

```
systemctl start chronyd.service
```

初始化 `chronyd` 后，系统需要花费一些时间来稳定时间，以及创建用于纠正本地计算机时钟的偏移文件。利用偏移文件，当计算机启动时，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。

要启用 `chrony` 以便该服务在引导时自动启动，请运行：

```
systemctl enable chronyd.service
```



## 警告：冲突的 `yast-timesync.service` 服务

除了 `chronyd.service` 服务之外，SLED 还包含 `yast-timesync.service`。`yast-timesync.service` 由计时器每 5 分钟触发一次，该服务会运行 `chronyd`（使用 `-q` 选项）来设置系统时间，之后会退出。由于同一时间只能运行 `chronyd` 的一个实例，因此请勿同时启用或启动两个 `chronyd` 相关的服务。

## 39.3 在运行时使用 `chronyc` 配置 `chronyd`

在运行时，您可以使用 `chronyc` 来更改 `chronyd` 的行为。它会生成有关 `chronyd` 的操作的状态报告。

您可以采用交互模式或非交互模式运行 `chronyc`。要以交互模式运行 `chronyc`，请在命令行上输入 `chronyc`。如此会显示提示符并等待您输入命令。例如，要检查有多少个 NTP 源联机或脱机，请运行：

```
# chronyc
chronyc> activity
200 OK
4 sources online
2 sources offline
1 sources doing burst (return to online)
1 sources doing burst (return to offline)
0 sources with unknown address
```

要退出 `chronyc` 的提示符，请输入 `quit` 或 `exit`。

如果您不需要使用交互提示符，请直接输入命令：

```
# chronyc activity
```



## 注意：临时更改

使用 `chronyc` 进行的更改不是永久性的。当 `chronyd` 下次重启动时，它们将会丢失。要进行永久更改，请修改 `/etc/chrony.conf`。

有关 **chronyc** 命令的完整列表, 请参见它的手册页 ([man 1 chronyc](#))。

## 39.4 运行时的动态时间同步

尽管 **chronyd** 通常在引导时没有网络连接的系统上启动, 但该工具无法解析配置文件中指定的时间服务器的 DNS 名称。

**chronyd** 会按递增的时间间隔不断尝试解析由 **server**、**pool** 和 **peer** 指令指定的时间服务器名称, 直到成功为止。

如果启动 **chronyd** 时将无法访问时间服务器, 您可以指定 **offline** 选项:

```
server server_address offline
```

如此, **chronyd** 将不会尝试轮询服务器, 除非使用以下命令启用它:

```
# chronyc online server_address
```

如果设置了 **auto\_offline** 选项, 在向时间服务器发送了两个请求, 但未接收到任何响应时, **chronyd** 会假设时间服务器已脱机。此选项使 **chronyc** 在网络链路断开时, 无需运行“脱机”命令。

## 39.5 设置本地参考时钟

软件包 **chrony** 依赖于其他程序 (例如 **gpsd**) 来通过 SHM 或 SOCK 驱动程序访问计时数据。在 **/etc/chrony.conf** 中使用 **refclock** 指令可指定要用作时间源的硬件参考时钟。它有两个必填参数: 驱动程序名称和驱动程序特定的参数。这两个参数后面跟着零个或多个 **refclock** 选项。**chronyd** 包含以下驱动程序:

- PPS: 内核 **pulse per second** API 的驱动程序。例如:

```
refclock PPS /dev/pps0 lock NMEA refid GPS
```

- SHM: NTP 共享内存驱动程序。例如:

```
refclock SHM 0 poll 3 refid GPS1
```

```
refclock SHM 1:perm=0644 refid GPS2
```

- SOCK: Unix 域套接字驱动程序。例如:

```
refclock SOCK /var/run/chrony.ttyS0.sock
```

- PHC: PTP 硬件时钟驱动程序。例如:

```
refclock PHC /dev/ptp0 poll 0 dpoll -2 offset -37
refclock PHC /dev/ptp1:nocrosssts poll 3 pps
```

有关各个驱动程序的选项的详细信息, 请参见 [man 8 chrony.conf](#)。

## 39.6 与外部时间参考 (ETR) 的时钟同步

支持与外部时间参考 (ETR) 时钟同步。外部时间参照每  $2^{20}$  (2 的 20 次幂) 微秒发出一次振荡器信号和同步信号, 以将连接的所有服务器的 TOD 时钟保持同步。

两个 ETR 计算机可以连接到一台计算机。如果时钟偏差超过同步检查容差, 将会对所有 CPU 执行一次计算机检查, 这表示时钟未同步。如果发生这种情况, 在该时钟再次同步前, 所有与支持 XRC 的设备进行的 DASD I/O 都将停止。

可通过两个 sysfs 属性激活 ETR 支持; 请以 root 身份运行以下命令:

```
echo 1 > /sys/devices/system/etr/etr0/online
echo 1 > /sys/devices/system/etr/etr1/online
```

# VI 查错

- 40 帮助和文档 481
- 41 收集系统信息以供支持所用 486
- 42 常见问题及其解决方案 517

# 40 帮助和文档

SUSE® Linux Enterprise Desktop 提供了各种来源的信息和文档，其中有许多已经集成在您安装好的系统中。

## /usr/share/doc 中的文档

这是传统的帮助目录，包含系统的文档文件以及发行说明。它还包含子目录 `packages` 中的已安装软件包的信息。有关详细信息可以在[第 40.1 节 “文档目录”](#) 中找到。

## 外壳命令的手册页和信息页

使用外壳时，您不需要了解命令选项。外壳往往是通过手册页和信息页来提供集成帮助的。有关详细信息，请参见[第 40.2 节 “手册页”](#) 和[第 40.3 节 “信息页”](#)。

## 桌面帮助中心

GNOME 桌面的帮助中心（帮助）按可搜索的形式提供了对系统上最重要的文档资源的集中访问途径。这些资源包括已安装应用程序、手册页、信息页以及随产品提供的 SUSE 手册的联机帮助。

## 特定应用程序的独立帮助软件包

当用 YaST 安装新软件时，系统通常会自动安装软件文档，并且这些文档会出现在您桌面的帮助中心中。但是，某些应用程序（如 GIMP）可能具有不同的联机帮助软件包，可与 YaST 分开安装，并且不集成到帮助中心。

## 40.1 文档目录

在安装的 Linux 系统上用来查找文档的目录一直以来都是 `/usr/share/doc`。目录包含系统上所安装软件包的相关信息，以及发行说明和手册等。



### 注意：内容取决于所安装的软件包

在 Linux 系统中，许多手册和其他种类的文档都可以像软件一样以软件包的形式获取。`/usr/share/docs` 中的信息量和信息内容还取决于安装的（文档）软件包。如果您找不到这里提到的子目录，请检查相应的软件包是否已安装到您的系统上，并根据需要使用 YaST 添加它们。

## 40.1.1 SUSE 手册

我们以不同的语言提供了这些手册的 HTML 和 PDF 版本。在 `manual` 子目录下有您的产品适用的大多数 HTML 版 SUSE 手册。有关对您的产品可用的所有文档的概述，请参见这些手册的前言。

如果安装了多种语言，`/usr/share/doc/manual` 可能包含这些手册的不同语言版本。两个桌面的帮助中心内也提供 HTML 版本的 SUSE 手册。有关在安装媒体的何处能找到这些手册的 PDF 和 HTML 版本的信息，请参见 SUSE Linux Enterprise Desktop 发行说明。它们位于所安装系统的 `/usr/share/doc/release-notes/` 下，您也可以访问 <https://www.suse.com/releasenotes/> 上具体产品的网页，找到它们的联机版本。

## 40.1.2 软件包文档

在 `packages` 下可找到系统上安装的软件包中包含的文档。对每个软件包，都会创建子目录 `/usr/share/doc/packages/PACKAGENAME`。它经常包含该软件包的自述文件，有时还包含示例、配置文件或其他脚本。下表列出了 `/usr/share/doc/packages` 下常见的文件。以下每项不一定都存在，许多软件包中只包含其中的一部分。

### AUTHORS

主要开发者列表。

### BUGS

已知 bug 或故障。还可能包含到 Bugzilla 网页的链接，您可以在该页面上搜索所有 bug。

### CHANGES ,

### ChangeLog

每个版本的更改摘要。因为它很详细，所以对于开发人员而言很有帮助。

### COPYING ,

### LICENSE

许可信息。

### FAQ

从邮件列表或新闻组收集的问题和回答。

## INSTALL

如何在系统上安装此软件包。因为您读到该文件前该软件包已安装，您可以放心地忽略该文件的内容。

## README, README.\*

软件的常规信息。例如用途和用法。

## TODO

尚未实施但将来可能会实施的功能。

## MANIFEST

带有简述的文件列表。

## NEWS

描述此版本中的新增内容。

## 40.2 手册页

手册页是任何 Linux 系统的基本组成部分。它们介绍命令的用法以及所有可用的选项和参数。

许多页面都可以用 `man` 后面跟命令名来访问，例如 `man ls`。

手册页直接显示在外壳中。可以用 `Page ↑` 和 `Page ↓` 上下移动来浏览它们。用 `Home` 和 `End` 可以切换显示文档的开头和结尾。按 `Q` 可以结束这种查看模式。通过 `man man` 了解有关 `man` 命令本身的更多信息。手册页如表 40.1 “手册页 - 类别和说明” 所示按类别进行排序（取自 `man` 命令本身的手册页）。

表 40.1：手册页 - 类别和说明

| 数字 | 说明                            |
|----|-------------------------------|
| 1  | 可执行程序或外壳命令                    |
| 2  | 系统调用（内核提供的函数）                 |
| 3  | 库调用（程序库内的函数）                  |
| 4  | 特殊文件（通常位于 <code>/dev</code> ） |

| 数字 | 说明  |
|----|---|
| 5  | 文件格式和约定 ( <code>/etc/fstab</code> )                               |
| 6  | 游戏  |
| 7  | 其他 (包括宏软件包和约定) , 如<br><code>man(7)</code> 、 <code>groff(7)</code> |
| 8  | 系统管理命令 (通常仅供 <code>root</code> 使用)                                |
| 9  | 内核例程 (非标准)  |

每个手册页由以下几个部分组成：NAME、SYNOPSIS、DESCRIPTION、SEE ALSO、LICENSING 和 AUTHOR。根据具体的命令类型，可能还有其他部分。

## 40.3 信息页

信息页是系统上另一个重要的信息来源。它们通常比手册页更为详细。它们包含多个命令行选项，有时还包含完整的教学课程或参考文档。要查看特定命令的信息页，请输入 `info` 后面跟命令名，例如 `info ls`。您可以在外壳中直接用查看器浏览信息页，并显示名为“节点”的各个部分。使用 `Space` 可前移，使用 `<-` 可后移。在节点内，您也可以用 `Page ↑` 和 `Page ↓` 浏览，但只有 `Space` 和 `<-` 仍会带您到上个或下个节点。按 `Q` 结束查看模式。并非每个命令都有对应的信息页，反之亦然。

## 40.4 联机资源

有关 SUSE Linux Enterprise Desktop 所有可用文档的概述，请查看具体产品的文档网页，网址为：<https://documentation.suse.com/>。

如果要搜索更多产品相关信息，您也可以参见以下网站：

### SUSE 技术支持

如果您有技术方面的疑问或需要相应的解决方案，可在 <https://www.suse.com/support/> 上找到 SUSE 技术支持。

## SUSE Linux Enterprise Desktop 用户社区

SUSE and Rancher Community (<https://community.suse.com/>) ↗

## SUSE 博客

SUSE 博客提供了相关文章、提示和问答：<https://www.suse.com/c/blog/> ↗

## GNOME 文档

适用于 GNOME 用户、管理员和开发人员的文档可在 <https://help.gnome.org/> ↗ 上找到。

## Linux 文档计划

Linux 文档计划 (Linux Documentation Project, TLD) 由编写 Linux 相关文档的志愿者团队负责管理 (请参见 <https://www.tldp.org> ↗)。它可能是 Linux 相关的最全面的文档资源。这套文档包括初学者教程，但主要侧重于有经验的用户和职业系统管理员。TLD 以免费许可的形式发布 HOWTO、常见问题和指南 (手册)。SUSE Linux Enterprise Desktop 上也有来自 TLD 的部分文档。

您还可以尝试通用搜索引擎。例如，如果在刻录 CD 或转换 LibreOffice 文件时遇到问题，可以使用搜索词 Linux CD-RW help 或 OpenOffice file conversion problem。

# 41 收集系统信息以供支持所用

为了让用户快速概览计算机的所有系统相关信息，SUSE Linux Enterprise Desktop 提供了 `hostinfo` 软件包。该软件包还可以帮助系统管理员检查污染的（不支持的）内核，或者计算机上安装的任何第三方软件包。

出现问题时，可以使用 `supportconfig` 命令行工具或 YaST 支持模块创建详细的系统报告。这两种方法都会收集系统的相关信息，包括当前内核版本、硬件、已安装软件包、分区设置及其他信息。最后会生成一个包含多个文件的 TAR 存档。在建立服务请求 (SR) 后，您可以将该 TAR 存档上载至全球技术支持。该存档有助于找出您报告的问题，并帮助您解决问题。

此外，您可以分析 `supportconfig` 输出来发现已知问题，以帮助快速解决问题。为此，SUSE Linux Enterprise Desktop 提供了一个设备和一个命令行工具用于进行 `Supportconfig Analysis (SCA)`。

## 41.1 显示当前系统信息

在登录到服务器时，要想快速方便地概览所有相关系统信息，请使用软件包 `hostinfo`。在计算机上安装该软件包后，控制台将会向登录到该计算机的任何 `root` 用户显示以下信息：

### 例 41.1：以 `root` 身份登录时的 `hostinfo` 输出

```
Welcome to SUSE Linux Enterprise Server 15 SP2 Snapshot8 (x86_64) - Kernel \r\n(\\"l).\n\nDistribution: SUSE Linux Enterprise Server 15 SP2\nCurrent As Of: Wed 25 Mar 2020 12:09:20 PM PDT\nHostname: localhost\nKernel Version: 5.3.18-8-default\nArchitecture: x86_64\nInstalled: Thu 19 Mar 2020 11:25:13 AM PDT
```

```
Status: Not Tainted
Last Installed Package: Wed 25 Mar 2020 11:42:24 AM PDT
Patches Needed: 0
Security: 0
3rd Party Packages: 219
Network Interfaces
  eth0: 192.168.2/24 2002:c0a8:20a::/64
Memory
  Total/Free/Avail: 7.4Gi/6.4Gi/6.8Gi (91% Avail)
CPU Load Average: 7 (3%) with 2 CPUs
```

如果输出显示 tainted 内核状态, 请参见第 41.6 节 “内核模块支持” 以了解更多细节。

## 41.2 使用 Supportconfig 收集系统信息

要创建包含详细系统信息的 TAR 存档, 以提交给全球技术支持, 请使用:

- supportconfig 命令, 或者
- YaST 支持模块。

该命令行工具由默认安装的软件包 supportutils 提供。YaST 支持模块也基于该命令行工具。

系统上的有些软件包集成了 Supportconfig 插件, 具体视系统上安装的软件包而定。执行 Supportconfig 时, 还会执行所有插件, 并创建该存档的一个或多个结果文件。这样做的好处是只检查包含特定插件的主题。Supportconfig 插件存储在 /usr/lib/supportconfig/plugins/ 目录中。

### 41.2.1 创建服务请求编号

系统随时都可以生成 Supportconfig 存档。但是, 要将 Supportconfig 数据提交给全球技术支持, 首先需要生成一个服务请求编号。上载存档以获取支持时, 需要使用此编号。

要创建服务请求, 请访问 <https://scc.suse.com/support/requests> 并遵照屏幕上的说明操作。记下服务请求编号。



## 注意：隐私声明

SUSE 将系统报告视为机密数据。有关我们在隐私方面所做承诺的详细信息，请参见 <https://www.suse.com/company/policies/privacy/>。

### 41.2.2 上载目标

在创建服务请求编号后，可以根据[过程 41.1 “使用 YaST 向支持部门提交信息”](#)或[过程 41.2 “从命令行向支持部门提交信息”](#)中所述将 Supportconfig 存档上载到全球技术支持站点。使用下列上载目标之一：

- 北美：FTP <ftp://support-ftp.us.suse.com/incoming/>、FTPS <https://support-ftp.us.suse.com/incoming/>
- EMEA（欧洲、中东和非洲）：FTP <ftp://support-ftp.emea.suse.com/incoming/>、FTPS <https://support-ftp.emea.suse.com/incoming/>

或者，可以使用以下服务请求 URL 手动将该 TAR 存档附加到您的服务请求：<https://scc.suse.com/support/requests>。

### 41.2.3 使用 YaST 创建 supportconfig 存档

要使用 YaST 收集系统信息，请执行以下步骤：

1. 启动 YaST 并打开支持模块。



2. 单击创建报告 tarball。
3. 在随后出现的窗口中，从单选按钮列表中选择一个 Supportconfig 选项。系统默认会预先选择使用自定义（专家）设置。如果要先测试报告功能，请使用仅收集最少量的信息。有关其他选项的更多信息，请参考 [supportconfig](#) 手册页。  
单击下一步。
4. 输入您的联系信息。此信息将保存在 `basic-environment.txt` 文件中，并会包含在创建的存档中。
5. 要将存档提交到全球技术支持，请提供所需的上载信息。YaST 会自动推荐一个上载服务器。要修改该服务器，请参考[第 41.2.2 节 “上载目标”](#)，以详细了解可以使用哪些上载服务器。  
要在以后再提交存档，请将上载信息留空。
6. 单击下一步启动信息收集过程。



该过程完成后，单击下一步。

7. 要查看收集的数据，请从文件名中选择所需的文件，以在 YaST 中查看其内容。在将 TAR 存档提交到支持部门之前，如果要去除该存档中的某个文件，请使用从数据中去除。按下一步。
8. 保存该 TAR 存档。如果您以 `root` 用户身份启动了 YaST 模块，YaST 会提示您将该存档保存到 `/var/log` (否则将保存到您的主目录)。文件名格式为 `scc_HOST_DATE_TIME.tbz`。
9. 要直接将该存档上载到支持部门，请确保将日志文件 tarball 上载到 URL 已激活。此处显示的上载目标是步骤 5 中 YaST 推荐的上载目标。要修改上载目标，请在第 41.2.2 节 “上载目标” 中检查有哪些可用的上载服务器。
10. 要跳过上载步骤，请停用将日志文件 tarball 上载到 URL。
11. 确认更改以关闭 YaST 模块。

## 41.2.4 从命令行创建 supportconfig 存档

下面的过程显示了如何创建 Supportconfig 存档，但不将它直接提交给支持人员。要上载该存档，需要根据[过程 41.2 “从命令行向支持部门提交信息”](#) 中所述，结合某些选项运行命令。

1. 打开外壳并转换为 root 用户。
2. 运行 **supportconfig**。一般情况下，运行此工具时可以不使用任何选项。不过，也可以使用以下最常见的选项：

-E MAIL,  
-N NAME,  
-O COMPANY,  
-P PHONE

设置您的联系数据：电子邮件地址 (-E)、公司名称 (-O)、您的姓名 (-N) 和电话号码 (-P)。

-i KEYWORDS,  
-F

限制检查的功能。KEYWORDS 占位符是区分大小写的关键字的逗号分隔列表。使用 **supportconfig -F** 可获得所有关键字列表。

-r SRNUMBER

定义上载生成的 TAR 存档时所用的服务请求编号。

3. 等待工具完成操作。
4. 默认的存档位置为 /var/log，文件名格式为 scc\_HOST\_DATE\_TIME.tbz

## 41.2.5 了解 supportconfig 的输出

不管您是通过 YaST 还是直接运行 **supportconfig**，该脚本都会显示其已执行操作的摘要。

```
Support Utilities - Supportconfig
Script Version: 3.0-98
Script Date: 2017 06 01
```

```

[...]
Gathering system information

  Data Directory:      /var/log/scc_d251_180201_1525 ①

  Basic Server Health Check...          Done ②
  RPM Database...                      Done ②
  Basic Environment...                Done ②
  System Modules...                  Done ②
[...]
  File System List...                Skipped ③
[...]
  Command History...                Excluded ④
[...]
  Supportconfig Plugins:           1 ⑤
    Plugin: pstree...
[...]
Creating Tar Ball

==[ DONE ]=====
  Log file tar ball: /var/log/scc_d251_180201_1525.txz ⑥
  Log file size:      732K
  Log file md5sum:    bf23e0e15e9382c49f92cbce46000d8b
=====
```

- ① 用于存储结果的临时数据目录。此目录存档为 tar 文件，请参见 ⑥。
- ② 已成功启用（默认或手动选择）并执行该功能。结果存储在文件中（请参见表 41.1 “TAR 存档中的功能和文件名对照”）。
- ③ 已跳过该功能，因为一个或多个 RPM 软件包的文件发生了更改。
- ④ 已排除该功能，因为已通过 `-x` 选项取消选择该功能。
- ⑤ 该脚本发现了一个插件，并执行该插件 `pstree`。该插件位于 `/usr/lib/supportconfig/plugins/` 目录中。有关细节，请参见手册页。
- ⑥ 存档的 Tar 文件名，默认使用 `xz` 压缩。

## 41.2.6 常用的 supportconfig 选项

**supportconfig** 实用程序在调用时通常不带任何选项。使用 **supportconfig -h** 显示所有选项的列表或参见手册页。下面简要列出了常见的使用情形：

### 减少所收集信息的大小

使用最少量选项 (**-m**)：

```
> sudo supportconfig -m
```

### 将信息限制为特定的主题

如果已确定问题所在，并发现该问题只与特定的区域或功能集相关，则您在下次运行 **supportconfig** 时，应限制针对特定的区域收集信息。例如，如果您检测到 LVM 出现问题，想要测试最近对 LVM 配置所做的更改。在这种情况下，合适的做法是只收集有关 LVM 的最少量 Supportconfig 信息：

```
> sudo supportconfig -i LVM
```

可通过逗号分隔额外的关键字。例如，额外的磁盘测试：

```
> sudo supportconfig -i LVM,DISK
```

要查看可用于将所收集信息限制为特定区域的功能关键字的完整列表，请运行：

```
> sudo supportconfig -F
```

### 在输出中包含其他联系信息：

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

(在一行中输入所有命令)

### 收集已经过轮转的日志文件

```
> sudo supportconfig -l
```

这对日志记录量较大的环境，或重引导后 syslog 轮换日志文件时发生内核崩溃的情况尤为有用。

## 41.2.7 存档内容概述

TAR 存档包含各功能的所有结果。根据所选的内容（所有功能或只有一小部分功能），存档包含的文件可能有多有少。您可通过 `-i` 选项来限制功能集（请参见[第 41.2.6 节 “常用的 supportconfig 选项”](#)）。

要列出存档的内容，请使用以下 `tar` 命令：

```
# tar xf /var/log/scc_earth_180131_1545.tbz
```

TAR 存档中始终包含以下文件名：

### 存档中一定会包含的文件

#### basic-environment.txt

包含执行此脚本的日期和系统信息（如发行套件版本）、超级管理程序信息等。

#### basic-health-check.txt

包含基本的运行状态检查，如运行时间、虚拟内存统计数字、可用内存和硬盘、僵尸进程检查等。

#### hardware.txt

包含基本硬件检查（如有关 CPU 体系结构的信息）、所有已连接硬件的列表、中断、I/O 端口、内核引导消息等。

#### messages.txt

包含系统日志中的日志消息。

#### rpm.txt

包含所有已安装的 RPM 软件包、名称、来源及其版本的列表。

#### summary.xml

包含 XML 格式的信息，如发行套件、版本和产品特定的分段。

#### supportconfig.txt

包含有关 `supportconfig` 脚本本身的信息。

#### y2log.txt

包含 YaST 特定的信息，如特定的软件包、配置文件和日志文件。

表 41.1 “**TAR 存档中的功能和文件名对照**” 列出了所有可用的功能及其文件名。后续的服务包或插件将会补充到列表中。

表 41.1：**TAR 存档中的功能和文件名对照**

| 功能       | 文件名                          |
|----------|------------------------------|
| APPARMOR | <u>security-apparmor.txt</u> |
| AUDIT    | <u>security-audit.txt</u>    |
| AUTOFS   | <u>fs-autofs.txt</u>         |
| BOOT     | <u>boot.txt</u>              |
| BTRFS    | <u>fs-btrfs.txt</u>          |
| DAEMONS  | <u>systemd.txt</u>           |
| CIMOM    | <u>cimom.txt</u>             |
| CRASH    | <u>crash.txt</u>             |
| CRON     | <u>cron.txt</u>              |
| DHCP     | <u>dhcp.txt</u>              |
| DISK     | <u>fs-diskio.txt</u>         |
| DNS      | <u>dns.txt</u>               |
| DOCKER   | <u>docker.txt</u>            |
| DRBD     | <u>drbd.txt</u>              |
| ENV      | <u>env.txt</u>               |
| ETC      | <u>etctxt</u>                |
| HA       | <u>ha.txt</u>                |
| HAPROXY  | <u>haproxy.txt</u>           |

| 功能               | 文件名                         |
|------------------|-----------------------------|
| <u>HISTORY</u>   | <u>shell_history.txt</u>    |
| <u>IB</u>        | <u>ib.txt</u>               |
| <u>IMAN</u>      | <u>novell-iman.txt</u>      |
| <u>ISCSI</u>     | <u>fs-iscsi.txt</u>         |
| <u>LDAP</u>      | <u>ldap.txt</u>             |
| <u>LIVEPATCH</u> | <u>kernel-livepatch.txt</u> |
| <u>LVM</u>       | <u>lvm.txt</u>              |
| <u>MEM</u>       | <u>memory.txt</u>           |
| <u>MOD</u>       | <u>modules.txt</u>          |
| <u>MPIO</u>      | <u>mpio.txt</u>             |
| <u>NET</u>       | <u>network-*.txt</u>        |
| <u>NFS</u>       | <u>nfs.txt</u>              |
| <u>NTP</u>       | <u>ntp.txt</u>              |
| <u>NVME</u>      | <u>nvme.txt</u>             |
| <u>OCFS2</u>     | <u>ocfs2.txt</u>            |
| <u>OFILES</u>    | <u>open-files.txt</u>       |
| <u>PRINT</u>     | <u>print.txt</u>            |
| <u>PROC</u>      | <u>proc.txt</u>             |
| <u>SAR</u>       | <u>sar.txt</u>              |
| <u>SLERT</u>     | <u>slert.txt</u>            |
| <u>SLP</u>       | <u>slp.txt</u>              |

| 功能                   | 文件名                             |
|----------------------|---------------------------------|
| <u>SMT</u>           | <u>smt.txt</u>                  |
| <u>SMART</u>         | <u>fs-smartmon.txt</u>          |
| <u>SMB</u>           | <u>samba.txt</u>                |
| <u>SRAID</u>         | <u>fs-softraid.txt</u>          |
| <u>SSH</u>           | <u>ssh.txt</u>                  |
| <u>SSSD</u>          | <u>sssd.txt</u>                 |
| <u>SYSCONFIG</u>     | <u>sysconfig.txt</u>            |
| <u>SYSFS</u>         | <u>sysfs.txt</u>                |
| <u>TRANSACTIONAL</u> | <u>transactional-update.txt</u> |
| <u>TUNED</u>         | <u>tuned.txt</u>                |
| <u>UDEV</u>          | <u>udev.txt</u>                 |
| <u>UFILES</u>        | <u>fs-files-additional.txt</u>  |
| <u>UP</u>            | <u>updates.txt</u>              |
| <u>WEB</u>           | <u>web.txt</u>                  |
| <u>X</u>             | <u>x.txt</u>                    |

## 41.3 将信息提交到全球技术支持部门

可以使用 YaST 支持模块或 **supportconfig** 命令行实用程序向全球技术支持提交系统信息。如果您遇到服务器问题并想要获得支持人员的帮助，则首先需要建立一个服务请求。有关详细信息，请参见第 41.2.1 节 “创建服务请求编号”。

以下示例使用 12345678901 作为服务请求编号的占位符。请将 12345678901 替换为您在第 41.2.1 节 “创建服务请求编号” 中创建的服务请求编号。

#### 过程 41.1：使用 YaST 向支持部门提交信息

下面的过程假设您已创建某个 Supportconfig 存档，但尚未将其上载。请确保已按第 41.2.3 节 “使用 YaST 创建 supportconfig 存档” 中的步骤 4 所述在存档中包含了您的联系信息。有关如何一步生成并提交 Supportconfig 存档的说明，请参见第 41.2.3 节 “使用 YaST 创建 supportconfig 存档”。

1. 启动 YaST 并打开支持模块。
2. 单击上载。
3. 在有日志文件的软件包中，指定现有 Supportconfig 存档的路径，或者单击浏览找到该存档。
4. YaST 会自动推荐一个上载服务器。要修改该服务器，请参考第 41.2.2 节 “上载目标”，以详细了解可以使用哪些上载服务器。



单击下一步继续。

5. 单击完成。

#### 过程 41.2：从命令行向支持部门提交信息

下面的过程假设您已创建某个 Supportconfig 存档，但尚未将其上载。有关如何一步生成并提交 Supportconfig 存档的说明，请参见第 41.2.3 节 “使用 YaST 创建 supportconfig 存档”。

## 1. 具有互联网连接的服务器：

- a. 要使用默认上载目标，请运行：

```
> sudo supportconfig -ur 12345678901
```

- b. 对于安全上载目标，请使用以下命令：

```
> sudo supportconfig -ar 12345678901
```

## 2. 不具有互联网连接的服务器：

- a. 运行以下命令：

```
> sudo supportconfig -r 12345678901
```

- b. 将 /var/log/scc\_SR12345678901\*tbz 存档手动上载到我们的其中一个 FTP 服务器。要使用哪个服务器取决于您所在的位置。有关概览，请参见[第 41.2.2 节 “上载目标”](#)。

## 3. 将 TAR 存档上载到我们 FTP 服务器的接收目录后，它会自动附加到您的服务请求中。

## 41.4 分析系统信息

您可以分析使用 **supportconfig** 创建的系统报告来发现已知问题，以帮助快速解决问题。为此，SUSE Linux Enterprise Desktop 提供了一个设备和一个命令行工具用于进行 **Supportconfig Analysis (SCA)**。SCA 设备是一个非交互式服务器端工具。SCA 工具 (sca-server-report 软件包提供的 **scatool**) 在客户端运行，并从命令行执行。这两个工具都能分析来自受影响服务器的 Supportconfig 存档。初始服务器分析在 SCA 设备或运行 **scatool** 的工作站上进行。生产服务器上不发生任何分析周期。

此外，该设备和命令行工具还需要产品特定的模式，这样它们才能分析关联产品的 Supportconfig 输出。每个模式就是一个脚本，用于针对某个已知问题分析和评估 Supportconfig 存档。模式以 RPM 软件包的形式提供。

您也可以根据[第 41.4.3 节 “开发自定义分析模式”](#) 中的简述开发自己的模式。

## 41.4.1 SCA 命令行工具

SCA 命令行工具可让您既可使用 **supportconfig** 又可使用本地计算机上安装的特定产品的分析模式来分析该本地计算机。该工具将创建一份 HTML 报告来显示分析结果。有关示例，请参见图 41.1 “SCA 工具生成的 HTML 报告”。

| Supportconfig Analysis Report              |  |   |  |
|--|--|---|--|
| <b>Server Information</b>                  |  |   |  |
| <b>Analysis Date:</b>                      | /4/2014 11:22                            | <b>Archive File:</b>  | /var/log/nts_barett-2_140425_1119.html |
| <b>Server Name:</b>                        | barett-2                                 | <b>Hardware:</b>  | Bochs                                  |
| <b>Distribution:</b>                       | SUSE Linux Enterprise Server 12 (x86_64) | <b>Service Pack:</b>  | 0                                      |
| <b>Hypervisor:</b>                         | KVM (QEMU Virtual CPU)                   | <b>Identity:</b>  | Virtual Machine (QEMU Virtual CPU)     |
| <b>Kernel Version:</b>                     | 3.12.14-1-default                        | <b>Supportconfig Version:</b>   | 3.0-18                                 |
| <b>Conditions Evaluated as Critical</b>    |  |   |  |
|  | <b>Category</b>                          | <b>Message</b>  | <b>Solutions</b>                       |
|  | Basic Health                             | 2 Basic Health Message(s)   |  |
|  | Basic Health SLE                         | Kernel Kernel Status -- Tainted: F O  | TID                                    |
|  | Basic Health SLE                         | System Last system down was not clean on Mon Mar 24 17:37:04 2014 and 1 additional failure(s) | TID TID1                               |
|  | SLE                                      | 2 SLE Message(s)  |  |
| <b>Conditions Evaluated as Warning</b>     |  |   |  |
|  | <b>Category</b>                          | <b>Message</b>  | <b>Solutions</b>                       |
|  | SLE                                      | 1 SLE Message(s)  |  |
| <b>Conditions Evaluated as Recommended</b> |  |   |  |
|  | <b>Category</b>                          | <b>Message</b>  | <b>Solutions</b>                       |
|  | SLE                                      | 1 SLE Message(s)  |  |
| <b>Conditions Evaluated as Success</b>     |  |   |  |
|  | <b>Category</b>                          | <b>Message</b>  | <b>Solutions</b>                       |
|  | Security                                 | 1 Security Message(s)   |  |
|  | Security SLE                             | AppArmor There are no AppArmor reject messages  | TID Doc                                |
|  | Basic Health                             | 8 Basic Health Message(s)   |  |
|  | Basic Health SLE                         | Kernel Context switches per second observed: 79   | TID                                    |
|  | Basic Health SLE                         | Kernel Interrupts per second observed: 51   | TID                                    |
|  | Basic Health SLE                         | CPU Utilization: 1.00%, Idle: 99.00%  | TID                                    |
|  | Basic Health SLE                         | Disk Mount on / has highest used space: 22%   | TID TID2                               |
|  | Basic Health SLE                         | Kernel 2% CPU load within limits. CPUs: 1, Load Average: 0.02                                 | TID Web Wikipedia                      |
|  | Basic Health SLE                         | Memory Memory used 29% - Swapping: No   | TID                                    |
|  | Basic Health SLE                         | Processes 0 Uninterruptible processes observed  | TID                                    |
|  | Basic Health SLE                         | Processes 0 Zombie processes observed   | TID                                    |

图 41.1：SCA 工具生成的 HTML 报告

**scatool** 命令由 **sca-server-report** 软件包提供。默认情况下不会安装该软件包。此外，您还需要 **sca-patterns-base** 软件包，以及满足以下条件的任何产品特定的 **sca-patterns-\*** 软件包：与您想要在其上运行 **scatool** 命令的计算机上安装的产品匹配。

以 **root** 用户身份或者结合 **sudo** 执行 **scatool** 命令。在调用 SCA 工具时，可以分析现有的 **supportconfig** TAR 存档，或者一步即让该工具生成并分析新的存档。该工具还提供具有 Tab 补全功能的交互控制台。您可以在外部计算机上运行 **supportconfig**，在本地计算机上执行后续分析。

下面提供了一些示例命令：

### **sudo scatool -s**

调用 **supportconfig**，并在本地计算机上生成新的 Supportconfig 存档。通过应用与所安装产品匹配的 SCA 分析模式来分析存档以发现已知问题。显示基于分析结果生成的 HTML 报告的路径。该报告将写入到 Supportconfig 存档所在的同一目录中。

### **sudo scatool -s -o /opt/sca/reports/**

与 **sudo scatool -s** 类似，唯一的差别在于，HTML 报告将写入到 **-o** 指定的路径。

### **sudo scatool -a PATH\_TO\_TARBALL\_OR\_DIR**

分析指定的 Supportconfig 存档文件（或者指定用于存放解压缩的 Supportconfig 存档的目录）。生成的 HTML 报告保存在 Supportconfig 存档或目录所在的位置。

### **sudo scatool -a SLES\_SERVER.COMPANY.COM**

与外部服务器 **SLES\_SERVER.COMPANY.COM** 建立 SSH 连接，并在该服务器上运行 **supportconfig**。Supportconfig 存档随后将复制回本地计算机，并在该处进行分析。生成的 HTML 报告保存在默认的 **/var/log** 目录中。（**SLES\_SERVER.COMPANY.COM** 上只创建 Supportconfig 存档）。

### **sudo scatool -c**

启动 **scatool** 的交互式控制台。按 **-I** 两次可查看可用命令。

有关其他选项和信息，请运行 **sudo scatool -h** 或参见 **scatool** 手册页。

## 41.4.2 SCA 设备

如果您决定使用 SCA 设备来分析 Supportconfig 存档，请配置一台服务器（或虚拟机）作为专用的 SCA 设备服务器。然后，可以使用该 SCA 设备服务器，对企业中运行 SUSE Linux Enterprise Server 或 SUSE Linux Enterprise Desktop 的所有计算机上的 Supportconfig 存档进行分析。您只需要将 Supportconfig 存档上载到设备服务器，等待它执行分析。此过程无需任何交互。在 MariaDB 数据库中，SCA 设备将会跟踪已分析的所有 Supportconfig 存档。您可以直接从设备 Web 界面阅读 SCA 报告。或者，可以让设备通过电子邮件将 HTML 报告发送给任何管理用户。有关详细信息，请参见第 41.4.2.5.4 节“通过电子邮件发送 SCA 报告”。

## 41.4.2.1 安装快速入门

要通过命令行快速安装和设置 SCA 设备, 请遵照此处的说明操作。该过程适用于专家用户, 主要通过单纯的安装与设置命令来完成。有关更多信息, 请参考[第 41.4.2.2 节 “先决条件”](#) 到[第 41.4.2.3 节 “安装和基本设置”](#) 中的详细说明。

### 先决条件

- Web 和 LAMP 模式
- Web 和脚本编写模块 (您必须注册计算机才能选择此模块)。



### 注意: 需要 root 特权

下面过程中的所有命令必须以 root 身份来运行。

#### 过程 41.3 : 使用匿名 FTP 进行上载的安装

设置并运行设备后, 无需任何人工交互。因此, 对于使用 cron 作业创建和上载 Supportconfig 存档的情形, 非常适合使用这种方法来设置设备。

1. 在要安装设备的计算机上, 登录到控制台并执行以下命令 (确保接受推荐的软件包) :

```
> sudo zypper install sca-appliance-* sca-patterns-* \
vsftpd yast2 yast2-ftp-server
> sudo systemctl enable apache2
> sudo systemctl start apache2
> sudo systemctl enable vsftpd
> sudo systemctl start vsftpd
> sudo yast ftp-server
```

2. 在 YaST FTP 服务器中, 选择身份验证 > 启用上载 > 允许匿名用户上载 > 完成 > 是, 以创建 /srv/ftp/upload。
3. 执行以下命令:

```
> sudo systemctl enable mysql
> sudo systemctl start mysql
> sudo mysql_secure_installation
> sudo setup-sca -f
```

`mysql_secure_installation` 将创建 MariaDB root 口令。

#### 过程 41.4：使用 SCP/TMP 进行上载的安装

这种设置设备的方法需要在键入 SSH 口令时进行人工交互。

1. 在安装设备的计算机上，登录到控制台。

2. 执行以下命令：

```
> sudo zypper install sca-appliance-* sca-patterns-*
> sudo systemctl enable apache2
> sudo systemctl start apache2
> sudo sudo systemctl enable mysql
> sudo systemctl start mysql
> sudo mysql_secure_installation
> sudo setup-sca
```

#### 41.4.2.2 先决条件

要运行 SCA 设备服务器，需要满足以下先决条件：

- 所有 sca-appliance-\* 软件包。
- sca-patterns-base 软件包。此外，需要针对您要使用该设备分析的 Supportconfig 存档类型安装任意产品特定的 sca-patterns-\*。
- Apache
- PHP
- MariaDB
- 匿名 FTP 服务器（可选）

#### 41.4.2.3 安装和基本设置

如第 41.4.2.2 节 “先决条件” 中所列，SCA 设备与其他软件包存在若干依赖项。因此，在安装和设置 SCA 设备服务器之前，需要做一些准备工作：

1. 对于 Apache 和 MariaDB, 请安装 Web 和 LAMP 安装软件集。
2. 设置 Apache 和 MariaDB, 并有选择性地设置一个匿名 FTP 服务器。
3. 将 Apache 和 MariaDB 配置为在引导时启动:

```
> sudo systemctl enable apache2 mysql
```

4. 启动这两个服务:

```
> sudo systemctl start apache2 mysql
```

现在, 您便可以根据[过程 41.5 “安装和配置 SCA 设备”](#) 中所述安装和设置 SCA 设备。

#### 过程 41.5 : 安装和配置 SCA 设备

安装这些软件包后, 可以使用 setup-sca 脚本来对 SCA 设备使用的 MariaDB 管理与报告数据库进行基本的配置。

使用该脚本可以配置以下选项, 以便将 Supportconfig 存档从您的计算机上载到 SCA 设备:

- scp
- 匿名 FTP 服务器

1. 安装设备和 SCA 基本模式库:

```
> sudo zypper install sca-appliance-* sca-patterns-base
```

2. 此外, 请针对您要分析的 Supportconfig 存档类型安装模式软件包。例如, 如果您的环境中安装了 SUSE Linux Enterprise Server 12 和 SUSE Linux Enterprise Server 15 服务器, 请安装 sca-patterns-sle12 和 sca-patterns-sle15 这两个软件包。

安装所有可用模式:

```
> sudo zypper install sca-patterns-*
```

3. 要对 SCA 设备进行基本的设置, 请使用 setup-sca 脚本。如何调用该命令取决于您要如何将 Supportconfig 存档上载到 SCA 设备服务器:

- 如果您配置了使用 `/srv/ftp/upload` 目录的匿名 FTP 服务器, 请结合 `-f` 选项执行设置脚本。按照屏幕上的说明操作:

```
> sudo setup-sca -f
```



## 注意: 使用其他目录的 FTP 服务器

如果 FTP 服务器使用的目录不是 `/srv/ftp/upload`, 请先调整以下配置文件, 使其指向正确的目录: `/etc/sca/sdagent.conf` 和 `/etc/sca/sdbroker.conf`。

- 要想通过 `scp` 将 `Supportconfig` 文件上载到 SCA 设备服务器的 `/tmp` 目录, 请不带任何参数调用该设置脚本。按照屏幕上的说明操作:

```
> sudo setup-sca
```

该设置脚本将会根据它的要求运行一些检查, 并配置所需的组件。它会提示您输入两个口令: 您设置的 MariaDB 的 MySQL `root` 口令, 以及用于登录 SCA 设备 Web 界面的 Web 用户口令。

- 输入现有的 MariaDB `root` 口令。SCA 设备将使用该口令连接到 MariaDB。
- 定义 Web 用户的口令。该口令将写入 `/srv/www/htdocs/sca/web-config.php`, 并设置为用户 `scdiag` 的口令。以后, 您可随时更改用户名和口令, 请参见[第 41.4.2.5.1 节“Web 界面的口令”](#)。

在成功完成安装和设置后, 便可以开始使用 SCA 设备, 请参见[第 41.4.2.4 节“使用 SCA 设备”](#)。但是, 您应该修改一些选项, 例如更改 Web 界面的口令, 更改 SCA 模式更新源, 启用存档模式, 或者配置电子邮件通知。有关细节, 请参见[第 41.4.2.5 节“自定义 SCA 设备”](#)。



## 警告: 数据保护

由于 SCA 设备服务器上的报告包含安全相关信息, 因此, 必须保护好 SCA 设备服务器上的数据, 以防未经授权的人员访问。

## 41.4.2.4 使用 SCA 设备

您可以将现有的 Supportconfig 存档手动上载到 SCA 设备，也可以一步即完成创建新 Supportconfig 存档并上载到 SCA 设备的操作。可以通过 FTP 或 SCP 来上载存档。对于这两种上载方式，您需要知道可用来访问 SCA 设备的 URL。要通过 FTP 上载，需要为 SCA 设备配置一台 FTP 服务器，请参见[过程 41.5 “安装和配置 SCA 设备”](#)。

### 41.4.2.4.1 将 supportconfig 存档上载到 SCA 设备

- 要创建 Supportconfig 存档并通过（匿名）FTP 上载：

```
> sudo supportconfig -U "ftp://SCA-APPLIANCE.COMPANY.COM/upload"
```

- 要创建 Supportconfig 存档并通过 SCP 上载：

```
> sudo supportconfig -U "scp://SCA-APPLIANCE.COMPANY.COM/tmp"
```

系统将提示您输入运行 SCA 设备的服务器的 root 用户口令。

- 如果要手动上载一个或多个存档，请将现有的存档文件（通常位于 /var/log/scc\_\*.tbz）复制到 SCA 设备中。对于目标，请选择设备服务器的 /tmp 目录或 /srv/ftp/upload 目录（如果为 SCA 设备服务器配置了 FTP）。

### 41.4.2.4.2 查看 SCA 报告

可以在装有浏览器并可以访问 SCA 设备报告索引页的任何计算机上查看 SCA 报告。

- 启动 Web 浏览器并确保 JavaScript 和 Cookie 已启用。
- 输入 SCA 设备的报告索引页作为 URL。

```
https://sca-appliance.company.com/sca
```

如果有任何疑问，请咨询系统管理员。

- 系统将提示您输入用于登录的用户名和口令。

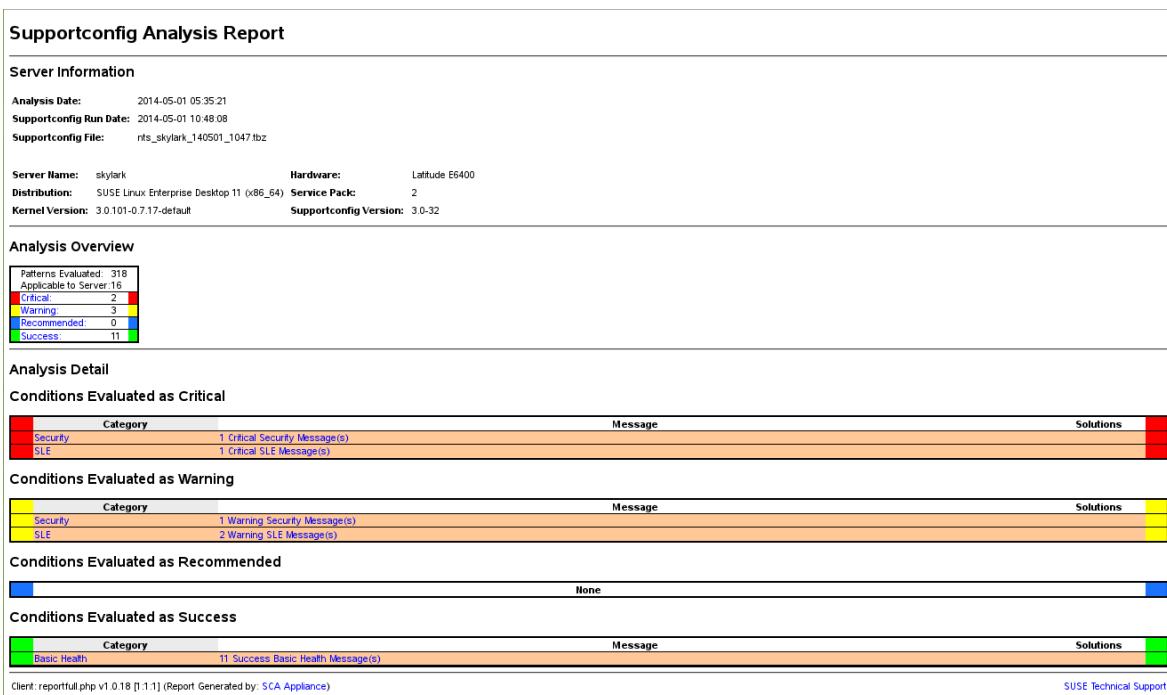


图 41.2：SCA 设备生成的 HTML 报告

4. 登录后，单击您想要阅读的报告的日期。
5. 首先单击基本状态类别将其展开。
6. 在消息列中单击某一个项。SUSE 知识库中的相应文章即会打开。阅读建议的解决方案，并遵循所述的说明。
7. 如果 Supportconfig 分析报告的解决方案列显示了其他项，请单击这些项。阅读建议的解决方案，并遵循所述的说明。
8. 请查看 SUSE 知识库 (<https://www.suse.com/support/kb/>)，以了解与 SCA 发现的问题直接相关的结果。设法解决这些问题。
9. 查看可前瞻性处理的结果，以免将来发生问题。

#### 41.4.2.5 自定义 SCA 设备

以下几节显示了如何更改 Web 界面的口令、更改 SCA 模式更新源、启用存档模式以及配置电子邮件通知。

#### 41.4.2.5.1 Web 界面的口令

SCA 设备 Web 界面要求使用用户名和口令登录。默认的用户名为 `scdiag`，默认的口令为 `linux`（如果未做其他的指定，请参见[过程 41.5 “安装和配置 SCA 设备”](#)）。请尽早将默认口令更改为较为安全的口令。您也可以修改用户名。

##### 过程 41.6：更改 WEB 界面的用户名或口令

1. 在 SCA 设备服务器的系统控制台上以 `root` 用户身份登录。
2. 在编辑器中打开 `/srv/www/htdocs/sca/web-config.php`。
3. 根据需要更改 `$username` 和 `$password` 的值。
4. 保存文件并退出。

#### 41.4.2.5.2 SCA 模式的更新

默认情况下，所有 `sca-patterns-*` 软件包将由一个 `root` cron 作业来定期更新，该作业将每晚执行 `sdagent-patterns` 脚本，而该脚本又会运行 `zypper update sca-patterns-*`。定期的系统更新将会更新所有 SCA 设备软件包和模式软件包。要手动更新 SCA 设备和模式，请运行：

```
> sudo zypper update sca-*
```

默认情况下，将从 SUSE Linux Enterprise 15 SP5 更新储存库安装更新。如果需要，您可以将更新源更改为某台 RMT 服务器。当 `sdagent-patterns` 运行 `zypper update sca-patterns-*` 时，将从当前配置的更新通道获取更新。如果该通道在 RMT 服务器上，将从该服务器提取软件包。

##### 过程 41.7：禁用 SCA 模式的自动更新

1. 在 SCA 设备服务器的系统控制台上以 `root` 用户身份登录。
2. 在编辑器中打开 `/etc/sca/sdagent-patterns.conf`。
3. 将项

```
UPDATE_FROM_PATTERN_REPO=1
```

更改为

```
UPDATE_FROM_PATTERN_REPO=0
```

4. 保存文件并退出。计算机无需重启动就能应用更改。

#### 41.4.2.5.3 存档模式

系统在分析了 Supportconfig 存档并将其结果存储在 MariaDB 数据库中后，会从 SCA 设备中删除所有这些存档。但是，如果要进行查错，在计算机中保留 Supportconfig 存档的副本可能会比较实用。默认情况下，存档模式处于禁用状态。

##### 过程 41.8：在 SCA 设备中启用存档模式

1. 在 SCA 设备服务器的系统控制台上以 root 用户身份登录。
2. 在编辑器中打开 /etc/sca/sdagent.conf。
3. 将项

```
ARCHIVE_MODE=0
```

更改为

```
ARCHIVE_MODE=1
```

4. 保存文件并退出。计算机无需重启动就能应用更改。

启用存档模式后，SCA 设备会将 Supportconfig 文件保存到 /var/log/archives/saved 目录，而不会将其删除。

#### 41.4.2.5.4 通过电子邮件发送 SCA 报告

SCA 设备可通过电子邮件发送所分析的每个 Supportconfig 的 HTML 报告文件。在默认情况下禁用该功能。启用该功能后，您可以定义要将报告发送到的电子邮件地址列表。定义会触发报告发送动作的状态消息级别 (STATUS\_NOTIFY\_LEVEL)。

**STATUS\_NOTIFY\_LEVEL** 的可能值

#### **\$STATUS\_OFF**

停用发送 HTML 报告功能。

#### **\$STATUS\_CRITICAL**

仅发送包含“关键”状态的 SCA 报告。

#### **\$STATUS\_WARNING**

仅发送包含“警告”或“关键”状态的 SCA 报告。

#### **\$STATUS\_RECOMMEND**

仅发送包含“建议”、“警告”或“关键”状态的 SCA 报告。

#### **\$STATUS\_SUCCESS**

发送包含“成功”、“建议”、“警告”或“关键”状态的 SCA 报告。

### 过程 41.9：为 SCA 报告配置电子邮件通知

1. 在 SCA 设备服务器的系统控制台上以 root 用户身份登录。
2. 在编辑器中打开 /etc/sca/sdagent.conf。
3. 搜索 STATUS\_NOTIFY\_LEVEL 项。该项默认设置为 \$STATUS\_OFF（禁用电子邮件通知）。
4. 要启用电子邮件通知，请将 \$STATUS\_OFF 更改为要针对其生成电子邮件报告的状态消息的级别，例如：

```
STATUS_NOTIFY_LEVEL=$STATUS_SUCCESS
```

有关详细信息，请参见 **STATUS\_NOTIFY\_LEVEL** 的可能值。

5. 定义要将报告发送到的收件人列表：

- a. 搜索 EMAIL\_REPORT='root' 项。
- b. 将 root 替换为要向其发送 SCA 报告的电子邮件地址列表。各电子邮件地址必须以空格分隔。例如：

```
EMAIL_REPORT='tux@my.company.com wilber@your.company.com'
```

6. 保存文件并退出。计算机无需重启动就能应用更改。以后生成的所有 SCA 报告都将通过电子邮件发送到指定的地址。

#### 41.4.2.6 备份和恢复数据库

要备份和恢复存储 SCA 报告的 MariaDB 数据库，请按如下所述使用 **scadb** 命令。**scadb** 由软件包 sca-appliance-broker 提供。

##### 过程 41.10：备份数据库

1. 在运行 SCA 设备的服务器的系统控制台上以 root 用户身份登录。
2. 通过执行以下命令将设备置于维护模式：

```
# scadb maint
```

3. 使用以下命令启动备份进程：

```
# scadb backup
```

数据将保存到 TAR 存档 sca-backup-\*sql.gz 中。

4. 如果您正在使用模式创建数据库开发自己的模式（参见第 41.4.3 节“开发自定义分析模式”），则还要备份以下数据：

```
# sdpdb backup
```

数据将保存到 TAR 存档 sdp-backup-\*sql.gz 中。

5. 将以下数据复制到另一台计算机或外部存储媒体：

- sca-backup-\*sql.gz
- sdp-backup-\*sql.gz
- /usr/lib/sca/patterns/local (仅当您已创建自定义模式时才需要复制该数据)

6. 使用以下命令重新激活 SCA 设备：

```
# scadb reset agents
```

#### 过程 41.11：恢复数据库

要基于备份恢复数据库，请按如下所述操作：

1. 在运行 SCA 设备的服务器的系统控制台上以 root 用户身份登录。
2. 将最新的 sca-backup-\*sql.gz 和 sdp-backup-\*sql.gz TAR 存档复制到 SCA 设备服务器。
3. 要解压缩文件，请运行：

```
# gzip -d *-backup-*sql.gz
```

4. 要将数据导入数据库，请执行：

```
# scadb import sca-backup-*sql
```

5. 如果您正在使用模式创建数据库创建自己的模式，则还要通过以下命令导入以下数据：

```
# sdpdb import sdp-backup-*sql
```

6. 如果您正在使用自定义模式，则还要基于备份数据恢复 /usr/lib/sca/patterns/local。

7. 使用以下命令重新激活 SCA 设备：

```
# scadb reset agents
```

8. 使用以下命令更新数据库中的模式模块：

```
# sdagent-patterns -u
```

### 41.4.3 开发自定义分析模式

SCA 设备随附了一个完整的模式开发环境（SCA 模式数据库），可让您开发自己的自定义模式。模式可用任何编程语言编写。要使这些模式可用于 Supportconfig 分析过程，需要将其保存到 /usr/lib/sca/patterns/local 并使其可执行。然后，SCA 设备和 SCA 工

具将会针对作为分析报告一部分的新 Supportconfig 存档运行这些自定义模式。有关如何创建（和测试）自己的模式的详细说明，请参见<https://www.suse.com/c/blog/sca-pattern-development/>。

## 41.5 在安装期间收集信息

在安装过程中，**supportconfig** 不可用。不过，您可以使用 **save\_y2logs** 从 YaST 收集日志文件。该命令会在 `/tmp` 目录下创建一个 `.tar.xz` 存档。

如果在安装过程的早期就出现问题，您或许可以通过 **linuxrc** 创建的日志文件收集信息。**linuxrc** 是一个在 YaST 启动之前运行的小命令。该日志文件位于 `/var/log/linuxrc.log`。

### ！ 重要：安装日志文件在已安装系统中不可用

日志文件在安装过程中可用，但在已安装好的系统中却不可用。当安装程序仍在运行时，请正确保存安装日志文件。

## 41.6 内核模块支持

对于任何企业操作系统，一个重要的要求就是您获得的环境方面的支持级别。内核模块是硬件（“控制器”）与操作系统之间关联最密切的连接器。SUSE Linux Enterprise 中的每个内核模块都有一个 **supported** 标志，该标志可使用以下三个值：

- “yes”，相当于 **supported**
- “external”，相当于 **supported**
- （空白，未设置），相当于 **unsupported**

可以使用下列规则：

- 默认情况下，经过自我重新编译的内核的所有模块都会标记为 **unsupported**。
- SUSE 合作伙伴支持的内核模块以及使用 **SUSE SolidDriver Program** 提供的内核模块会标记为 “external”。

- 如果未设置 `supported` 标志，装载此模块便会污染该内核。系统不支持污染的内核。不支持的内核模块随附于额外的 RPM 软件包 (`kernel-FLAVOR-extra`) 中。该软件包仅可用于 SUSE Linux Enterprise Desktop 和 SUSE Linux Enterprise Workstation Extension。默认情况 (`FLAVOR=default|xen|...`) 下，不会装载这些内核。此外，安装程序中不会提供这些不支持的模块，并且 `kernel-FLAVOR-extra` 软件包也不会包含在 SUSE Linux Enterprise 媒体中。
- 不是根据与 Linux 内核许可证兼容的许可证提供的内核模块也会污染内核。有关详细信息，请参见 `/usr/src/linux/Documentation/sysctl/kernel.txt` 和 `/proc/sys/kernel/tainted` 的状态。

### 41.6.1 技术背景

- Linux 内核：在 SUSE Linux Enterprise 15 SP5 上，`/proc/sys/kernel/unsupported` 的值默认为 `2 (do not warn in syslog when loading unsupported modules)`。在安装程序和已安装好的系统中，均会使用此默认值。有关更多信息，请参见 `/usr/src/linux/Documentation/sysctl/kernel.txt`。
- modprobe**：用于检查模块依赖项和装载模块的 `modprobe` 实用程序会相应地检查 `supported` 标志的值。如果该值为 “yes” 或 “external”，则会装载该模块，否则不会。有关如何覆盖此行为的信息，请参见第 41.6.2 节 “使用不支持的模块”。



#### 注意：支持

SUSE 一般不支持通过 `modprobe -r` 去除存储模块。

### 41.6.2 使用不支持的模块

虽然普遍的支持能力很重要，但在有些情况下可能需要装载不支持的模块。例如，为了测试或调试目的，或者如果您的硬件供应商提供了热修复。

- 要覆盖默认模块, 请将 `/lib/modprobe.d/10-unsupported-modules.conf` 复制到 `/etc/modprobe.d/10-unsupported-modules.conf`, 并将变量 `allow_unsupported_modules` 的值从 `0` 更改为 `1`。请不要直接编辑 `/lib/modprobe.d/10-unsupported-modules.conf`, `suse-module-tools` 软件包每次更新时都会覆盖所有更改。  
如果 initrd 中需要一个不支持的模块, 请记得运行 `dracut -f` 来更新 initrd。  
如果只想尝试装载模块一次, 可将 `--allow-unsupported-modules` 选项与 `modprobe` 结合使用。有关详细信息, 请参见 `/lib/modprobe.d/10-unsupported-modules.conf` 中的注释以及 `modprobe` 手册页。
- 在安装期间, 可通过驱动程序更新磁盘添加不支持的模块, 这样便会装载这些模块。要在引导过程中及引导之后强制装载不支持的模块, 请使用内核命令行选项 `oem-modules`。安装和初始化 `suse-module-tools` 软件包时, 系统将评估内核标志 `TAINT_NO_SUPPORT` (`/proc/sys/kernel/tainted`)。如果内核已污染, 将启用 `allow_unsupported_modules`。这可以防止不支持的模块在安装的系统中装载失败。如果安装期间没有任何不支持的模块, 并且未使用其他特殊的内核命令行选项 (`oem-modules=1`), 则默认行为仍是禁止不支持的模块。

请记住, 装载和运行不支持的模块会导致 SUSE 不支持该内核和整个系统。

## 41.7 更多信息

- `man supportconfig — supportconfig` 手册页。
- `man supportconfig.conf — Supportconfig 配置文件的手册页。`
- `man scatool — scatool` 手册页。
- `man scadb — scadb` 手册页。
- `man setup-sca — setup-sca` 手册页。
- <https://mariadb.com/kb/en/> — MariaDB 文档。
- <https://www.suse.com/c/blog/sca-pattern-development/> — 有关如何创建 (和测试) 您自己的 SCA 模式的说明。

- <https://www.suse.com/c/blog/basic-server-health-check-supportconfig/> – 使用 Supportconfig 的基本服务器状态检查。
- <https://community.microfocus.com/img/gw/groupwise/w/groupwise/34308/create-your-own-supportconfig-plugin> – 创建自己的 Supportconfig 插件。
- <https://www.suse.com/c/blog/creating-a-central-supportconfig-repository/> – 创建中心 Supportconfig 储存库。

## 42 常见问题及其解决方案

本章将描述一系列可能发生的问题及其解决方法。即使您的情况并未精确地列在这里，也可能有足够的相似的情况可提供解决您的问题的方法提示。

### 42.1 查找和收集信息

Linux 会较为详细地报告事件。当您的系统发生问题时，可以查看几个位置。一般而言，它们大多数都是 Linux 系统的标准日志，有一些是与 SUSE Linux Enterprise Desktop 系统相关的日志。多数日志文件可以用 YaST（杂项 > 启动日志）查看。

使用 YaST 可以收集支持团队所需的所有系统信息。使用其他 > 支持，然后选择问题类别。当所有信息都被集合后，将其附加在您的支持请求。

将出现最常检查的日志文件的列表，并附有其典型用途说明。包含 ~ 的路径表示当前用户的主目录。

表 42.1：日志文件

| 日志文件                            | 说明   |
|---------------------------------|--|
| <u>~/.xsession-errors</u>       | 来自当前运行的桌面应用程序的消息。  |
| <u>/var/log/apparmor/</u>       | 来自 AppArmor 的日志文件，详细信息请参见《安全和强化指南》。                          |
| <u>/var/log/audit/audit.log</u> | 来自审计的日志文件，用来跟踪对系统的文件、目录或资源的任何访问，并跟踪系统调用。有关详细信息，请参见《安全和强化指南》。 |
| <u>/var/log/mail.*</u>          | 来自邮件系统的消息。   |
| <u>/var/log/NetworkManager</u>  | 来自 NetworkManager 的日志文件通过网络连接收集问题                            |

| 日志文件                             | 说明   |
|----------------------------------|--|
| <code>/var/log/samba/</code>     | 包含 Samba 服务器及客户端日志消息的目录。                           |
| <code>/var/log/warn</code>       | 所有来自内核与系统日志守护程序的消息为“警告”或更高级别。                      |
| <code>/var/log/wtmp</code>       | 包含当前计算机会话的用户登录记录的二进制文件。可以使用 <code>last</code> 来查看。 |
| <code>/var/log/Xorg.*.log</code> | 来自 X Window 系统的各种启动和运行时日志文件。在调试失败的 X 启动时，该日志很有用。   |
| <code>/var/log/YaST2/</code>     | 包含 YaST 操作及其结果的目录。                                 |
| <code>/var/log/zypper.log</code> | Zypper 的日志文件。                                      |

除了日志文件外，您的计算机还可提供关于运行中的系统的信息。请参见表 42.2: `/proc` 文件系统的系统信息

表 42.2: `/proc` 文件系统的系统信息

| 文件                            | 说明                         |
|-------------------------------|----------------------------|
| <code>/proc/cpuinfo</code>    | 包含处理器信息，包括处理器类型、制造商、型号和性能。 |
| <code>/proc/dma</code>        | 显示当前使用的 DMA 通道。            |
| <code>/proc/interrupts</code> | 显示正在使用的中断和已使用的中断数量。        |
| <code>/proc/iomem</code>      | 显示 I/O (输入/输出) 内存的状态。      |
| <code>/proc/ioports</code>    | 显示当时正在使用的 I/O 端口。          |
| <code>/proc/meminfo</code>    | 显示内存状态。                    |

| 文件                            | 说明              |
|-------------------------------|-----------------|
| <code>/proc/modules</code>    | 显示各个模块。         |
| <code>/proc/mounts</code>     | 显示当前挂载的设备。      |
| <code>/proc/partitions</code> | 显示所有硬盘的分区。      |
| <code>/proc/version</code>    | 显示当前的 Linux 版本。 |

除了 `/proc` 文件系统外, Linux 内核还可通过 `sysfs` 模块 (内存中文件系统) 导出信息。该模块表示了内核对象及其属性以及关系。有关 `sysfs` 的更多信息, 请参见[第 29 章 “使用 udev 进行动态内核设备管理”](#) 中 `udev` 的环境。[表 42.3](#) 包含 `/sys` 下最常见目录的概览。

**表 42.3 : `/sys` 文件系统的系统信息**

| 文件                       | 说明                            |
|--------------------------|-------------------------------|
| <code>/sys/block</code>  | 包含系统中发现的每个块设备的子目录。通常多数是磁盘类设备。 |
| <code>/sys/bus</code>    | 包含每个物理总线类型的子目录。               |
| <code>/sys/class</code>  | 包含组合为功能型设备 (如图形、网络、打印机等) 的子目录 |
| <code>/sys/device</code> | 包含全局设备层次结构。                   |

Linux 自带了几个用于系统分析和监视的工具。请参见《系统分析和微调指南》, 第 2 章 “系统监视实用程序” 以选择在系统诊断中使用的最重要的工具。

以下包含的每个情景都以一个描述问题的标题开头, 后跟一两段内容, 提供建议的解决方案、解决方案详细信息的参考, 以及对其他可能相关的情景的交叉引用。

## 42.2 引导问题

引导问题是系统不能正确引导时出现的情况 (不能引导到预期的目标和登录屏幕)。

## 42.2.1 GRUB 2 引导加载程序无法装载

如果硬件运行正常，则可能是由于引导加载程序已损坏而使 Linux 无法在计算机上启动。在这种情况下，需要修复引导加载程序。为此，您需要按[第 42.5.2 节 “使用救援系统”](#) 中所述启动救援系统，然后根据[第 42.5.2.4 节 “修改和重新安装引导加载程序”](#) 中的说明操作。

另外，您可以按以下方式使用救援系统来修复引导加载程序。从安装媒体引导计算机。在引导屏幕中，选择更多 > 引导 Linux 系统。使用默认内核选项选择包含已安装的系统和内核的磁盘。系统引导后，启动 YaST 并切换到系统 > 引导加载程序。确保启用了将通用引导代码写入主引导记录选项，然后单击确定。如此会通过重写来修复损坏的引导加载程序，或者安装缺失的引导加载程序。

其他导致计算机无法引导的原因可能与 BIOS 相关：

### BIOS 设置

检查与硬盘相关的 BIOS 设置。如果在当前的 BIOS 设置中找不到硬盘本身，GRUB 2 可能就不能启动。

### BIOS 引导顺序

请检查您的系统引导顺序中是否包含硬盘。如果未启用硬盘选项，即使系统正确安装，在访问所需的硬盘时仍可能无法引导。

## 42.2.2 不显示登录或提示界面

这种情况通常在内核更新失败后发生，称为 **kernel panic**，原因是该过程中最后阶段有时可以在系统控制台上的错误类型。如果实际上计算机刚刚在软件更新后重引导，则当前目标是使用旧的经过验证的 Linux 版本内核和关联文件重引导。引导过程中，可以在 GRUB 引导加载程序屏幕中通过以下方式来实现此目的：

1. 用重置按钮重引导计算机，或关闭它之后再打开。
2. 当 GRUB 2 引导屏幕显示时，选择高级选项这一项，然后从菜单中选择之前的内核。计算机将使用内核的前一个版本及其关联文件引导。

3. 在引导过程完成后，使用 YaST 的引导加载程序模块去除新安装的内核，并根据需要将默认引导项设置为原先的内核。有关详细信息，请参见[第 18.3 节 “使用 YaST 配置引导加载程序”](#)。不过，这可能并不是必须要进行的操作，因为回滚过程中自动更新工具通常会为您修改它。

4. 重引导。

如果这样不能修复问题，请使用安装媒体引导计算机。计算机引导之后，继续至[步骤 3](#)。

### 42.2.3 无图形登录

如果计算机能够启动，但是无法引导进入图形登录管理器，则问题可能出在默认的 `systemd` 目标选项或 X Window 系统的配置上。要检查当前的 `systemd` 默认目标，请运行命令 `sudo systemctl get-default`。如果返回的值为 `not graphical.target`，请运行命令 `sudo systemctl isolate graphical.target`。如果图形登录屏幕已启动，请登录并启动 YaST > 系统 > 服务管理器，然后将默认系统目标设置为图形界面。此后，系统应该能够引导进入图形登录屏幕。

如果即使已引导或者切换到图形目标，图形登录屏幕也不启动，则原因可能是您的桌面或 X Window 软件设置错误或者已损坏。检查 `/var/log/Xorg.*.log` 中的日志文件，以了解 X 服务器尝试启动时发出的详细消息。如果桌面在启动期间发生错误，可能会在系统日志中记录错误消息，您可以使用命令 `journalctl` 查询该日志（有关详细信息，请参见[第 21 章 “journalctl：查询 systemd 日志”](#)）。如果这些错误消息指出问题出在 X 服务器中的配置上，请尝试修正这些问题。如果图形系统仍无法启动，请考虑重安装图形桌面。

### 42.2.4 无法挂载 Btrfs 根分区

如果 `btrfs` 根分区已损坏，请尝试以下选项：

- 使用 `-o recovery` 选项挂载该分区。
- 如果不起作用，请在您的根分区上运行 `btrfs-zero-log`。

## 42.2.5 强制检查根分区

如果根分区被破坏, 请在引导提示符处使用 forcefsck 参数: 这样会将选项 -f (force) 传递给 fsck 命令。

## 42.2.6 禁用交换以启用引导

当某个交换设备不可用且系统在引导期间无法启用该设备时, 引导可能会失败。通过将以下选项追加到内核命令行, 来尝试禁用所有交换设备:

```
systemd.device_wants_unit=off systemd.mask=swap.target
```

您也可以尝试禁用特定的交换设备:

```
systemd.mask=dev-sda1.swap
```

## 42.2.7 GRUB 2 在双引导系统上重引导时失败

如果 GRUB 2 在重引导期间失败, 请禁用 BIOS 中的 Fast Boot 设置。

## 42.3 登录问题

登录问题是指计算机虽然引导到预期的欢迎屏幕或登录提示界面, 却拒绝接受用户名和口令, 或者虽然接受了用户名和口令, 但是行为异常 (无法启动图形桌面、发生错误或转到了命令行等)。

### 42.3.1 有效的用户名和口令组合失败

如果系统配置为使用网络身份验证或目录服务, 但出于某些原因无法从为其配置的服务器上检索到结果, 通常就会发生此问题。只有作为唯一本地用户的 root 用户仍能登录到这些计算机。以下是计算机看似运行良好却无法正确处理登录的常见原因:

- 网络出现故障。有关此问题的进一步说明，请转到第 42.4 节 “网络问题”。
- DNS 在当时不起作用（这使得 GNOME 不起作用，并使系统无法向安全服务器发出经验证的请求）。如果是这种情况，则表现为计算机对任何操作的响应都需要极其长的时间。有关该主题的详细信息，请参见第 42.4 节 “网络问题”。
- 如果将系统配置为使用 Kerberos，则系统的本地时间与 Kerberos 服务器时间之间的差异可能超过了可接受的值（通常为 300 秒）。如果 NTP（网络时间协议）未正确地起作用，或者本地 NTP 服务器不起作用，则 Kerberos 身份验证将不再工作，因为该身份验证依赖于整个网络的通用时钟同步。
- 系统的身份验证配置不正确。请对相关的 PAM 配置文件进行检查以确定是否存在指令输入错误或排序错误。有关 PAM 的其他背景信息及相关配置文件的语法，请参见《安全和强化指南》，第 2 章 “通过 PAM 进行身份验证”。
- 主分区是加密的。有关该主题的详细信息，请参见第 42.3.3 节 “登录至加密的主分区失败”。

在不涉及外部网络问题的所有情况下，解决方法是将系统重引导到单用户方式并修复配置，然后再次引导到操作方式并重试登录。要引导到单用户方式，请执行以下操作：

1. 重新启动系统。此时将出现引导屏幕，其中显示一个提示。
2. 按 **Esc** 退出启动屏幕，并转到 GRUB 2 基于文本的菜单。
3. 按 **B** 进入 GRUB 2 编辑器。
4. 在包含内核参数的行中添加以下参数：

```
systemd.unit=rescue.target
```

5. 按 **F10**。
6. 输入 root 的用户名与口令。
7. 进行必要的一切更改。
8. 在命令行中输入 **systemctl isolate graphical.target**，以引导进入完全的多用户和网络模式。

## 42.3.2 有效的用户名和口令不被接受

这是到目前为止用户最常遇到的问题，因为有许多原因可能引起该问题。登录失败可由多种原因造成，取决于您是使用本地用户管理和身份验证，还是使用网络身份验证。

本地用户管理失败可由以下原因造成：

- 用户可能输入了错误的口令。
- 用户包含桌面配置文件的主目录已损坏或被写保护。
- X Window 系统可能无法验证此特定用户，尤其是在安装当前 Linux 发行套件之前，该用户的主目录已用于其他 Linux 发行套件的情况下。

要找到本地登录失败的原因，请执行如下操作：

1. 在尝试调试整个身份验证机制之前，请检查用户所记的口令是否正确。如果用户记错了口令，请使用“YaST 用户管理”模块更改其口令。注意 **Caps Lock** 键，如果需要请解锁。
2. 以 `root` 身份登录，并使用 `journalctl -e` 检查系统日志，找出登录过程和 PAM 的错误消息。
3. 尝试从控制台登录（使用 `Ctrl - Alt - F1`）。如果成功了，则问题不在 PAM 上，因为可以在该计算机上身份验证此用户。尝试找出与 X Window 系统或 GNOME 桌面相关的任何问题。有关详细信息，请参见第 42.3.4 节“GNOME 桌面有问题”。
4. 如果用户的主目录被其他 Linux 产品所使用，请将该用户主目录中的 `Xauthority` 文件删除。使用控制台登录（通过 `Ctrl - Alt - F1`），然后以该用户的身份运行 `rm .Xauthority`。这样应该可以消除该用户的 X 身份验证问题。然后再次尝试图形登录。
5. 如果桌面由于配置文件损坏而无法启动，请参见第 42.3.4 节“GNOME 桌面有问题”。

下面列出了在特定的计算机上对特定用户进行的网络身份验证可能失败的常见原因：

- 用户可能输入了错误的口令。
- 用户名存在于计算机的本地身份验证文件中，但同时网络身份验证系统也提供了该用户名，从而引起冲突。
- 主目录存在，但已损坏或不可用。该目录可能处于写保护状态或位于此刻无法访问的服务器上。

- 用户无权登录到身份验证系统中的该特定主机。
- 计算机出于某种原因更改了主机名，而用户无权登录到该主机。
- 计算机无法访问包含该用户信息的身份验证服务器或目录服务器。
- X Window 系统可能无法验证此特定用户，尤其是在安装当前 Linux 发行套件之前，该用户的主目录已用于其他 Linux 发行套件的情况下。

要通过网络身份验证找到登录问题的原因，请执行以下步骤：

1. 在尝试调试整个身份验证机制之前，请检查用户所记的口令是否正确。
2. 确定计算机在身份验证时要依赖的目录服务器，并确保计算机在正常运行且与其他计算机正常通讯。
3. 确定该用户的用户名和口令在其他计算机上是否有效，以确保保存在该用户的身份验证数据且已正确分发。
4. 确定其他用户是否可以登录到该故障计算机。如果其他用户可以正常登录，或 `root` 可以登录的话，请登录并使用 `journalctl -e` 文件检查系统日志。找到与登录尝试相对应的时间戳记，然后确定 PAM 是否生成了任何错误消息。
5. 尝试从控制台登录（使用 `Ctrl - Alt - F1`）。如果成功，则问题不在用户主目录中的 PAM 或目录服务器上，因为可以在该计算机上验证此用户。尝试找出与 X Window 系统或 GNOME 桌面相关的任何问题。有关详细信息，请参见第 42.3.4 节“GNOME 桌面有问题”。
6. 如果用户的主目录被其他 Linux 产品所使用，请将该用户主目录中的 `Xauthority` 文件删除。使用控制台登录（通过 `Ctrl - Alt - F1`），然后以该用户的身份运行 `rm .Xauthority`。这样应该可以消除该用户的 X 身份验证问题。然后再次尝试图形登录。
7. 如果桌面由于配置文件损坏而无法启动，请参见第 42.3.4 节“GNOME 桌面有问题”。

### 42.3.3 登录至加密的主分区失败

对于便携式计算机建议使用加密的主分区。如果无法登录到您的便携式计算机，原因可能是您的分区无法解锁。

在引导时，需要输入通行口令来解锁加密的分区。如果不输入它，引导进程继续，但保持分区锁定。

要解锁您的加密分区，请如下操作：

1. 按 **Ctrl** – **Alt** – **F1** 切换到文本控制台。

2. 以 root 用户身份登录。

3. 用以下步骤重启解锁进程：

```
# systemctl restart home.mount
```

4. 输入您的通行口令以解锁加密的分区。

5. 用 **Alt** – **F7** 退出文本控制台并切换回登录屏幕。

6. 如常登录。

#### 42.3.4 GNOME 桌面有问题

如果您遇到 GNOME 桌面问题，可以使用几种方法对行为异常的图形桌面环境进行查错。下面所述的建议过程提供了修复损坏的 GNOME 桌面的最安全选项。

##### 过程 42.1：GNOME 查错

1. 启动 YaST 并切换到安全和用户。

2. 打开用户和组管理对话框并单击添加。

3. 填写必填字段，然后单击确定以创建新用户。

4. 注销并以新用户身份登录。这将为您提供一个全新的 GNOME 环境。

5. 将旧用户帐户的 ~/.local/ 和 ~/.config/ 目录中的单个子目录复制到新用户帐户的相应目录。

每次复制操作之后，注销并以新用户身份再次登录，以检查 GNOME 是否仍正常工作。

6. 重复上一步，直到找到破坏 GNOME 的配置文件。

7. 以旧用户身份登录，并将有问题的配置文件移到其他位置。注销并以旧用户身份再次登录。
8. 删除之前创建的用户。

## 42.4 网络问题

系统的许多问题可能都与网络相关，即使初看起来不是这样。例如，系统不允许用户登录可能是因为网络有问题。本节介绍一个简单的核对表，您可以使用它来确定任何所遇到的网络问题的原因。

### 过程 42.2：如何识别网络问题

在检查计算机的网络连接时，请执行如下操作：

1. 如果使用的是以太网连接，请首先检查硬件。确保您的网络电缆已正确插入计算机和路由器（或集线器等）。以太网连接器旁边的控制灯通常应全部激活。  
如果连接失败，请检查网线在别的计算机上是否正常。如果正常，则可能是网卡引起了该问题。如果您的网络设置中有集线器或交换机，它们也可能有故障。
2. 如果使用的是无线连接，请检查是否可与其他计算机建立此无线链接。如果没有，请联系无线网络的管理员。
3. 完成对基本网络连通性的检查后，请尝试找出没有响应的服务。收集设置中所需的所有网络服务器的地址信息。在相应的 YaST 模块中查找这些信息，或者询问您的系统管理员。  
下面列出了设置中涉及的典型网络服务器及其服务中断时系统出现的症状。

### DNS (名称服务)

名称服务中断或发生故障会在许多方面影响网络运行。如果本地计算机依赖于任一网络服务器进行身份验证，而这些服务器由于名称解析问题而无法找到，则用户甚至还不能登录。网络中由中断的名称服务管理的计算机将无法“看到”彼此且不能通信。

### NTP (时间服务)

NTP 服务发生故障或中断可能会影响 Kerberos 身份验证和 X 服务器功能。

## NFS (文件服务)

如果任何应用程序所需的数据存储在 NFS 挂载目录中，则一旦此服务停止或配置错误，应用程序便无法启动或正常运行。最坏的情况是，如果由于 NFS 服务器发生故障导致找不到包含 .gconf 子目录的用户主目录，则该用户的个人桌面配置将无法起作用。

## Samba (文件服务)

如果任何应用程序需要的数据存储在有故障的 Samba 服务器上的某个目录中，它便无法启动或正常运行。

## NIS (用户管理)

如果您的 SUSE Linux Enterprise Desktop 系统依赖有故障的 NIS 服务器提供用户数据，用户将无法登录此计算机。

## LDAP (用户管理)

如果您的 SUSE Linux Enterprise Desktop 系统依赖有故障的 LDAP 服务器提供用户数据，用户将无法登录此计算机。

## Kerberos (身份验证)

身份验证无法进行，也无法登录任何计算机。

## CUPS (网络打印)

用户无法打印。

### 4. 请检查网络服务器是否正在运行并且您的网络设置是否允许您建立连接：

## ! 重要：限制

下面介绍的调试步骤只适用于简单的网络服务器/客户端设置，不涉及任何内部路由。假设服务器和客户端都是同一子网的成员，不需要额外的路由。

- a. 使用 ping IP\_ADDRESS/HOSTNAME (用服务器的主机名或 IP 地址替换) 来检查各台服务器是否正在运行，且能够对网络作出响应。如果此命令成功，表示您所查找的主机在正常运行，并且网络的名称服务配置正确。

如果 ping 命令失败，同时显示消息 `destination host unreachable`，则表明您的系统或期望的服务器未正确配置或已宕机。从另一台计算机运行 `ping IP address` 或 `YOUR_HOSTNAME`，检查是否可连接您的系统。如果可以从其他计算机访问您的计算机，则表明该服务器未运行或未正确配置。

如果 ping 命令失败且返回 `unknown host`，则表示名称服务未正确配置或使用的主机名不正确。要对该问题进行进一步的检查，请参见[步骤 4.b](#)。如果 ping 命令仍然失败，则可能网卡未正确配置或网络硬件存在故障。

- b.** 请使用 `host HOSTNAME` 来检查您尝试连接的服务器的主机名是否正确地转译为 IP 地址，反之亦然。如果此命令返回了该主机的 IP 地址，则名称服务已在正常运行。如果 `host` 命令失败，请检查您主机上所有与名称和地址解析相关的网络配置文件：

#### /var/run/netconfig/resolv.conf

此文件用于对当前使用的名称服务器和域进行跟踪。它是 `/run/netconfig/resolv.conf` 的符号链接，YaST 或 DHCP 通常会自动对其进行调整。请确保此文件采用以下结构，并且所有的网络地址和域名都正确无误：

```
search FULLY_QUALIFIED_DOMAIN_NAME
nameserver IPADDRESS_OF_NAMESERVER
```

此文件中可以包含多个名称服务器地址，但是其中必须至少有一个能够对您的主机提供正确的名称解析。如果需要，请使用 YaST 的“网络设置”模块（“主机名/DNS”选项卡）调整此文件。

如果您的网络连接是通过 DHCP 处理的，请在 YaST 的“网络设置”模块（“主机名/DNS”选项卡）中选择通过 DHCP 设置主机名（可针对所有接口全局设置，也可以逐个接口设置）和通过 DHCP 更新名称服务器和搜索列表，以允许 DHCP 更改主机名和名称服务信息。

#### /etc/nsswitch.conf

此文件告诉 Linux 到何处查找名称服务信息。它应显示为：

```
...
hosts: files dns
networks: files dns
...
```

dns 项是必不可少的。它告诉 Linux 要使用外部名称服务器。通常这些项是 YaST 自动管理的，但最好谨慎地检查。

如果主机上的所有相关条目均正确，请让系统管理员检查 DNS 服务器配置，以确定时区信息是否正确。如果确信主机和 DNS 服务器的 DNS 配置正确，请检查网络和网络设备的配置。

- c. 如果系统无法与网络服务器建立连接，并且已排除了名称服务出现问题的可能，则请检查网卡的配置。

使用 **ip addr show** NETWORK\_DEVICE 命令可检查此设备是否已正确配置。确保已正确配置带网络掩码 (/MASK) 的 inet address。如果 IP 地址中出现错误或网络掩码中缺少一位，将使您的网络配置无法使用。如有必要，也在服务器上执行该检查。

- d. 如果名称服务和网络硬件均配置正确且正在运行，但某些外部网络连接仍然长时间超时或完全失败，请使用 **traceroute** FULLY\_QUALIFIED\_DOMAIN\_NAME 命令（以 root 身份执行）来跟踪这些请求所经过的网络路由。此命令将列出某一请求从您的计算机传递到其目标所经过的所有网关（跳跃）。其中列出了每个跃点的响应时间以及此跃点是否可访问。请将 traceroute 和 ping 结合使用以确定故障原因并通知管理员。

确定了网络故障的原因后，就可以自行解决（如果问题出在您自己的机器上），或告诉网络系统管理员您的发现，以便其重新配置服务或修复必要的系统。

#### 42.4.1 NetworkManager 问题

如果网络连接有问题，请按[过程 42.2 “如何识别网络问题”](#) 中所述缩小范围。如果 NetworkManager 看上去有问题，请执行以下步骤，以获取相关的日志来找出 NetworkManager 失败原因的线索：

1. 以 root 用户身份打开外壳并登录。
2. 重启 NetworkManager：

```
> sudo systemctl restart NetworkManager
```

3. 以普通用户身份打开一个网页，例如 <http://www.opensuse.org>，看看是否可以连接。
4. 收集 /var/log/NetworkManager 中有关 NetworkManager 状态的任何信息。

有关 NetworkManager 的更多信息，请参考第 31 章 “使用 NetworkManager”。

## 42.5 数据问题

数据问题是指无论计算机是否能够正确引导，有一点是明确的，即系统上的数据损坏并且系统需要恢复。这些情况下需要对关键数据进行备份，以便您能够在系统出现故障时恢复故障前的状态。

### 42.5.1 管理分区映像

有时您需要从整个分区甚至硬盘来执行备份。Linux 附带了 dd 工具，该工具可用于创建磁盘的精确副本。与 gzip 一起使用可节省空间。

#### 过程 42.3：备份和恢复硬盘

1. 以 root 用户身份启动外壳。
2. 选择源设备。名称通常类似于 /dev/sda（标记为 SOURCE）。
3. 确定要把您的映像存储在何处（标记为 BACKUP\_PATH）。它不能与您的源设备相同。换句话说：如果从 /dev/sda 创建备份，则映像文件不得存储在 /dev/sda 下。
4. 运行以下命令创建压缩映像文件：

```
# dd if=/dev/SOURCE | gzip > /BACKUP_PATH/image.gz
```

5. 用以下命令恢复硬盘：

```
# gzip -dc /BACKUP_PATH/image.gz | dd of=/dev/SOURCE
```

如果只需备份某个分区，请将 SOURCE 占位符替换为相应的分区。在这种情况下，映像文件可以位于同一硬盘上不同的分区中。

## 42.5.2 使用救援系统

有多种原因会造成系统无法正常启动和运行。系统崩溃后造成文件系统损坏、配置文件损坏或引导加载程序配置损坏是最常见的原因。

为了帮助您解决这些状况，SUSE Linux Enterprise Desktop 提供了一套您可以引导的救援系统。该救援系统是一个小型 Linux 系统，可以装载到一个 RAM 磁盘并以 root 文件系统的形式挂载，使您可以从外部访问 Linux 分区。使用该救援系统，可以恢复或修改系统中任何一个重要的方面：

- 操作任意类型的配置文件。
- 检查文件系统中的缺陷和启动自动修复进程。
- 访问“更改 root”环境下的已安装系统。
- 检查、修改和重新安装引导加载程序配置。
- 从安装有误的设备驱动程序或不可用内核恢复。
- 使用 `parted` 命令调整分区大小。在 GNU Parted 网站 <http://www.gnu.org/software/parted/parted.html> 上可以找到有关该工具的更多信息。

该救援系统可以从各种来源和位置进行装载。最简单的选择是从原始安装媒体上引导该救援系统。

1. 将安装媒体插入 DVD 驱动器中。
2. 重新启动系统。
3. 在引导屏幕上按 **F4** 并选择 DVD-ROM。然后从主屏幕选择救援系统。
4. 在 `Rescue:` 提示符处输入 `root`。无需口令。

如果硬件设置不包含 DVD 驱动器，可以从网络源引导该救援系统。以下示例适用于远程引导的情形，如果使用另一引导媒体（例如 DVD），则要相应地修改 `info` 文件，并像正常安装一样进行引导。

1. 进入 PXE 引导设置的配置中，添加下面的行：`install=PROTOCOL://INSTSOURCE` 和 `rescue=1`。但如果需要启动修复系统，请使用 `repair=1`。如同正常的安装一样，`PROTOCOL` 代表任何支持的网络协议（NFS、HTTP、FTP 等），`INSTSOURCE` 代表网络安装源的路径。

2. 使用“网络唤醒”引导系统（如《Deployment Guide》，第 13 章“Preparing network boot environment”，第 13.5 节“Using wake-on-LAN for remote wakeups”中所述）。

3. 在 Rescue：提示符处输入 root。无需口令。

进入该救援系统后，可通过 Alt – F1 到 Alt – F6 键来访问虚拟控制台。

/bin 目录中提供了一个外壳和其他有用的实用程序，如 mount 程序。/sbin 目录包含重要的用于查看和修复文件系统的文件和网络实用程序。此目录还包含用于系统维护的最重要的二进制文件，如 fdisk、mkfs、mkswap、mount 和 shutdown，以及用于维护网络的 ip 和 ss。目录 /usr/bin 包含 vi 编辑器、find、less 和 SSH。

要查看系统消息，请使用命令 dmesg，或使用 journalctl 查看系统日志。

#### 42.5.2.1 检查和操作配置文件

举一个可以通过该救援系统修复配置的例子，假设有一个被损坏的配置文件，使该系统无法正常引导。您可以通过救援系统修复该配置文件。

要操作配置文件，请执行以下步骤：

1. 用上述方法之一启动救援系统。

2. 要将位于 /dev/sda6 下的根文件系统挂载到救援系统中，请使用如下命令：

```
> sudo mount /dev/sda6 /mnt
```

系统的所有目录现在均位于 /mnt 下

3. 将目录切换为所挂载的 root 文件系统：

```
> sudo cd /mnt
```

4. 在 vi 编辑器中打开有问题的配置文件。调整并保存配置。

5. 从救援系统中卸载 root 文件系统：

```
> sudo umount /mnt
```

## 6. 重引导计算机。

### 42.5.2.2 修复和检查文件系统

通常，不能在正在运行的系统上修复文件系统。如果遇到严重问题，您甚至都无法挂载 root 文件系统，系统引导可能以显示“kernel panic”结束。在这种情况下，唯一的方法是从外部修复系统。该系统包含的实用程序可检查并修复 `btrfs`、`ext2`、`ext3`、`ext4`、`xfs`、`dosfs` 和 `vfat` 文件系统。您可以试试命令 `fsck. FILESYSTEM`。例如，如果需要对 `btrfs` 进行文件系统检查，请使用 `fsck.btrfs`。

### 42.5.2.3 访问已安装系统

如果需要从救援系统访问已安装系统，需要在更改 `root` 环境中执行此操作。例如，修改引导加载程序配置或执行硬件配置实用程序。

要设置基于已安装系统的更改 `root` 环境，请执行以下步骤：

#### 1. 提示：导入 LVM 卷组

如果您使用的是 LVM 设置（有关更多一般性细节，请参见《Deployment Guide》，第 6 章“Expert Partitioner”，第 6.2 节“LVM configuration”），请导入所有现有的卷组，以便能够查找和挂载设备：

```
rootvgimport -a
```

运行 `lsblk` 以检查哪个节点对应于根分区。在本例中，该节点为 `/dev/sda2`：

```
> lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINT
sda        8:0     0 149,1G  0 disk
|---sda1   8:1     0    2G  0 part  [SWAP]
|---sda2   8:2     0   20G  0 part   /
|---sda3   8:3     0 127G  0 part
    |---cr_home 254:0  0 127G  0 crypt /home
```

## 2. 从安装的系统挂载根分区：

```
> sudo mount /dev/sda2 /mnt
```

## 3. 挂载 /proc、/dev 和 /sys 分区：

```
> sudo mount -t proc none /mnt/proc  
> sudo mount --rbind /dev /mnt/dev  
> sudo mount --rbind /sys /mnt/sys
```

## 4. 现在可以“更改根分区”为新的环境，并保留 bash 外壳：

```
> chroot /mnt /bin/bash
```

## 5. 最后，挂载已安装系统的剩余分区：

```
> mount -a
```

## 6. 现在可以访问已安装系统了。在重引导系统之前，请使用 umount -a 卸载分区，并使用 exit 退出“更改根”环境。



### 警告：限制

尽管对已安装系统的文件和应用程序有完全访问权，但仍有一些限制。运行的内核是用救援系统引导的那个，不是用更改 root 环境引导的那个。它仅支持关键硬件，如果内核版本不完全相同，则无法从已安装系统中添加内核模块。始终使用 uname -r 检查当前正在运行的（救援）内核版本，然后确定更改根环境中的 /lib/modules 目录下是否有匹配的子目录。如果是，可以使用已安装模块；否则，需要在其他媒体（例如闪盘）上提供模块的正确版本。很多时候，救援内核版本与已安装模块不同，这样您完全无法访问声卡等。也不可能启动图形用户界面。

还应注意，在使用 Alt - F1 - Alt - F6 键切换控制台时，要退出“更改 root”环境。

### 42.5.2.4 修改和重新安装引导加载程序

有时，系统无法引导是因为引导加载程序配置已损坏。例如，如果没有正常工作的引导加载程序，启动例程将无法将物理驱动器转化为 Linux 文件系统中的实际位置。

要检查引导加载程序配置并重新安装引导加载程序，请执行以下步骤：

1. 如第 42.5.2.3 节 “访问已安装系统” 中所述执行必要的步骤以访问已安装系统。
2. 检查系统上是否已安装 GRUB 2 引导加载程序。如果未安装，请安装 `grub2` 软件包并运行

```
> sudo grub2-install /dev/sda
```

3. 根据第 18 章 “引导加载程序 GRUB 2” 中所述的 GRUB 2 配置原则，检查下列文件是否正确配置，并根据需要应用修复。
  - `/etc/default/grub`
  - `/boot/grub2/device.map`
  - `/boot/grub2/grub.cfg` (此文件是系统生成的，不要编辑)
  - `/etc/sysconfig/bootloader`
4. 使用以下命令序列重新安装引导加载程序：

```
> sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 卸载分区，注销“更改根目录”环境并重引导该系统：

```
> umount -a  
exit  
reboot
```

## 42.5.2.5 修复内核安装

内核更新会产生新的 bug，这样会影响系统运行。例如，系统某个硬件的驱动程序有故障，您就无法访问和使用该硬件。在这种情况下，需还原到上一个工作内核（如果在系统上可用），或从安装媒体安装原始内核。



## 提示：如何在更新后保留最后几个内核

为了防止内核更新出现故障后无法进行引导，请使用内核多版本功能，并告知 `libzypp` 在更新后保留哪些内核。

例如，要始终保留最后两个内核和当前正在运行的内核，请将

```
multiversion.kernels = latest,latest-1,running
```

到 `/etc/zypp/zypp.conf` 文件。有关更多信息，请参见第 27 章 “安装多个内核版本”。

类似的情况是，当您需要重新安装或更新已损坏而不受 SUSE Linux Enterprise Desktop 支持的设备驱动程序时。例如，当硬件供应商使用特定设备时，比如硬件 RAID 控制器，它需要一个二进制驱动程序才能被操作系统识别。供应商通常会发布含有固定版或更新版必要驱动程序的驱动程序更新磁盘 (DUD)。

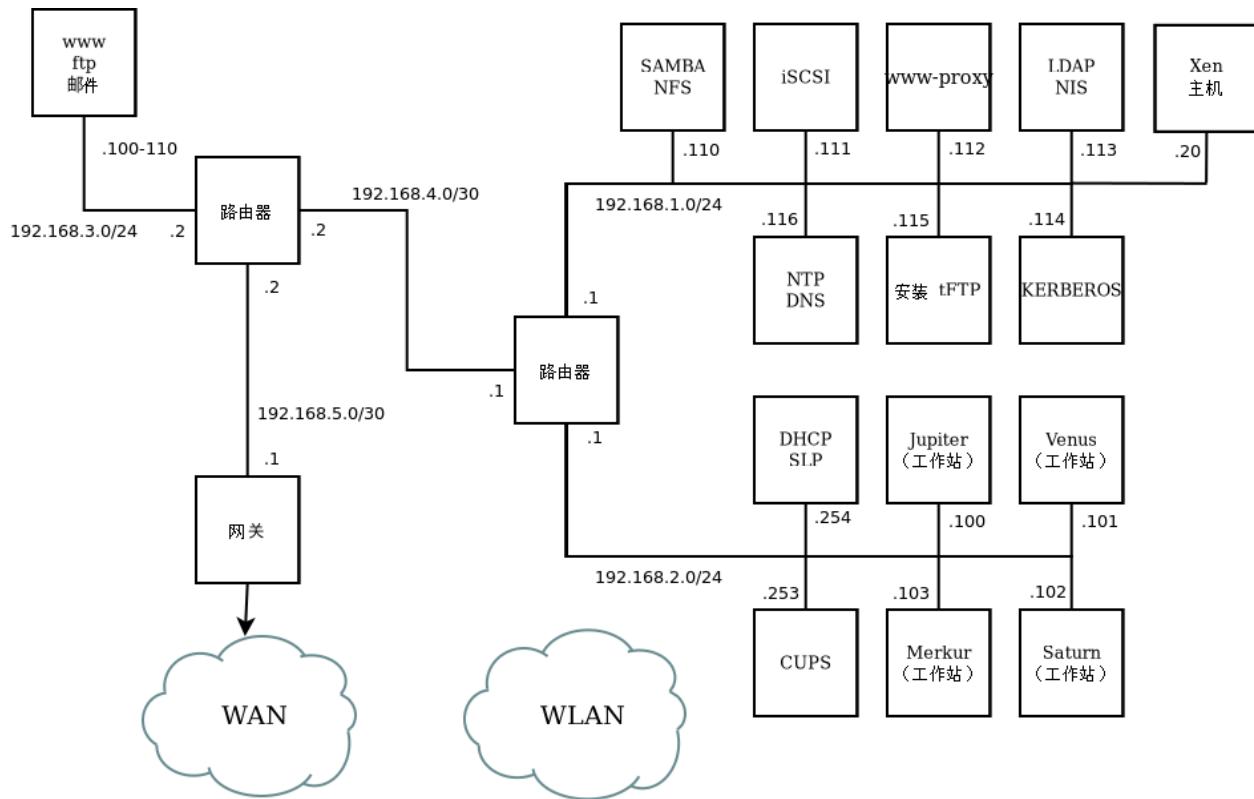
这两种情况下，您都需要以救援模式访问已安装系统，并修复内核相关问题，否则系统可能无法正确引导：

1. 从 SUSE Linux Enterprise Desktop 安装媒体引导。
  2. 如果您正在从内核更新故障中恢复，请跳过此步骤。如果需要使用驱动程序更新磁盘 (DUD)，请在出现引导菜单后按 `F6` 装载驱动程序更新，并选择驱动程序更新的路径或 URL，然后确认是。
  3. 从引导菜单中选择救援系统，并按 `Enter`。如果选择使用 DUD，将要求您指定存储驱动程序更新的位置。
  4. 在 `Rescue:` 提示符处输入 `root`。无需口令。
  5. 手动将目标系统和“更改 root”挂载新环境。有关详细信息，请参见第 42.5.2.3 节“访问已安装系统”。
  6. 如果使用的是 DUD，请安装/重新安装/更新有故障的设备驱动程序软件包。始终确保已安装的内核版本与您正在安装的驱动程序版本完全相同。
- 如果要修复有故障的内核更新安装，可以从安装媒体使用以下过程安装原始内核。

- a. 使用 `hwinfo --cdrom` 识别您的 DVD 设备，并使用 `mount /dev/sr0 /mnt` 挂载该设备。
  - b. 导航到 DVD 上存储内核文件的目录，例如 `cd /mnt/suse/x86_64/`。
  - c. 使用 `rpm -i` 命令根据您的偏好安装所需的 `kernel-*`、`kernel-*base` 和 `kernel-*extra` 软件包。
7. 更新配置文件，必要时可重初始化引导加载程序。有关详细信息，请参见[第 42.5.2.4 节“修改和重新安装引导加载程序”](#)。
8. 从系统驱动器中删除所有可引导的媒体，然后重引导。

## A 网络示例

此网络示例贯穿 SUSE® Linux Enterprise Desktop 文档中所有与网络相关的章节。



# B GNU licenses

## 修订历史

2023-02-03

### This appendix contains the GNU Free Documentation License version 1.2.

#### GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software. We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

#### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law. A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role

of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document

within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

## ADDENDUM: How to use this License for your documents

Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.