凝思安全操作系统

V6.0.100 ARM64 用户手册

V1.0

北京凝思软件股份有限公司

目录

产品概述

凝思安全操作系统是北京凝思软件股份有限公司自主研发、拥有完全自主知识产权、具备等保四级要求、并且达到军 B 级安全级别的操作系统，是国内首家达到安全服务器保护轮廓 EAL3 级别的安全产品。

凝思软件经过多年的研究，开发出的凝思安全操作系统具有以下主要特点：

- 高可用性

  凝思安全操作系统发布前进行了长时间的压力测试，能够保证在高内存和 CPU 负载环境下稳定运行，为各类应用提供稳定的运行平台。

  为进一步提高操作系统稳定性，凝思安全操作系统还提供多种冗余容错机制，降低部件故障引起的整机失效。这些机制包括：

  - 磁盘冗余技术

    提供软 RAID 技术，通过磁盘冗余降低磁盘故障引起的系统风险。支持在线重建，减少系统恢复时间，提高系统稳定性。

  - 网卡冗余技术

    提供网卡的负载均衡和冗余备份，当两块网卡自身及其链路正常时，可提供网卡负载均衡功能，提高网络传输效率；当一块网卡或其链路发生故障时，另一网卡仍可继续提供服务，提高网络的整体可靠性。

  - 磁盘阵列卡冗余技术

    支持磁盘阵列卡的主备机制，当主卡或其通信链路发生故障时，凝思安全操作系统将自动切换至备份卡进行设备访问，保证磁盘阵列数据的连续性。

  - 软件固化技术

    支持关键操作系统数据以只读形式存储于电子盘中，防止恶意或偶然操作破坏系统数据。支持无硬盘工作模式，系统运行在基于电子盘的环境中，进一步提高系统稳定性。

  - 双机热备和服务热切换机

    支持分布式双机数据热备份和服务热切换机制，当主服务器宕机时服务将自动漂移至辅服务器，由于数据实现实时同步，在服务器单点故障时，服务仍不会终止，系统的整体稳定性得以充分保障。

- 高兼容性

  凝思安全操作系统适用于从大型计算到桌面办公等各种环境，支持各类通用和专业应用，具有良好的软硬件兼容性。

  凝思安全操作系统的 API 接口和实用工具完全遵循 POSIX 标准，并支持 LSB 和 FHB 等 Linux 相关标准，能够二进制兼容各类为 Linux 系统开发的应用软件，并可在二进制保持与其它 Linux 发行版的兼容。支持基于 Java 的跨平台软件，可实现基于多款中间件软件的 Web 应用系统。支持 32 位和 64 位应用程序，使用户在 32 位系统上开发的软件可直接运行于 64 系统，缩短用户应用系统的研发周期。

  凝思安全操作系统提供丰富的驱动程序，支持各类主流磁盘驱动器、网卡驱动器

和显示控制器等硬件设备。兼容国内外各大主流厂商的多款服务器和桌面计算机。凝思软件还可协助第三方硬件厂商完成驱动程序的研发和移植，实现加密卡等专用硬件设备支持。

- 高效性

  凝思安全操作系统可针对服务器、工作站、专用设备和桌面环境进行特别优化，获得比通用操作系统更高的运行效率。

  在特定项目的应用环境中，凝思可以还可针对应用系统的特性对操作系统做进一步的优化，使系统的运行效率最大化。各个操作系统内核组件的定制和优化，可充分发挥硬件平台的性能，对应用程序提供最佳支持，构建稳定、高效的计算环境。定制的内容包括:

  - 定制和剪裁最小应用软件运行环境，保证系统组件的可控性，提高系统的运行效率和可用性。
  - 定制进程调度策略，减少调度开销，提高操作系统和应用程序的响应速度和运行效率。
  - 定制文件系统类型和存储模式，提供可靠和有效的文件系统支持，使应用程序能够快速访问文件数据。
  - 定制驱动程序，实现用户程序对设备数据的快速访问，降低设备访问的系统资源开销，提高应用系统的数据处理能力。

- 易维护性

  凝思安全操作系统的配置、使用和维护方法与传统 UNIX 和 Linux 保持一致，提供丰富的管理和维护软件，既可通过命令行工具完成系统配置和维护，又可通过图形界面完成相关操作，简化用户操作步骤，减少管理员的维护工作量。支持键盘/显示器、串口和网络等多种接入方式，便于管理员的本地和远程管理。基于网络的远程管理支持加密通道的远程登录，可使用命令行方式管理远程系统；同时支持远程图形化管理和远程桌面重定向，为管理员和用户提供图形界面的系统维护和操作。服务自启动功能使系统无须人工干预即可自动进入服务提供状态，减少系统故障恢复时间，提高系统可获得性。丰富的审计日志使管理员能够准确分析并定位各类系统故障，为快速排除问题，恢复系统正常运行提供支持。凝思安全操作系统还提供状态显示、声音报警、邮件和短信通知等多种报警机制，使管理员能够及时掌握系统的运行状态，第一时间获取系统紧急故障和安全性信息。

目前，以凝思安全操作系统为核心的安全服务器系统平台已广泛应用在国家部委、军队系统及电信、电力等行业关键部门，并获得了用户的一致好评。

凝思安全操作系统 V6.0.100-ARM64 在以下方面对系统进行了改进和完善:

- 内核版本

  凝思安全操作系统 V6.0.100-ARM64 采用 4.19.90 版内核

- CPU 架构
  凝思安全操作系统 V6.0.100-ARM64 为 64 位，支持 arm64 架构。
- 系统后台服务
  采用 systemd，支持更快的启动速度，兼容原先的 sysvinit。
- 虚拟化软件
  支持 qemu 3.1 等虚拟化软件。
- 大数据平台
  支持 cm/cdh 6.x、华三大数据平台等。
- 容器支持
  支持 docker 1.5。
- 开发语言
  支持 gcc-8.3.0、perl 5.28、python 2.7/3.7、php 7.3、jdk1.11 等多种开发语言。
- libc 版本
  采用 2.28 版 libc。
- 图形环境
  支持 gnome 3.30、kde 3.53、mate 1.20、xfce 4.12 等多种图形环境。
- 办公软件
  支持 LibreOffice 6.1.5、thunderbird 68.12、iceweasel 68.12。
- 强制访问控制
  提供强制访问控制（MAC）机制，支持基于安全标记，按照组织安全策略决定访问许可。
- 强制行为控制
  提供强制行为控制（MBC）机制，控制进程可执行程序的范围，防范恶意代码攻击。
- 强制能力控制
  提供强制能力控制（MCC）机制，缩小系统 TCB，消除超越安全机制的特权程序，降低系统安全隐患。
- 四权分立系统管理
  提供四权分立的系统管理机制，各系统管理员相互牵制，不能独立控制系统，防止管理员恶意或偶然操作破坏系统安全。
- 无 root 运行模式
  提供无 root 运行模式，禁止 root 登录，禁止用户进程切换为 root 身份，保护关键数据的私密性和完整性。

关于本书

文档约定

1、本手册中会有某些字词使用了不一样的字体、样式，规律如下：

- command

  命令，表示您可以在命令行中键入单词或短语，然后按 Enter 键来启用命令。

  例如：使用 cat testfile 命令来查看当前目录中一个叫 testfile 的文件。

- file name

  文件名、目录名、路径，表示系统中存在着一个叫这个名称的文件或目录。

  例如：

  - 您的主目录中的.bashrc 文件包括您自用的 bash shell 定义和别名。
  - /etc/fstab 文件包括关于不用系统设备和文件系统的信息。

- parameter value

  参数及其设定值。

  例如：

  ```
  ServerType standalone
  ```

  ServerType 定义服务器的启动方式，缺省值为独立方式 standalone。

- Script

  脚本。

  例如：

  ```
  <Directory />
      Options FollowSymLinks
      AllowOverride None
  </Directory>

  <Directory "/var/www/htdocs">
  ```

- key

  键盘上的按键。

  例如：使用 Tab 键补全，键入一个字符然后按 Tab 键，您的终端上就会显示目录中起首为那个字符的文件列表。

2、除此之外，我们还使用几种不同的方式来强调某些信息。按照信息对您的系统的重要程度，它们被标为注意、提示、重要、小心或警告。例如：

注意

切记：凝思安全操作系统区分大小写。换一句话说 ,rose 不是 ROSE 或 Rose 。

提示

目录 /usr/share/doc 包括了关于您的系统上安装的软件包的附加信息。

重要

如果您修改了 DHCP 配置文件，这些改变在您重启 DHCP 守护进程之后才会生效。

小心

管理员帐号是用来执行系统管理任务的，执行日常任务请使用一个常规的用户帐号。

警告

删除分区操作没有确认过程，会直接将已有分区删除。如果您想撤销该删除操作，只能不将分区结果写入磁盘而直接退出分区界面，然后再重新进入分区界 面，但之前的所有分区操作都将丢失。慎用！

术语简介

您会在凝思安全操作系统的文档中经常看到以下这些术语:

- shell

  登录 Linux 系统时，系统会为登录用户启动一个程序。通过这个程序，可以和 Linux 进行交互。这个程序就是通常所说的 Linux shell，它的作用就是执行通过键盘输入的命令。

- 命令（Command）

  给计算机的命令，多数使用键盘或鼠标输入。

- 命令行（Command line）

  在 shell 提示中键入命令的地方。

- shell 提示（shell prompt）

  用户和操作系统的命令行界面。shell 解释用户输入的命令，并把它们传递给操作系统。shell 运行的时候会在屏幕上显示一个提示符，这表示它在等待用户的命令。当输入了命令并按 Enter 建，shell 就开始解释命令并且执行这个命令。如果输入了一个并不存在的命令，shell 也会提示您，并且重新显示提示符，等待输入下一个命令。

- 手册页（Man page）和信息页（Info page）

  手册（Man 是 manual 的简写）页和信息页提供了关于命令或文件的详细信息（手册页比信息页提供的解释要简略）。例如，要阅读 su 命令的手册页，在 shell 提示下输入 man su（或输入 info su 来阅读信息页）。要关闭手册页或信息页，按 q 。

- 根（root）

  根是在安装中创建的超级用户账号，除非您设置了强制访问控制，否则它对您的系统有完全的访问权。在一般 UNIX/Linux 系统中，您必须登录为根用户来完成某些系统管理任务，如改变管理口令和运行系统配置工具。用户帐号的创建目的是使您不必使用根帐号来完成普通的用户任务，从而减少永久性损坏您的操作系统安装或应用程序的机会。

- 终端

  终端是一种字符型设备，它有多种类型，通常使用 tty 来简称各种类型的终端设备。tty 一词源于 Teletypes，或者 teletypewriters，原来指的是电传打字机，是通过串行线用打印机键盘通过阅读和发送信息的东西，后来这东西被键盘与显示器取代，所以现在叫终端比较合适。

- 控制台

  控制台是与操作系统交互的设备，系统将一些信息直接输出到控制台上。

  在 Linux 系统中，计算机显示器通常被称为控制台终端（Console）。它仿真了类型为 Linux 的一种终端，并且有一些特殊文件与之相关联：tty0、tty1、tty2 等。当您在控制台登录时，使用的是 tty1。使用 Alt + F1 ~ F6 组合键时，我们就可

以切换到 tty2、tty3 等上面去。tty1～tty6 等称为虚拟终端，而 tty0 则是当前所使用虚拟终端的一个别名，系统所产生的信息会发送到该终端上（这时也叫控制台终端）。因此不管当前正在使用哪个虚拟终端，系统信息都会发送到控制台终端上。您可以登录到不同的虚拟终端上去，因而可以让系统同时有几个不同的会话存在。

- 图形化用户界面（Graphical User Interface，GUI）

  互动界面、图标、菜单、以及允许用户使用鼠标和键盘来引发启动程序和打开文件等行动的统称。

- 图形化桌面（Graphical Desktop）

  GUI 中的最可见部分。桌面是您的“用户主目录”和“从这里开始”启动器图标的位置。您可以给桌面配置特殊的背景、颜色和图片来为它添加一点儿个人色彩。

- 图标（Icon）

  代表应用程序、文件夹、快捷方式或系统资源（如软盘驱动器）的小图像。启动器（Launcher）图标通常指启动应用程序的快捷方式。

- 面板（Panel）

  桌面工具栏。通常横贯桌面底部。面板上包含“主菜单”按钮和启动常用程序的快捷方式图标。您可以定制面板来满足您的个人需要。

- X 或 X 窗口系统（X Window System）

  这两个术语代表图形化用户界面环境。如果您“在 X 内”或“在运行 X”，这意味着您的工作环境是 GUI 而非控制台。

- 互联网

  互联网（internet，普通名词），泛指网络集合，是由各种不同类型和规模的计算机网络组成的计算机网络。组成互联网的计算机网络，可以是局域网（LAN）、城域网（MAN）或者广域网（WAN）等，其基本组成单元是网络，可以将互联网称为“连接网络的网络”。

- 因特网

  因特网（Internet，专用名词），特指由 TCP/IP 协议构建的互联网（internet）。因特网是互联网的一种。

- 万维网

万维网（World Wide Web，简称“Web”、“WWW”、“W3”）是一个由许多互相链接的超文本文档组成的系统，通过 Internet 访问。在这个系统中，每个有用的事物，称为一样“资源”；并且由一个全域“统一资源标识符”（URI）标识；这些资源通过超文本传输协议（Hypertext Transfer Protocol）传送给使用者，而后者通过点击链接来获得资源。

万维网常被当成因特网的同义词，这是一种误解，万维网是靠着因特网运行的一项服务。

如何使用本书

本书旨在指导用户使用凝思安全操作系统 V6.0.100-ARM64，系统安装请阅读《凝思安全操作系统 V6.0.100-ARM64-安装手册》。

第 1 章 登录和注销

1.1 登录

1、启动计算机后进入 GRUB 界面，如图 1.1 所示。

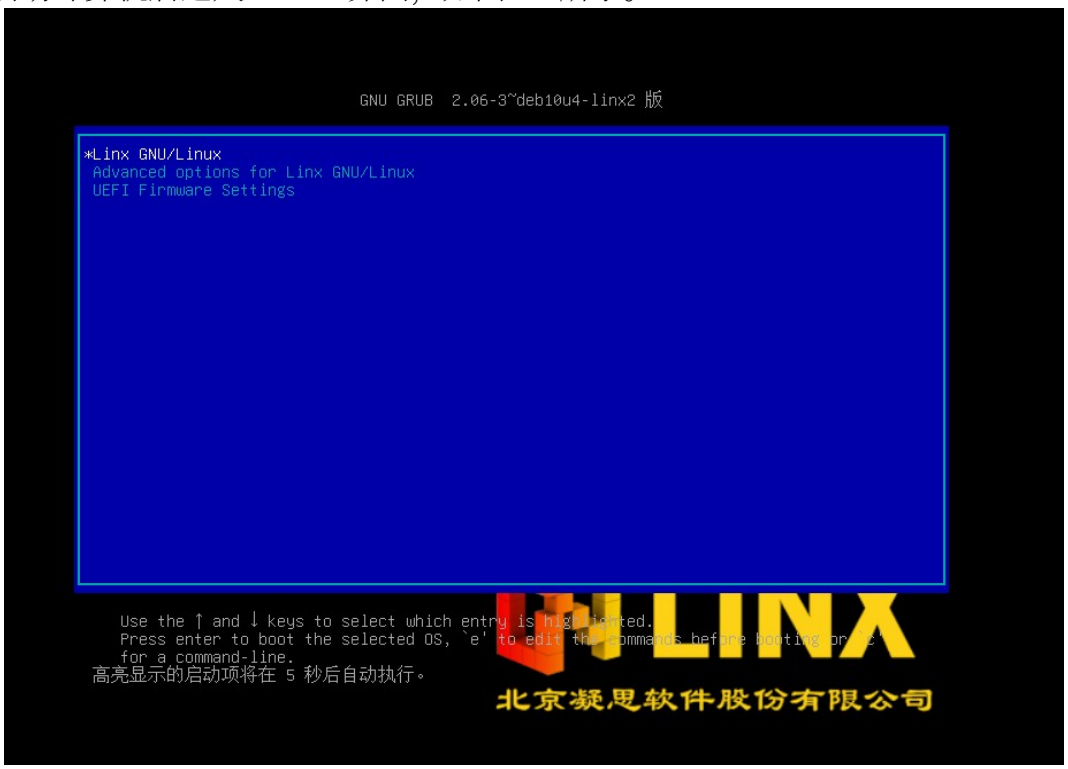


图 1. 1 GRUB 界面

2、凝思安全操作系统 V6.0.100 启动界面有两个可选择的菜单项:

- Linux GNU/Linux
  维护模式，启动后任何用户均可登录，主要用来进行系统配置、维护事务。
- Advanced options for Linx GNU/Linux
  高级模式，该模式下又提供了三种可选的启动模式分别为:
  - Linx GNU/Linux,Linux 4.19.0-11-linx-security-amd64
    维护模式，启动后任何用户均可登录，主要用来进行系统配置、维护事务。
  - Linx GNU/Linux,with Linux 4.19.0-11-linx-security-amd64 (recovery mode)
    恢复模式，提供一个 root shell 和只读的文件系统，用于进行系统恢复。
  - Linx GNU/Linux,with Linux 4.19.0-11-linx-security-amd64 (linx no root mode)
    运行模式，启动后无 root 用户，由管理员用户和普通用户进行日常事务管理和操作。
- UEFI Firmware Settings
  UEFI 固件设置模式，启动后可以对 UEFI 固件进行设置。

3、当前 grub 界面默认选择的 Linux GNU/Linux，可以直接按 Enter 键启动系统。或者

按 ↓ 键移动光标至 Advanced options for Linx GNU/Linux，按 Enter 键进入高级模式，再根据需要选择启动模式。

4、所有用户都必须进行鉴定。当您启动系统后，系统会提示您输入用户名和口令，即：系统中已创建的用户名和口令。如果尚未创建，请与管理员联系以获取用户名和口令。

5、如果您安装了桌面环境，系统将自动进入桌面登录画面，详见第 1.1.1 节。如果没有安装桌面环境，系统将使用控制台的登录方式，详见第 1.1.2 节。您也可以切换到其它控制台登录。

1.1.1 从桌面登录

1、登录屏幕中包含用户名输入字段，输入用户名后敲回车键，继续输入用户口令并回车，可登录到系统，如图 1.2 所示。

图 1. 2 登录

提示

如果用户口令输入错误，请按 ESC 键退出，然后重新输入密码。

2、系统默认使用 Mate 桌面环境，如图 1.3 所示。

图 1. 3 Mate 桌面环境

1.1.1.1 桌面的使用

1、Mate 桌面的面板位于屏幕下方，包括开始菜单、火狐浏览器、MATE 终端、Caja 文件管理器、网络以及声音等设置。也可以在开始菜单选择要启动的程序即可打开使用，如图 1.4 所示。您还可以直接点击快速启动栏上的程序，以及将应用程序添加到快速启动栏。

图 1. 4 开始使用系统

2、桌面切换位于屏幕下方，您可以将不同的会话分别放置在各个桌面上，直接点击屏幕右下角的图标即可进行桌面切换。默认设置为 4 个桌面，您还可以自己更改数量。

1.1.1.2 控制会话

1、用户名和口令通过登录进程的鉴定后，将启动会话管理器。通过会话管理器，可以保存每个会话的设置。还可以保存最近一次的会话状态，这样，在下次登录时就能返回该状态。

2、会话管理器可以保存并恢复以下设置：

- 外观和行为设置，如字体、颜色和鼠标设置
- 运行的一些应用程序，如文件管理器

> 重要
>
> 系统不能保存或恢复会话管理器没有进行管理的应用程序。例如，如果通过终端窗口中的命令行启动 vi 编辑器，则会话管理器不能恢复您的编辑会话。

1.1.1.3 锁定屏幕

1、要锁定屏幕，请执行以下一项操作：

- 从启动栏中选择“开始”——“锁住屏幕”，如图 1.5 所示。

图 1. 5 锁定屏幕

- 使用键盘快捷方式。通常为 Ctrl + Alt ~ L 。

锁定屏幕时会启动屏幕保护程序。要解除屏幕锁定，请移动鼠标以显示锁定的屏幕对话框。输入用户名和口令，然后按 Enter 键，如图 1.6 所示

图 1. 6 解除屏幕锁定

1.1.2 从控制台登录

1、系统启动后，用户可以得到登录提示：

localhost login:

输入用户名，按 Enter 键，系统提示输入密码：

Password:

输入正确的密码并按 Enter 键后，该用户将登录到系统，显示如：

```
username@localhost:~ >
```

> 注意
> 输入密码的过程是不显示的，也就是说您会看到屏幕上一点反应也没有，这也是本系统安全的原因之一，为的是保持用户密码的安全，输入密码后按 Enter 键即可。

提示

这里的 username 指您实际登录的用户名 ,localhost 是您的主机名，在安装时可以设置，默认设置为 Rocky 。如 : 在计算机 linx 上使用 testuser 用户登录，输入正确的密码后，系统将提示“ testuser@linx:~ >”。

2、若密码输入错误该用户不存在，系统将提示登录失败，需重新登录:

Login incorrect

localhost login:

3、您可以重新尝试登录，但是有次数限制，如果达到预定义次数的非成功尝试，系统将在一段时间间隔后才允许重新登录。预定义次数和时间间隔由授权管理员设置。

4、用户还可以在其它情况下登录到凝思安全操作系统。

- 使用 Ctrl + Alt + F1 ～ F6 切换到不同的控制台，用不同的用户登录。
- 使用命令 su username 从当前用户切换到用户 username 登录。

  如：当前用户为 testuser1，输入命令 su testuser2 或 su - testuser2，输入用户 testuser2 的口令后，从用户 testuser1 切换到用户 testuser2。

testuser1@localhost:~ >su testuser2 Passowrd:

testuser2@localhost:/home/testuser1 >

注意

使用 su - username 切换用户登录后会切换到用户 username 的主目录，而使用 su username 则不会。

1.2 文档和帮助

用户使用系统过程中，可以使用 man 命令查看各命令的手册页，或使用 info 命令查看各信息页。

例如，要阅读 ls 命令的手册页，在 shell 提示下输入 man ls（或输入 info ls 来阅读信息页）。要关闭手册页或信息页，按 q 。

1.3 注销

1.3.1 从桌面注销

1、在桌面环境下，您不再使用计算机时，您可以注销并让系统保持运行。要注销并保持系统运行，请执行以下操作：

- 单击面板上的“开始”，从菜单中选择［注销］，如图 1.7 所示。

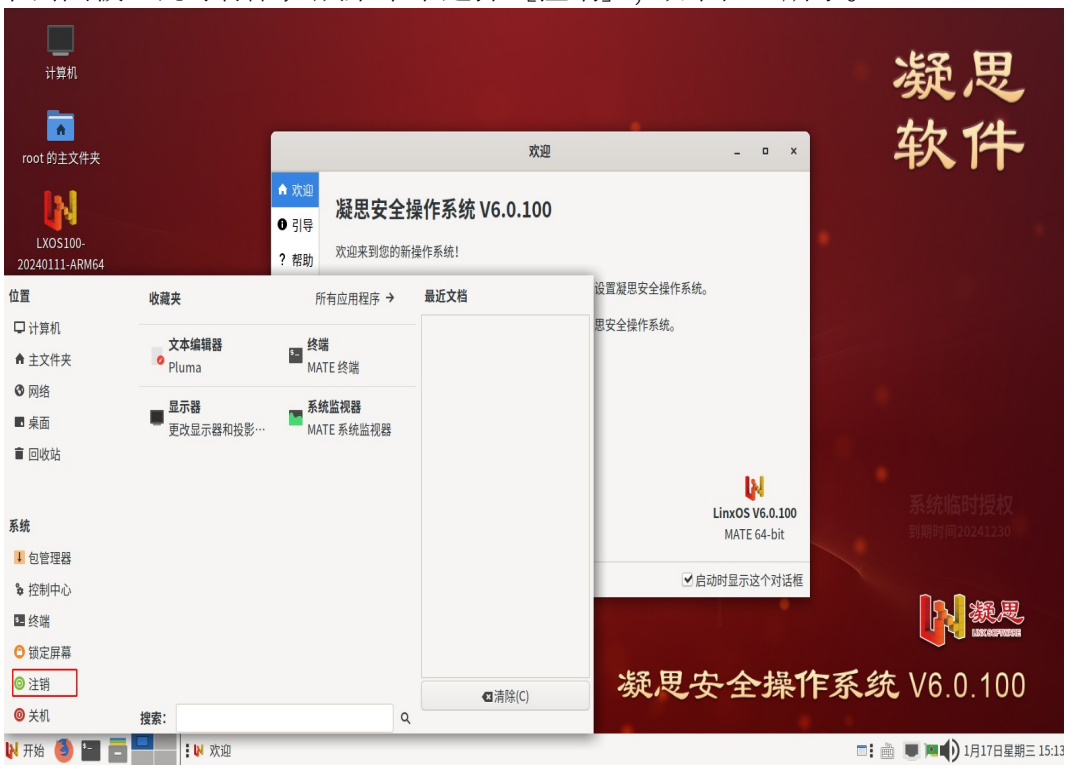


图 1. 7 从桌面注销

- 点击“注销”，如图 1.8 所示：

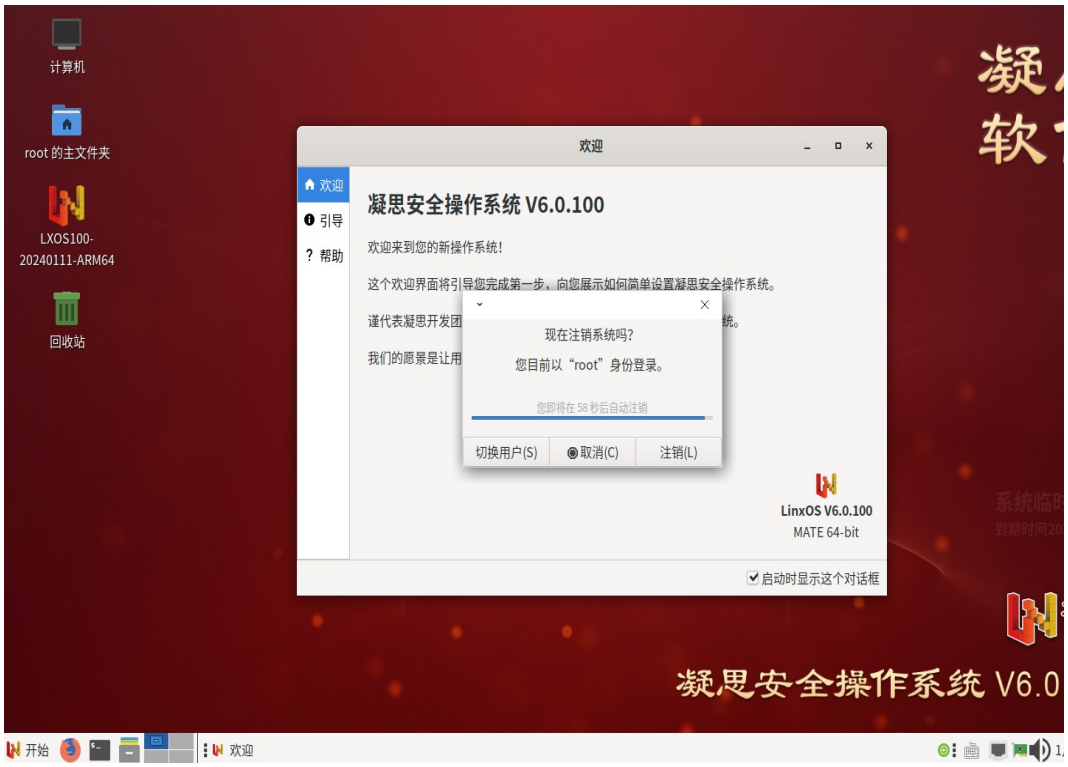


图 1. 8 注销

1.3.2 从控制台注销

1、在控制台中，输入 logout，该用户将注销登录，系统返回登录提示：

login:

用户需重新登录才能使用系统。

第 2 章 系统基本使用

尽管图形用户界面对 Linux 已经越来越重要，但是使用鼠标并不总是处理日常工作的最佳方式。命令行不仅高度灵活而且效率较高。本章主要是对 Bash shell 基本使用的简要介绍，解释 Linux 中的用户权限概念，并提供较重要的命令列表。

要控制文本模式下的计算机，基于文本的应用程序尤为重要。这种情况下需要使用虚拟控制台。有六个控制台可以在文本模式下使用。对应的组合键分别为 Alt + F1 ～ F6 。第七个控制台是为 X 窗口系统保留的。

2.1 Bash 简介

用户登录后进入控制台，该控制台通常运行 Bash（Bourne again shell，该程序是 GNU 项目中的一部分）。运行该 shell 后，请查看第一行的提示，通常有用户名、主机名和当前路径组成。但可对其进行自定义。当光标移到该提示后面时，您可以直接向所在计算机系统发送命令。

2.1.1 命令

一条命令包含若干个元素。第一个元素总是真正的命令，随后是参数或选项。按 Enter 键即可执行命令。在此之前，您可以很容易地编辑命令行、添加选项或更正输入错误。ls 是一个最常用的命令，该命令可以使用参数，也可以不使用。在控制台中只输入 ls 命令将显示当前目录的内容。

选项以连字符为前缀。例如，命令 ls -l 将显示同一目录中内容的详细信息。在每个文件名后，都可以看到文件的创建日期、以字节表示的文件大小和下文要涉及的其它详细信息。--help 是许多命令都有的非常重要的选项。输入 ls --help 可以显示 ls 命令的所有选项。

也可以使用 ls 命令查看其它目录下的内容。为此，必须将该目录指定为参数。例如，要查看/home 下的内容，应输入 ls -l /home。

2.1.2 文件和目录

要高效使用 shell，掌握一些关于 Linux 系统的文件和目录结构的知识将有很大帮助。您可以将目录视为存储文件、程序和子目录的电子文件夹。层次中的顶级目录是根目录，用 / 表示。从此目录可以访问其它所有目录。/home 目录包含用于存储个人用户私人文件的目录。目录树如图 2.1 所示。

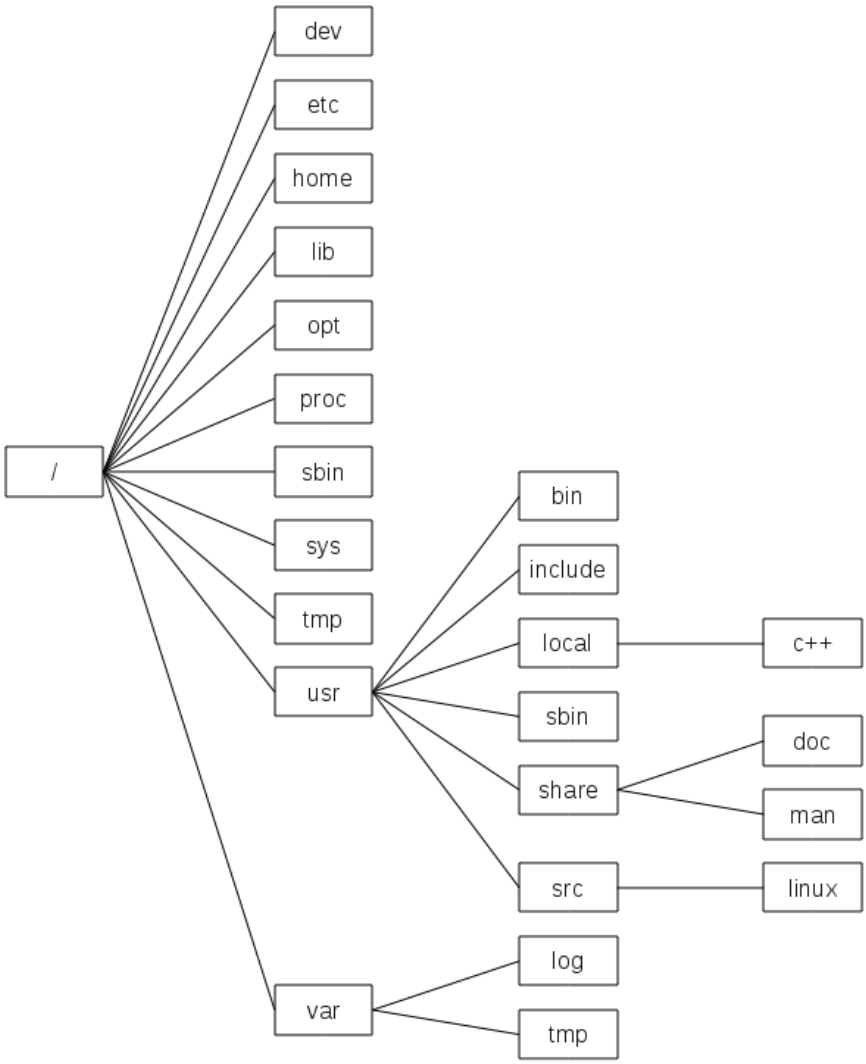


图 2. 1 目录树

以下简要说明了 Linux 中的标准目录。

- / 根目录，目录树的起点
- /bin 在引导过程初期需要的程序
- /dev 代表硬件组件的设备文件
- /etc 重要的系统配置文件
- /etc/init.d 脚本
- /home 用户目录
- /lib 共享库（供动态链接程序使用）
- /opt 选件，大型装载程序包（如 KDE、GNOME、Netscape）
- /proc 进程文件系统
- /sbin 为系统管理员保留的用于引导的程序
- /sys “system”文件系统，在其中收集内核的所有设备信息
- /tmp 临时文件

- /usr 所有应用程序
- /usr/bin 通常可访问的程序
- /usr/include C 编译器的头文件
- /usr/include/c++ C++编译器的头文件
- /usr/local 本地的、与发布版无关的扩展
- /usr/sbin 为系统管理员保留的程序
- /usr/share/doc 各种文档文件
- /usr/share/man 系统手册页
- /usr/src 系统软件的源代码
- /usr/src/linux 内核源代码
- /var 系统正常运行时所改动的数据
- /var/log 系统日志文件
- /var/tmp 临时文件

2.1.3 Bash 功能

Bash shell 提供的两种重要功能可以让您的工作变得很简单：

- 历史记录

  要重复以前输入的命令，请按 ↑ 键，直到先前命令在提示符处出现。按 ↓ 键可以在先前输入的命令列表中前移。要编辑命令行，只需使用箭头键将光标移至所需位置并开始键入。使用 Ctrl + R 可在历史记录中搜索。

- 展开

  在键入文件名的第几个字母后，展开完整的文件名，直至它可以被唯一标识。只需在键入前几个字母后按 Tab 键即可实现此功能。如果有多个文件名的前几个字母都相同，则按两次 Tab 键可获取这些文件的列表。

2.1.4 指定路径

处理文件或目录时，指定正确的路径十分重要。不过，您不必输入从主目录到各个文件的完整（绝对）路径。您可以从当前目录开始指定。直接用 ～ 来表示主目录。例如主目录下有个 test 目录，那么有两种方法可以列出 test 目录中的文件：用 ls text 输入相对路径或用 ls ~/test 指定绝对路径。

要列出其他用户主目录的内容，请输入 ls ~username。例如某个用户是 testuser，这样，使用 ls ~testuser 就会列出 testuser 主目录的内容。

用一个圆点来表示当前目录。当前目录的上一级目录用两个点来表示。输入 ls ..可以查看当前目录的父目录的内容。命令 ls ../.. 可用于比当前目录高两个级别的目录的内容。

2.1.5 通配符

Shell 的便捷之处还体现在通配符上。Bash 提供四种不同的通配符：

- ? 完全匹配任一字符。
- * 匹配任意数目的字符。
- [set] 匹配在方括号中指定的字符组中的任一字符，这里用字符串 set 表示字符组。
- [!set] 匹配 set 标识字符之外的任一字符。

假定 test 目录包含文件 testfile、testfile1、testfile2 和 datafile， 使用命令 ls testfile?则可以列出文件 testfile1 和 testfile2。使用 ls testfile*，列表还将包括 testfile。ls *file*会显示所有示例文件。最后，您可以使用 set 通配符表示所有末尾字符为数字的示例文件： ls testfile[1-9]。

四个通配符中匹配范围最广的是星号。使用它可以将某个目录内的所有文件复制到另一个目录，或通过一个命令删除所有文件。例如，使用命令 rm *file* 可以删除当前目录中文件名包含字符串 file 的所有文件。

2.1.6 less 和 more

Linux 包含两个直接在 shell 中查看文本文件的小程序。不必启动编辑器来阅读 Readme.txt 之类的文件，只需输入 less Readme.txt 即可在控制台窗口中显示其中的文本。使用 Enter 键可以向下滚动一行，使用 Space 键可以向下滚动一页，使用 ↑ 和 ↓ 键可以使文本按行上下滚动，使用 Page Up 和 Page Down 键可以使文本按页上下滚动，要退出 less，请按 Q 。

除了使用 less 之外，您还可以使用 more 这种较早的程序。不过，该程序使用起来不太方便，因为它不允许向上滚动文本，只能使用 Enter 或 Space 键向下滚动一行或一页。

less 程序得名于 less is more（少即是多）原则，并且还可用来方便查看命令输出。要了解该程序的这种功能，详见下一节“管道”。

2.1.7 管道

通常，shell 的标准输出界面是您的屏幕或控制台窗口，而标准输入设备是键盘。要将命令的输出转发到像 less 这样的应用程序，就需要使用管道。

要查看 text 目录下的文件，请输入命令 ls test | less。test 目录的内容将使用 less 来显示。这只是在 ls 命令正常输出过长时才有意义。例如，当您使用 ls /dev 命令查看 dev 目录的内容时，您只能在窗口中看到一小部分。而使用 ls /dev | less 命令则能够查看整个列表。

也可以将命令的输出保存到文件中。例如:ls test > content 会生成一个名为 content 的新文件，其中包含 test 中的文件和目录的列表。用 less content 命令可以查看该文件。

您也可以将文件作为命令的输入。例如，用 sort < testfile 命令对 testfile 中的文本行进行排序。sort 命令的输出被发送到屏幕。文本按各行的首字符进行排序，如果两行的首字符相同，该命令将继续比较这两行的下一字符，如果还相同，将继续进行比较。

如果需要用一个新文件来包含排序列表，则需要将 sort 命令的输出用管道输出至一个 文件。要测试该命令，在编辑器中创建一个未排序的名称列表，并在命名为 list 后保存到 test 目

录中。然后转至 test 目录并输入命令 sort < unsortedlist > sortedlist。最 后，使用 less 查看经过排序的列表。

标准错误输出和标准输出一样，都发送至控制台。不过，要将标准错误输出重定向到名为 errors 的文件，则需要在相应命令中追加 2 > errors。如果追加的是 >& alloutput，标准输出和标准错误都将保存到名为 alloutput 的文件中。最后，要将命令输出结果追加到现有文件，该命令后面必须跟有 >> 而不是单个>。

提示

在 Linux 中， shell 为每个命令自动打开 3 个文件描述符，分别是命令的输入、输出和错误信息文件。每个打开的文件都有一个小的整数与之对应，称为文件描述符。标准输入、标准输出、标准出错的文件描述符分别是 0 、 1 、 2 。

2.1.8 存档和数据压缩

您已经创建了一些文件和目录，现在该考虑一下存档和数据压缩的问题了。假定您想将整个 test 目录打包在一个文件中，以便备份到软盘或通过电子邮件发送。要执行该操作，请使用命令 tar（代表 tape archiver，即磁带存档程序）。使用 tar --help 可查看 tar 命令的所有选项。下面对最重要的一些选项进行了说明:

- -c （代表 create）创建新档案。
- -t （代表 table）显示档案中的内容。
- -x （代表 extract）对档案解包。
- -v （代表 verbose）创建档案时在屏幕上显示所有文件。
- -f （代表 file）为档案文件选择一个文件名。创建档案时，此选项总应放在最后。

要将 test 目录下的所有文件和子目录打包到名为 testarchive.tar 的档案中，请使用选项-c 和-f。尽管不是必需的，但出于测试的目的，请同时添加-v 选项，以便跟踪存档过程。在使用 cd 命令转至 test 目录所在的主目录后，输入 tar -cvf testarchive.tar test。之后，可使用 tar -tf testarchive.tar 查看档案文件的内容。test 目录及其所有文件和目录都在您的硬盘上保持不变。要对档案解包，请输入 tar -xvf testarchive.tar。

对于文件压缩，常用的 gzip 程序当然是 Linux 系统的首选。只需输入 gzip testarchive.tar 即可。通过 ls，您可以看到文件 testarchive.tar 已不复存在，取而代之的是文件 testarchive.tar.gz。这个文件要小得多，因此也更适于通过电子邮件传送或存储到软盘上。

现在，将该文件解包到先前创建的 test2 目录中。这需要输入 cp testarchive.tar.gz test2 将文件复制到该目录中。使用 cd test2 转至该目录。扩展名为 .tar.gz 的压缩档案可用 gunzip 命令解压缩。输入 gunzip testarchive.tar.gz 将生成文件 testarchive.tar，然后还需使用 tar -xvf testarchive.tar 命令抽取或 untar 操作。您也可以通过添加 -z 选项一步完成解压和抽取压缩档案。完整的命令为 tar -xzvf testarchive.tar.gz。通过 ls，您会看到新建的 test 目录，其内容与主目录中的 test 目录的内容完全相同。

2.2 用户和访问控制

1、访问控制是网络安全防范和保护的主要核心策略，它的主要任务是保证网络资源不被非法使用和访问。访问控制规定了主体对客体访问的限制，并在身份识别的基础上，根据身份对提出资源访问的请求加以控制。它是对信息系统资源进行保护的重要措施，也是计算机系统最重要和最基础的安全机制。

访问控制的基本概念有:

- 主体（Subject）

  主体是指主动的实体，是访问的发起者，它造成了信息的流动和系统状态的改变，主体通常是进程。

- 客体（Object）

  客体是指包含或接受信息的被动实体，客体在信息流动中的地位是被动的，是处于主体的作用之下，对客体的访问意味着对其中所包含信息的访问。客体通常是用于信息共享、存储和通讯的系统资源（如文件、目录、套接字、共享内存等）。

- 访问（Access）

  访问（Access）是使信息在主体（Subject）和客体（Object）之间流动的一种交互方式。

- 访问权限（Access Permissions）

  访问权限决定了谁能够访问系统，能访问系统的何种资源以及如何使用这些资源。适当的访问权限能够阻止未经允许的用户有意或无意地获取数据。访问权限的手段包括用户识别代码、口令、登录控制、资源授权（例如用户配置文件、资源配置文件和控制列表）、授权核查、日志和审计等等。

2、凝思安全操作系统是一种多用户系统，它支持任意数目的用户同时操作。用户在自己的 工作站上启动会话之前必须先登录到系统中。每个用户都有一个用户名及对应的口令。设置 用户名和口令可以确保未经授权的用户无法查看他们无权查看的文件。进行这种限制后，普 通用户不可能对系统进行改动（如安装新程序），只有被授权的用户（如管理员）才能对系统进行更改或访问文件，而设置了强制访问控制后，即使是管理员也不可能访问所有的文件。下面章节只描述一般情况（未设置强制访问控制），关于凝思安全操作系统的强制访问控制等安全策略，详见 10。

2.2.1 文件系统权限

一般而言，Linux 文件系统中的每个文件都属于某个用户和某个组。可以为这些专有组和其它所有组授予读、写或执行这些文件的权限。

在这种情况下，可以将组定义为具有特定集合权限的一组相互连接的用户。例如，可以将共同处理某个项目的组称为 project1。Linux 系统中的每个用户都是至少一个专有组（通常是 users）的成员。可以根据需要设置系统中组的数目，但只有管理员才能添加组。每个用户都可以使用 groups 命令查出自己所属的组。

- 文件访问

文件系统中的权限组织结构不同于文件和目录的组织结构。使用 ls -l 命令可以显示文件权限信息。命令输出可能如下所示:

```
-rw-r-----    1    testuser1    project1    14197    Jun 21 15:03    testfile
```

如第三列中所示，此文件属于用户 testuser1。该文件被指派给组 project1。要确定 testfile 文件的用户权限，必须仔细检查第一列。

```
-      rw-      r--      ---
类型   用户权限  组权限    其他用户的权限
```

此列含有一个前置字符，后接九个字符，每三个字符为一组。这十个字符中的第一个字符代表文件系统组件的类型。连字符（-）表示这是一个文件，d 表示目录、l 表示链接、b 表示块设备，c 表示字符设备。

后面的三组字符遵循标准模式。第一个字符表示该文件可读（r）还是不可读（-），第二个字符表示该文件可写（w）还是不可写（-），第三个字符表示该文件可执行（x）还是不可执行（-）。

在本例中，作为文件 testfile 的拥有者，testuser1 有权读（r）写（w）该文件，但无法执行它（x）。project1 组中的成员可以读取该文件，但不能修改或执行 它。其他用户无权访问此文件。通过 ACL（Access Control List，访问控制列表）可以指派其它权限。

- 目录权限

目录的访问权限类型用 d 来表示。对目录而言，各种权限的含义稍有不同。

```
drwxrwxr-x    1    testuser1    project1    35    Jun 21 15:15    ProjectData
```

在上例“显示目录权限的示例输出”中，很容易识别出目录 ProjectData 的拥有者（testuser1）和所属组（project1）。与之前的文件访问权限相比，设置读权限（r）表示可以显示该目录的内容。写权限（w）表示可以在该目录下创建新文件或删除文件。执行权限（x）表示用户可以进入该目录。上例中的输出表示用户 testuser1 及组 project1 中的成员可以转到 ProjectData 目录（x）、查看其中的内容（r）并添加或删除文件（w）。其他用户的权限则受到限制。他们可以进入目录（x）并浏览其中的内容（r），但不能创建或删除任何文件（w）。

2.2.2 修改文件权限

- 更改访问权限

文件或目录的访问权限可以由拥有者更改，当然也可以由管理员更改；更改时要使用命令 chmod，后接更改权限的参数及一个或多个文件名。参数可归为四类:

1、用户相关参数

- u（user）：文件的拥有者。

- g（group）：与文件属主有相同组 ID 的所有用户。
- o（others）：其他用户。

> 注意
> 如果未指定用户参数，访问权限的更改将应用到所有用户。

2、用于增加（+）、删除（-）或设置（=）的字符。

3、缩写

- r：读
- w：写
- x：执行

4、一个文件名或由空格分隔的多个文件名

例如，在“显示目录权限的示例输出”中，如果用户 testuser1 还想授予其他用户写入（w）目录 ProjectData 的权限，则可以使用命令 chmod o+w ProjectData 执行该操作。

不过，如果该用户不希望任何用户具有写权限（其本人除外），则可以输入命令 chmod go-w ProjectData 执行该操作。要防止所有用户向 ProjectData 添加新文件，请输入 chmod -w ProjectData。此时，如果不重新设置写权限，即使是拥有者也无法再写入该文件。

- 更改所有权

另有一些重要的命令可用来控制文件系统组件的所有权和权限，这些命令包括 chown（更改拥有者）和 chgrp（更改组）。使用命令 chown 可将文件所有权转让给另 一用户。不过，只有 root 才有权执行该操作。

假定“显示目录权限的示例输出”中的文件 testfile 不应再属于 testuser1，而应属于用户 testuser2，则 root 应该输入 chown testuser2 testfile。

chgrp 用于更改文件的组所有权。不过，文件的拥有者必须是新组的成员。这样，使用命令 chgrp project2 ProjectData，“显示文件权限的示例输出”中的用户 testuser1 即可将文件 ProjectData 所属的组改换为 project2，只要该用户是这个新组的成员。

2.2.3 setuid 位

在某些情况下，访问权限可能过于严格。因此，Linux 另有一些设置，允许为执行特定操作临时更改当前用户和组标识。例如，passwd 程序通常要求拥有根权限才能访问/etc/passwd。此文件包含一些重要信息，如用户主目录及用户和组 ID。因此，普通用户将无法更改 passwd，因此授予所有用户直接访问此文件的权限太过危险。解决该问题的一种可行方案就是 setuid 机制。setuid（设置用户 ID）是一个特殊的文件特性，当一个程序被设置了该标记以后，运行该程序的进程将拥有该程序所有者同样的权限。以 passwd 命令为例：

```
-rwsr-xr-x   1   root shadow  80036   2004-10-02 11:08      /usr/bin/passwd
```

您可以看到为用户权限设置了 s 位。通过设置 setuid 位，启动 passwd 命令的所有用户都以 root 身份执行该命令。

2.2.4 setgid 位

setuid 特性适用于用户，setgid 特性则适用于组。进程在执行设置了 setgid 位的文件时，将拥有其文件所有组同样的权限。例如：

drwxrws--- 2 testuser1 archive 48 Nov 19 17:12 backup

您可以看到为组权限设置了 s 位。目录的拥有者和组 archive 的成员可以访问此目录。 不是该组成员的用户将被“映射到”相应的组；所有写入文件的有效组 ID 将为 archive。例如，以组 ID archive 运行的备份程序即便没有根权限也能访问此目录。

2.2.5 粘滞位

另外还可以设置粘滞位。

粘滞位最初的目的是“在 swap 中保存文本”,即以这种方式标记的文件将被装入 RAM，而不必在每次使用时从硬盘读取，从而提高命令处理的性能。现在 UXIN 系统已经不再使用这个功能，但“粘滞位”这个名称被保留。

如果为目录设置了粘滞位，则该目录下的文件只能由 root、该目录的所有者或该文件的所有者删除。典型示例如 /tmp 目录：

```
drwxrwxrwt   2  sys  sys  4096  Oct 21 17:15  tmp
```

2.2.6 自主访问控制

1、访问控制涉及的领域很广，方法也很多，通常访问控制策略可以划分为自主访问控制和强制访问控制：

- 自主访问控制（Discretionary Access Control）
- 强制访问控制（Mandatory Access Control）（详见 10）

2、自主访问控制（DAC）是在确认主体身份及所属组的基础上，根据访问者的身份和授权来决定访问模式，对访问进行限定的一种控制策略。所谓自主，是指具有被授予某种访问权力的用户能够自己决定是否将访问控制权限的一部分授予其他用户或从其他用户那里收回他所授予的访问权限。使用这种控制方法，用户或应用可任意在系统中规定谁可以访问它们的资源，这样，用户或用户进程就可有选择地与其他用户共享资源。它是一种对单独用户执行访问控制的过程和措施。

3、Linux 系统文件对象（如文件或目录）的传统权限概念通过 ACL（访问控制列表）得到了进一步扩展。通过访问控制列表可以将权限指派给文件系统对象的各个用户或组，而不再局限于最初的拥有者或所属组。

4、在凝思安全操作系统 V6.0.100 中，任何用户都可以使用自主访问控制。

- 文件创建时具有缺省自主访问控制属性

  创建客体时，新建客体的文件所有关系应为创建者，且文件访问权限为默认方式。

  在新建文件的时候，通常使用 666 作为默认许可位；在新建程序的时候，通常使用 777 作为默认许可位。实际的文件访问权限由 umask 按位取反再和默认许可按位取

与而得到。例如，默认许可位 666，umask 077，则实际的文件访问权限应该是 600，即文件所有者有读写权限，其他无权限。用户可以执行 umask 命令查看和更改当前的 umask，/etc/login.defs 里的 umask 定义了一个系统范围内的默认设置。

如：以 testuser1 用户身份登录，umask 为 022，运行 touch test.txt 新建文件 test.txt。用 ls –l test.txt 查看 test.txt 文件的属性，test.txt 的所有关系应为：testuser1.users，文件属性应为：rw-r—r--

- 自主访问控制的访问权限设定

  用户能根据属主/属组/其他关系设置客体的读/写/执行权限。

  属主（user）：文件拥有者。

  属组（group）：文件拥有者的用户组。

  其他（other）：除了属主及其同组成员的其他用户。

  如： 以 testuser1 用户身份登录，创建文件 test.tmp，将文件的属性设置为 600。用户 testuser1 能够读写但不能执行文件 test.tmp，其它用户不能访问该文件。

- 属主可访问其拥有的客体

  用户能够访问属于自己的文件。

  如：以 testuser1 用户身份登录，创建文件 test.txt，用户 testuser1 将能够读写 test.txt 文件。

- 属主可更改其拥有的客体的组所有权

  普通用户不能将其拥有的客体所有权转让给其他用户，但可以将其组所有权转让给具有指定组名的组。命令使用如下：

  chgrp [option(s)] groupname file(s)

  -R 更改所有子目录中的文件和目录。

- 属主可改变客体的访问权限

  客体属主可以使用 chmod 命令改变客体的访问权限。

  如：以 testuser1 用户身份登录，建立文件 test.txt 并将其属性改为 660（同组用户可读写，其它用户不能访问）。用 ls 查看 test.txt 的属性，应为 660，以 testuser1 不同组的用户身份 testuser2 登录，将不能访问 test.txt 文件。

```
$ touch test.txt
$ ls -l test.txt
-rw-r--r--    1 testuser1 users 0 Sep 4 06:27 test.txt

$ chmod 660 test.txt
$ ls -l test.txt
-rw-rw----    1 testuser1 users 0 Sep 4 06:27 test.txt

login:
testuser2
passwd:

$ cat test.txt
cat: test.txt: Permission denied
```

2.2.6.1 访问控制列表

传统类 UNIX 系统以文件属主、组和其它方式设置读、写和执行权限，这会造成文件访问权限的不必要扩散。例如，如果想让某个非同组的用户能够访问一个文件，就只能让所有人都能访问这个文件，这是非常不安全的，也是非常不必要的。

访问控制列表（Access Control List,ACL）提供细粒度的访问控制，能够以特定用户或特定组为单位的访问权限分配，从而防止文件访问权限的不必要扩散。利用访问控制列表机制，我们只要把文件共享给特定用户就可以了，而无须让所有人都能够访问该文件。

使用一个简单的 ls -l 命令即可检测到拥有扩展访问权限的文件或目录:

```
-rw-r--r--+   1  testuser1  project1  14197  Oct 21 15:03 testfile
```

与不带 ACL 的文件的 ls 输出相比，上述 ls 的输出并没有显著的不同。testfile 由属于组 project1 的 testuser1 拥有。testuser1 对此文件具有读写权限，而其所属的组及其他所有用户均具有读权限。带 ACL 和不带 ACL 的文件之间唯一显示的区别即在于：前者的第一列内多了一个用来包含权限位的+。

使用命令 getfacl 可以显示文件名、属主、所属组和访问控制列表（ACL），如果一个目录拥有默认的 ACL，getfacl 也将显示默认的 ACL，非目录文件不能拥有默认的 ACL。

通过执行 getfacl testfile 来获取 ACL 的详细信息:

```
1  # file: testfile
2  # owner: testuser1
3  # group: project1
4  user::rw-
5  user:testuser2:rw-        Effective:
r--
6  group::r--
7  group:project2:rw-
8  mask::r--
9  other::r--
```

输出中的前三行不包含 ls -l 中没有的信息。这些行只用来表明文件名、拥有者和所属组。第 4 到 9 行包含 ACL 条目。传统访问权限只是使用 ACL 可以授予的权限的一部分。本例中的 ACL 为该文件的拥有者以及用户 testuser2 授予了读写权限（第 4 到 5 行）。通过为其他用户授权，传统概念得到了扩展。这种权限的扩展也同样适用于组权限的处理。所属组拥有读权限（第 6 行），且 project2 组拥有读写权限。第 8 行中的 mask 条目将用户 testuser2 和组 project2 的有效权限限制为只能读取。其他用户和组对该文件不具备任何类型的权限（第 9 行）。

使用命令 setfacl 可以设置文件的访问控制列表。如:

- 以普通用户 testuser 身份登录系统，在/tmp 目录下创建文件 aaa，并将其 DAC 属性设置为 600，即除自己以外不允许任何用户进行读写:

```
$ touch /tmp/aaa
$ chmod 600 /tmp/aaa
```

  - 以普通用户 testuser1 身份登录系统，对 /tmp/aaa 文件无法进行读或写操作。
  - 以普通用户 testuser2 身份登录系统，对 /tmp/aaa 文件无法进行读或写操作。

- 以普通用户 testuser1 身份将 /tmp/aaa 文件的 ACL 属性设置为 testuser1 用户可以进行读写:

```
$ setfacl -m u:testuser1:rw /tmp/aaa
```

  - 以普通用户 testuser1 身份可以对 /tmp/aaa 文件进行读或写操作。
  - 以普通用户 testuser2 身份对 /tmp/aaa 文件仍然无法进行读或写操作。

重要

只有文件的属主和拥有 CAP_FOWNER 能力的进程拥有修改文件 ACL 设置的权力，只有能够读取访问文件的进程才能读取文件的 ACL 设置，类似于传统 UNIX 文件访问权限的要求。 CAP_FOWNER 能力由管理员设置。

2.3 重要的 Linux 命令

本节深入探讨 Linux 系统中较为重要的命令，伴随各个命令列出了参数，并且适当的时候还给出了典型的示例应用程序。有关各个命令的详细信息，请使用 man，并在后面键入命令名称来查看其手册页，例如 man ls。

在参考手册页中，用 Page Up 和 Page Down 可以上下移动，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。使用 man man 可以了解有关 man 命令本书的更多信息。

本节所列命令只是众多命令中的一小部分。有关其它命令的信息或更详细的信息，建议您参考 O'Reilly 出版的《Linux in a Nutshell》。下面的概述中使用不同的字体来表示各个命令元素。

命令及其必需选项始终显示为命令选项，需要指定的内容或非必需参数均放在[方括号]中。

实际使用时通常可以将几个参数组合起来，例如用 ls -la 来代替 ls -l -a。

2.3.1 文件命令

本小节将列出较重要的文件管理命令。它包括从总体文件管理到文件系统 ACL 操纵的所有文件管理命令。

1、文件管理

- ls [option(s)] [file(s)]

  如果运行 ls 时未附加任何参数，程序将以缩写格式列出当前目录中的内容。

  -l 详细列表。

  -a 显示隐藏文件。

- cp [option(s)] source target

  将 source 复制到 target。

  -i 在覆盖现有 target 之前等待确认（如果需要）。

  -r 递归复制（包含子目录）。

- mv [option(s)] source target

  将 source 复制到 target，然后删除原始 source。

  -b 在移动 source 之前创建该文件的备份副本。

  -i 在覆盖现有 targetfile 之前等待确认（如果需要）。

- rm [option(s)] file(s)

  从文件系统中删除指定文件。除非使用选项 -r，否则不能使用 rm 删除目录。

  -r 删除所有现有子目录。

  -i 在删除各个文件之前等待确认。

- ln [option(s)] source target

  创建从 source 到 target 的内部链接。通常这种链接直接指向同一文件系统上的 source。但是，如果执行带-s 选项的 ln 命令，则可以创建一个符号链接，仅指向

source 所在的目录，支持跨文件系统的链接。

-s 创建符号链接。

- cd [option(s)] [directory]

  更改当前目录。执行不带任何参数的 cd 命令将转到用户主目录。

- mkdir [option(s)] directoryname

  创建新目录。

- rmdir [option(s)] directoryname

  删除指定的目录（如果该目录已清空）。

- chmod [option(s)] mode file(s)

  更改访问权限。

  mode 参数由三个部分构成：group、access 和 access type。

  group 接受以下字符：

  u 用户

  g 组

  o 其它

  对于 access，用+可以授予权限，用-可以拒绝授予权限。

  access type 受以下选项控制：

  r 读

  w 写

  x 执行文件或访问目录。

  s 设置 uid 位——当一个程序被设置了该标记以后，运行该程序的进程将拥有该程序所有者同样的权限。

  您也可以使用数字模式来设置访问权限。数字模式是 1 到 4 个八进制数，每个数由位权为 4、2、1 的 3 位叠加而来。省略的数字缺省设置为 0。第一位用来设置 suid（4）、sgid（2）、粘滞位（1）；第二位设置文件所有者的权限：可读（4）、可写（2）、可执行（1）；第三位设置文件所有者所在组其他用户的权限，设置同第二位；第四位设置其他组的用户的权限，设置同第二位。

- gzip [parameter(s)] file(s)

  此程序使用复杂的数学算法压缩文件内容。以这种方式压缩的文件的扩展名为.gz，而且使用前需解压缩。要压缩若干文件甚至是整个目录，请使用 tar 命 令。

  -d 将打包的 gzip 文件解压缩，使其恢复原始大小，并且能够正常处理（类似命令 gunzip）。

- tar option(s) archive file(s)

  tar 将一个或多个文件放入档案。压缩是可选操作。tar 是相当复杂的命令，可以附带若干选项，最常用的选项如下：

  -f 将输出结果写入文件，而不是显示在屏幕上。

  -c 创建新的 tar 档案。

  -r 将文件添加到现有档案中。

  -t 输出档案内容。

-u 添加文件，但仅适用于文件比档案中已有的文件更新的情况。
-x 将档案中的文件解包（抽取）。
-z 用 gzip 将生成的档案打包。
-j 用 bzip2 压缩生成的档案。
-v 列出已处理的文件。

由 tar 创建的档案文件以.tar 结尾。如果这个 tar 档案还使用 gzip 进行了压缩，则以.tgz 或.tar.gz 结尾。如果是使用 bzip2 压缩的，则以.tar.bz2 结尾。

- locate pattern(s)

使用 locate 命令可以查找指定文件所处的目录。如果需要，可使用通配符来指定文件名。该程序的速度非常快，因为它使用专为此目的创建的数据库（而不是搜索整个文件系统）。但恰恰是这一点也带来了一个重大缺陷：无法找到在最后更新文件数据库后创建的任何文件。以管理员身份使用 updatedb 可以生成该数据库。

- updatedb [option(s)]

此命令可以对 locate 使用的数据库进行更新。要包含所有现有目录中的文件，请以管理员身份运行程序。最好通过追加与号（&）。此命令通常作为 daily cron 作业运行。

- find [option(s)]

使用 find 可以在指定目录中搜索文件。第一个参数指定搜索的起始目录。选项 -name 后面必须紧跟搜索字符串，字符串中也可以包含通配符。与使用数据库的 locate 不同，find 扫描的是实际目录。

2、用于访问文件内容的命令

- cat [option(s)] file(s)

cat 命令用于显示文件的内容，使用它可以将所有内容连续打印输出到屏幕上。

-n 在左侧输出编号。

- less [option(s)] file(s)

此命令可用于浏览指定文件的内容。使用 Page Up 和 Page Down 可以向上或向下滚动半屏，使用 Space 可以向下滚动一整屏。使用 Home 和 End 至文件的开头和结尾。按 Q 可以退出程序。

- grep [option(s)] searchstring filenames

grep 命令用于在指定 file(s)中查找特定的搜索字符串。如果找到搜索字符串，该命令将显示找到的 searchstring 所在的行及文件名。

-i 忽略大小写。
-H 为每个匹配打印文件名。
-n 另外显示含有匹配项的行的编号。
-l 只列出其中不含 searchstring 的文件。

- diff [option(s)] file1 file2

diff 命令用于比较两个文件的内容。该程序生成的输出将列出所有不匹配的行。

这是只需发送程序变更而不是全部源代码的编程人员经常使用的命令。

-q 只报告两个文件是否不同。

-u 生成一个“统一”的 diff，从而增加输出的可读性。

2.3.2 系统命令

本节列出了用于检索系统信息以及进程和网络控制的几个较重要的命令。

1、系统信息

- df [option(s)] [directory]

df（可用磁盘）命令如不与任何选项一同使用，则可以显示磁盘空间容量、当前占用磁盘空间以及所有已装入驱动器上的可用空间等相关信息。如果指定了目录，则只显示有关该目录所在的驱动器的信息。

-h 以用户可读的格式显示占用的块数（以 GB、MB 或 KB 为单位）。

-T 文件系统的类型（ext2、nfs，等等）。

- du [option(s)] [path]

执行此命令时若不带任何参数，则可以显示当前目录中的文件和子目录所占用的磁盘空间总量。

-a 显示各个文件的大小。

-h 以用户可读的格式输出。

-s 仅显示计算的总大小。

- free [option(s)]

free 命令用于显示有关占用 RAM 和交换空间的信息，可指明这两个类别中的空间总量和占用量。

-b 以字节为单位输出。

-k 以 KB 为单位输出。

-m 以 MB 为单位输出。

- date [option(s)]

这个简单程序可以显示当前系统时间。如果以管理员身份运行，该程序也可用于更改系统时间。

2、进程

- top [option(s)]

top 提供有关当前运行的进程的快速概览。按 H 可以进入一个页面，其中简要说明了用于定义该程序的主要选项。

- ps [option(s)] [process ID]

如果运行时未指定任何选项，此命令将显示一个表，其中包含您已经启动的所有程序或进程。

aux 显示所有进程的详细列表，不区分拥有者。

- kill [option(s)] process ID

有时程序并不能正常终止。多数情况下，通过在执行 kill 命令时指定相应的进程 ID 就应能够停止此类异常程序。kill 将发送 TERM 信号，指示程序自行关闭。如果仍无效，可使用以下参数：

-9 发送一个 KILL 信号而不是 TERM 信号，这将在几乎所有情况下终止指定的进程。

- killall [option(s)] processname

此命令类似 kill，但它使用进程名（而不是进程 ID）作为参数，可以取消具有该名称的所有进程。

3、网络

- ping [option(s)] hostname|IP address

ping 命令是用于测试 TCP/IP 网络基本功能的标准工具。它可以向目标主机发送一个小的数据包，请求立即回复。如果发送有效，ping 将据此显示一条消息，指明网络链接基本有效。

-c number 确定要发送的数据包总数，并且在发送数据包后终止（默认情况下未设置任何限制）。

-f 溢流 ping：发送尽可能多的数据包；这是为管理员保留的用于测试网络的常用方法。

-i value 指定发送两个数据包之间的时间间隔（默认值：1 秒）。

- nslookup

域名系统将域名解析为 IP 地址。使用此工具可以将查询发送到信息服务器（DNS 服务器）。

- telnet [option(s)] hostname or IP address [Port]

Telnet 实际上是一种 Internet 协议，能支持您跨网络在远程主机上操作。Telnet 同时也是一个 Linux 程序的名称， 该程序基于 telnet 协议允许用户登录到远程计算机并进行操作。

4、其它

- passwd [option(s)] [username]

用户可以使用此命令随时更改自己的口令。管理员可以使用该命令更改系统中任意用户的口令。

- su [option(s)] [username]

使用 su 命令可在当前正在运行的会话中以其它用户名登录。指定用户名及相应口令，以便使用该用户的环境。root 用户执行 su 命令时无需提供口令，因为 root 用户有权切换到任意用户。在未指定用户名的情况下使用该命令时，系统将提示您输入 root 用户口令并切换到 root 用户。

- Clear

此命令用于清空控制台中的可见区域。该命令不带选项。

第 3 章 分区与文件系统

3.1 磁盘分区

凝思安全操作系统同时支持命令行和图形界面的磁盘分区方式。

3.1.1 命令行分区工具——parted

parted 是一个用于对磁盘分区及其文件系统进行建立、修改、调整、检查、复制等操作的工具本同时支持交互模式和非交互模式。与 fdisk 比较，它支持给容量大于 2TB 的磁盘进行分区，功能更加丰富。

本节主要介绍分区表的建立，分区的查看、添加、删除等常见操作，将以非交互式命令模式举例说明。更多详细功能和用法请参考 parted 相关资料。

3.1.1.1 parted 基本用法

命令格式：

parted [options] [device [command [options...]...]]

parted [选项]... [设备 [命令 [参数]...]...]

将带有“参数”的命令应用于“设备”。如果没有给出“命令”，则以交互模式运行。

3.1.1.2 创建分区表

磁盘分区前，需要在磁盘上建立分区表，如果磁盘上已经创建好了分区表并且不需要改变分区表类型可以跳过该步骤。

parted 支持的分区表类型有：bsd、dvh、gpt、loop、mac、msdos、pc98、sun。

> **警告**
>
> 建立分区表，磁盘上原来的数据将被清空，磁盘上有正在使用的分区时将不能进行该操作，如已挂载分区、已激活的交换空间。

命令格式：

parted [device] mklabel [label-type]

示例：给一块磁盘 sda 创建一个 gpt 分区表，可以执行以下命令：

parted /dev/sda mklabel gpt

3.1.1.3 查看磁盘分区情况

命令格式：

```
parted [device] print free
```

示例：查看磁盘 sda 分区情况，可以执行以下命令：

```
parted /dev/sda print free
```

3.1.1.4 添加分区

命令格式：

```
parted [device] mkpart [part-type] [start] [end]
```

> 提示
> start 、 end 既可以是容量也可以是百分比。

示例：在一块容量为 20GB 刚建立 msdos 分区表的磁盘 sda 上建立一个 10GB 的主分区，可以执行以下命令：

```
parted /dev/sda mkpart primary 0% 10GB
```

或者：

```
parted /dev/sda mkpart primary 0% 50%
```

3.1.1.5 删除分区

命令格式：

```
parted [device] rm [partition-number]
```

示例：删除磁盘 sda 的第一个分区 sda1，可以执行以下命令：

```
parted /dev/sda rm 1
```

```
parted [device] mkpart [part-type] [start] [end]
```

3.2 文件系统

文件系统是操作系统用于明确磁盘或分区上的文件的方法和数据结构，即在磁盘上组织文件的方法。人们熟知，磁盘分区在存储数据之前需要对其进行格式化，其实这就是文件系统建立的过程。

凝思安全操作系统支持常见的文件系统，如 ext 系列、vfat、bfs、xfs 等，支持常见文件系统的命令行操作。

本节主要介绍如何在分区上创建文件系统和修复文件系统，更多详细的文件系统相关操作，请参考文件系统相关资料。

3.2.1 创建文件系统

> **警告**
>
> 在磁盘分区上建立文件系统，磁盘分区上原来的数据将被清空，该操作需要在分区未挂载的情况下才能进行。

命令格式:

```
mkfs.[fs-type] [partition]
```

或者:

```
mkfs -t [fs-type] [partition]
```

示例：在磁盘分区 sda1 上建立 ext3 文件系统，可以执行以下命令:

```
mkfs.ext3 /dev/sda1
```

或者:

```
mkfs -t ext3 /dev/sda1
```

3.2.2 修复文件系统

文件系统损坏时可以用命令 fsck 进行修复。

> **警告**
>
> 文件系统修复操作需要在分区未挂载的情况下进行本否则容易出现不可预料的情况。

命令格式:

```
fsck.[fs-type] [partition]
```

或者:

```
fsck -t [fs-type] [partition]
```

示例：修复磁盘分区 sda1 上的 etx3 文件系统，可以执行以下命令：

```
fsck.ext3 /dev/sda1
```

或者：

```
fsck -t ext3 /dev/sda1
```

3.3 磁盘阵列 RAID

独立磁盘冗余阵列（Redundant Arrays of Inexpensive Disks,RAID），旧称廉价磁盘冗余阵列，简称硬盘阵列。其基本思想就是把多个相对便宜的硬盘组合起来，成为一个硬盘阵列组，使性能达到甚至超过一个价格昂贵、容量巨大的硬盘。根据选择的版本不同，RAID 比单颗硬盘有以下一个或多个方面的好处：增强数据集成度，增强容错功能，增加处理量或容量。另外，磁盘阵列对于计算机来说，看起来就像一个单独的硬盘或逻辑存储单元。分为 RAID-0，RAID-1，RAID-1E，RAID-5，RAID-6，RAID-7，RAID-10，RAID-50，RAID-60。

简单来说，RAID 把多个硬盘组合成为一个逻辑扇区，因此，操作系统只会把它当作一个硬盘。RAID 常被用在服务器计算机上，并且常使用完全相同的硬盘作为组合。

RAID 可分为硬 RAID 和软 RAID。硬 RAID 需要依赖特定硬件实现，如磁盘阵列柜、磁盘 阵列卡。软 RAID 是通过仿真软件来实现的，如 mdadm。

本节主要简单介绍在凝思安全操作系统上如何利用 mdadm 命令来创建和关闭 RAID。

3.3.1 创建 RAID

命令格式:

```
mdadm –detail /dev/mdX
mdadm --create --auto=yes /dev/md[0-9] --raid-devices=N \
--level=[RAID-LEVEL] --spare-devices=N /dev/sdx /dev/hdx...
```

选项与参数:

- -create：创建 RAID 的选项
- -auto=yes：决定创建后面接的软件磁盘阵列装置，即/dev/md0、/dev/md1 ...
- -raid-devices=N：使用几个磁盘（或分区）作为磁盘阵列的装置
- -spare-devices=N：使用几个磁盘（或分区）作为备用(spare)装置
- -level=[RAID-LEVEL]：配置这组磁盘阵列的等级，常用的有 0、1、5 等
- -detail：查看磁盘阵列装置的详细情况

示例：利用 sdb{1,2,3,4,5}五个分区来做 RAID5（实际做 RAID 时建议使用整个磁盘而不是分区），可以执行以下命令:

```
mdadm --create --auto=yes /dev/md0 --level=5 \
--raid-devices=4 --spare-devices=1 /dev/sdb{1,2,3,4,5}
```

创建成功查看详细情况，可以使用命令:

```
mdadm --detail /dev/md0
```

或者

```
cat /proc/mdstat
```

然后就可以把 /dev/md0 当作普通的磁盘来使用了。

3.3.2 关闭 RAID

命令格式:

```
mdadm --stop /dev/mdX
```

提示

如果 RAID 已经格式化并挂载使用，需要先将其卸载。

示例：关闭上面创建的/dev/md0，可以执行以下命令:

```
mdadm --stop /dev/md0
```

3.4 逻辑卷管理 LVM

LVM 全面 Logical Volume Manager，翻译作逻辑卷轴管理员，利用 Linux 内核的 device-mapper 来实现存储系统的虚拟化（系统分区独立于底层硬件）。通过 LVM，可以实现存储空间的抽象化并在上面建立虚拟分区（virtual partitions），可以更简便地扩大和缩小分区，可以增删分区时无需担心某个硬盘上没有足够的连续空间。

LVM 基本组成如下：

- 物理卷 Physical volume(PV)：可以在上面建立卷组的媒介，可以是硬盘分区，也可以是硬盘本身或者回环文件（loopback file）。物理卷包括一个特殊的 header，其余部分被切割为一块块物理区域（physical extents）。
- 卷组 Volume group(VG)：将一组物理卷收集为一个管理单元。
- 逻辑卷 Logical volume(LV)：虚拟分区，由物理区域（physical extents）组成。
- 物理区域 Physical extent(PE)：硬盘可供指派给逻辑卷的最小单位（通常为 4MB）。

本节主要简单介绍在凝思安全操作系统上如何进行逻辑卷的创建、扩容和删除操作。

3.4.1 基本命令介绍

PV 相关：

- pvcreate：将实体 partition 创建成为 PV ；
- pvscan：搜寻目前系统里面任何具有 PV 的磁盘；
- pvdisplay：显示出目前系统上面的 PV 状态；
- pvremove：将 PV 属性移除，让该 partition 不具有 PV 属性；

VG 相关：

- vgvreate：创建 VG；
- vgscan：搜寻系统上面是否有 VG 存在。
- vgdisplay：显示目前系统上面的 VG 状态；
- vgextend：在 VG 内添加额外的 PV；
- vgreduce：在 VG 内移除 PV；
- vgchange：配置 VG 是否启动(active)；
- vgremove：删除一个 VG；

LV 相关：

- lvcreate：创建 LV；
- lvscan：查询系统上面的 LV；
- lvdisplay：显示系统上面的 LV 状态；
- lvextend：在 LV 里面添加容量；
- lvreduce：在 LV 里面减少容量；
- lvremove：删除一个 LV；
- lvresize：对 LV 进行容量大小的调整；

3.4.2 逻辑卷创建

基本步骤：

1、创建物理卷（PV）
2、创建卷组（VG）
3、创建逻辑卷（LV）

提示

卷组（VG）必须在至少一个物理卷（PV）上创建。

示例：假设创建一个名为 vg00 包含/dev/sdb1,2,3,4 的 PE 为 4M 的 VG 并在上面建立一个 2G 的逻辑卷 LV，可以执行以下命令：

```
pvcreate /dev/sdb{1,2,3，4}

vgcreate —s 4M vg00
/dev/sdb{1,2,3,4} lvcreate -L 2G vg00
-n lv00
```

逻辑卷创建完后便可以格式化挂载使用了。

3.4.3 逻辑卷扩容

示例：假设要给上面已经创建并正挂载使用的逻辑卷 lv00 扩容 1G 使其容量变为 3G，可以执行以下命令：

```
lvresize -L +1G /dev/vg00/lv00
```

如果 vg00（VG）空闲空间不够了，得先扩充 vg00，可以执行以下命令：

```
pvcreate /dev/sdb5
vgextended vg00
/dev/sdb5
```

3.4.4 删除逻辑卷

示例：假设要删除上面建立的逻辑卷 lv00，可以执行以下命令：

```
lvremove /dev/vg00/lv00
```

如果正在使用需要先卸载，可以执行以下命令：

```
umount /dev/vg00/lv00
```

如果彻底清除上面逻辑卷相关，可以执行以下命令：

```
vgremove vg00
pvremove sdb{1,2,3,4,5}
```

第 4 章 中文支持

4.1 多语言环境支持

凝思安全操作系统 V6.0.100 支持良好的多语言环境，程序运行时可以方便的无缝切换其输出的语言，以方便不同情况的需要。系统默认支持中文、英文两种语言环境，他们的切换可以通过环境变量 LANG 来指定。

示例：默认条件下系统的命令以中文方式返回响应结果：

```
jnwang@jnwang-workstation:~$ ls happy
ls: 无法访问happy: 没有那个文件或目录
jnwang@jnwang-workstation:~$
```

图 4. 1 中文响应信息

如果需要查看英文的提示信息可以使用如下命令：

```
jnwang@jnwang-workstation:~$ LANG=C ls happy
ls: cannot access happy: No such file or directory
jnwang@jnwang-workstation:~$
```

图 4. 2 英文响应信息

4.2 字符编码与中文字体

凝思安全操作系统 V6.0.100 提供国标 2 级字库及通用的中文字体，提供对简体中文的显示、输入和打印。

凝思安全操作系统支持的字符编码有:

- C
- C.UTF-8
- POSIX
- zh_CN.gb18030
- zh_CN.utf8

可以通过使用命令 locale -a /usr/share/i18n/SUPPORTED 查看。

凝思安全操作系统支持的中文字体有:

- -cc-song-medium-r-normal-jiantizi-0-0-75-75-c-0-gb2312.1980-0
- -cc-song-medium-r-normal-jiantizi-40-400-75-75-c-400-gb2312.1980-0
- -cc-song-medium-r-normal-jiantizi-48-480-75-75-c-480-gb2312.1980-0
- -guobiao-song-medium-r-normal--0-0-72-72-c-0-gb2312.80&gb8565.88-0
- -guobiao-song-medium-r-normal--16-160-72-72-c-160-gb2312.80&gb8565.88-0
- -guobiao-song-medium-r-normal--16-160-72-72-c-160-gb2312.80-0
- -guobiao-song-medium-r-normal--16-160-72-72-c-160-gb8565.88-0
- -isas-fangsong ti-medium-r-normal--0-0-72-72-c-0-gb2312.1980-0
- -isas-fangsong ti-medium-r-normal--16-160-72-72-c-160-gb2312.1980-0
- -isas-song ti-medium-r-normal--0-0-72-72-c-0-gb2312.1980-0
- -isas-song ti-medium-r-normal--16-160-72-72-c-160-gb2312.1980-0
- -isas-song ti-medium-r-normal--24-240-72-72-c-240-gb2312.1980-0
- -wenquanyi-wenquanyi bitmap song-bold-r-normal--0-0-100-100-p-0-gb18030.2000-0
- -wenquanyi-wenquanyi bitmap song-bold-r-normal--0-0-100-100-p-0-gb2312.1980-0
- -wenquanyi-wenquanyi bitmap song-bold-r-normal--0-0-100-100-p-0-gbk-0
- -wenquanyi-wenquanyi bitmap song-bold-r-normal--0-0-75-75-p-0-gb18030.2000-0
- -wenquanyi-wenquanyi bitmap song-bold-r-normal--0-0-75-75-p-0-gb2312.1980-0
- -wenquanyi-wenquanyi bitmap song-bold-r-normal--0-0-75-75-p-0-gbk-0
- -wenquanyi-wenquanyi bitmap song-medium-r-normal--0-0-100-100-p-0-gb18030.2000-0
- -wenquanyi-wenquanyi bitmap song-medium-r-normal--0-0-100-100-p-0-gb2312.1980-0
- -wenquanyi-wenquanyi bitmap song-medium-r-normal--0-0-100-100-p-0-gbk-0
- -wenquanyi-wenquanyi bitmap song-medium-r-normal--0-0-75-75-p-0-gb18030.2000-0
- -wenquanyi-wenquanyi bitmap song-medium-r-normal--0-0-75-75-p-0-gb2312.1980-0
- -wenquanyi-wenquanyi bitmap song-medium-r-normal--0-0-75-75-p-0-gbk-0
- hanzigb16fs
- hanzigb16st
- hanzigb24st

可以通过使用命令 xlsfonts |grep gb 查看。

4.3 输入法

凝思安全操作系统 V6.0.100 默认情况下英文输入，如果需要输入中文可以按组合键 Ctrl+Space 来切换到中文输入。

默认情况下中文输入使用的是 Google 拼音，如图 4.3 所示。

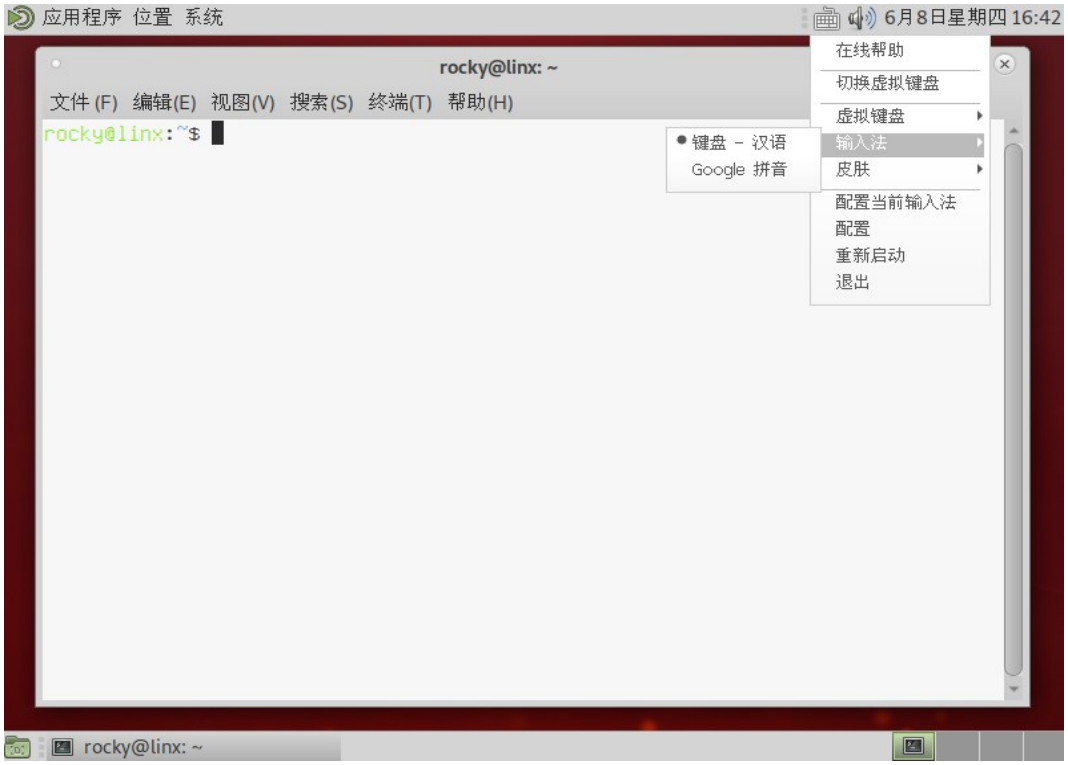


图 4. 3 输入法

第 5 章 系统管理

5.1 关闭或重新启动系统

5.1.1 关闭系统

您可以在桌面环境下从主菜单中选择[关机]，或者在终端中使用下列命令关闭凝思安全操作系统：

```
$ shutdown -h now
```

在系统使用过程中非正常关闭凝思安全操作系统可能造成系统损坏或数据丢失，因此，应该尽量避免这种情况的发生。

5.1.2 重启系统

您可以在桌面环境下从主菜单中选择[重启]，或者在终端中使用下列命令重新启动凝思安全操作系统：

```
$ shutdown -r now
```

5.2 初始化设置

5.2.1 设备驱动模块配置

- 路径

  /etc/modules

- 作用

  设置系统启动时的加载设备模块

- 参数设置

  alias 设备号 模块

- 例样

```
alias eth0 e100
```

5.2.2 profile 文件

- 路径

  /etc/profile（对系统所有用户登录有效）

  ~/.profile（对相应用户登录有效）

- 作用

  设置系统所有用户或者单用户登录时运行的环境变量

- 例样

```
if [ "‘id -u‘" -eq 0 ]; then PATH="/usr/local/sbin:/usr/local/
  bin:/usr/sbin:/usr/bin:/sbin:/bin"
else
  PATH="/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games"
fi
export PATH


if [ "$PS1" ]; then
  if [ "$BASH" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      .
    /etc/bash.bashrc fi
  else
    if [ "‘id -u‘" -eq 0 ];
      then PS1='# '
    else
      PS1='$ '
    fi
  fi
fi


if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh;
do
      if [ -r $i ]; then
          . $i
      fi
```

```
  done
  unset i
fi


alias ls='ls -l'
color=auto' alias l='ls -
l'
alias la='ls a'
umask 022
```

5.2.3 issue 文件

- 路径
  /etc/issue
- 作用
  系统登录时的欢迎界面
- 例样

```
LinxOS 6.0.100 20240111 \n \l
```

5.3 系统配置

5.3.1 fstab 文件

- 路径

  /etc/fstab

- 作用

  静态文件系统信息，包含系统磁盘分区以及存储设备如何挂载，以及挂载点等信息。

- 参数设置

  <file system> <mount point> <type> <options> <dump> <pass>

  - <file system>

    要挂载的设备文件，包含设备文件所在路径及全名。

  - <mount point>

    设备的挂载点，包含挂载点的路径。

  - <type>

    设备的分区格式，如 ext2、ext3、xfs、jfs、iso9660、fat、swap 等。

  - <options>

    挂载设备时所要设定的状态，如 ro（只读）、defaults（包括了其它参数如 rw、suid、exec、auto、nouser、async）等，默认为 defaults。

  - <dump>

    在系统 DUMP 时是否需要 BACKIP 的标志位，默认值是 0。

  - <pass>

    设定此<file system>是否要在开机时做 check 的动作，除了 / 的<file system>其必要的 check 为 1 之外，其它皆可视需要设定，默认值是 0。

- 例样

```
# <file system>  <mount point>          <type>  <options> <dump>  <pass>
proc             /proc                  proc    defaults   0       0
sysfs            /sys                   sysfs   defaults   0       0
devpts           /dev/pts               devpts  defaults   0       0
tmpfs            /dev/shm               tmpfs   defaults   0       0
/dev/sda1        swap                   swap    defaults   0       0
/dev/sda2        /                      ext3    defaults   1       1
```

5.4 挂载和卸载

只有 root 或授权管理员才能运行 mount 和 umount 命令来挂载和卸载。要使其他用户也能运行这些命令，需编辑 /etc/fstab 文件，为相应的设备指定选项 user，即允许一般用户挂载该设备。

5.4.1 挂载

mount [option(s)] [device] mountpoint

使用此命令可以将任意数据介质（如硬盘、CD_ROM 驱动器和其它设备）装入 Linux 文件系统的某个目录。

-r　只读装入。

-t filesystem 指定文件系统，通常包括：ext2（表示 Linux 硬盘）、msdos（表示 MS-DOS 介质）、vfat（表示 Windows 文件系统）、iso9660（表示 CD）。

对于没有在 /etc/fstab 中定义的硬盘，还须同时指定设备类型。在这种情况下只能由系统管理员装入。如果其他用户也应该能够装入文件系统，则应在 /etc/fstab 文件的对应行中输入选项 user（用逗号分隔多个用户），并保存所做更改。

5.4.2 卸载

umount [option(s)] mountpoint

此命令可用于从文件系统中卸载装入的驱动器。为防止数据丢失，请在将可移除的数据介质从其所在驱动器中移除之前运行此命令。

5.5 系统监控

凝思安全操作系统 V6.0.100 默认自带系统监控工具 mate-system-monitor。

mate-system-monitor 可以从“开始——搜索——MATE 系统监视器”启动。

图 5. 1 系统监控工具

MATE 系统监视器界面，如图 5.2 所示：

图 5. 2 MATE 系统监视器

5.6 数据备份与恢复

5.6.1 数据备份的重要性

凝思安全操作系统是稳定而健壮的。但是任何操作系统都有可能遭遇到一些无法考虑到的特殊情况，比如认为误操作，电气故障，以及自然界不可抗等因素，最终导致系统中最珍贵的数据丢失。所以数据备份和恢复就显得非常重要。

5.6.2 工具名称

凝思安全操作系统上的备份恢复工具可使用 dump 和 restore。

5.6.3 工具安装

光驱中插入凝思安全操作系统安装光盘，执行如下命令进行工具的安装:

```
apt-get install dump
```

5.6.4 备份与恢复策略

5.6.4.1 完全备份

完全备份是完整地备份数据。

完全备份的优点是当数据发生丢失时，可以恢复所有丢失的数据；缺点是重复地进行数据完整备份会占用大量的空间，而且备份时间也会较长。

完全备份的示意图如 5.3 所示。

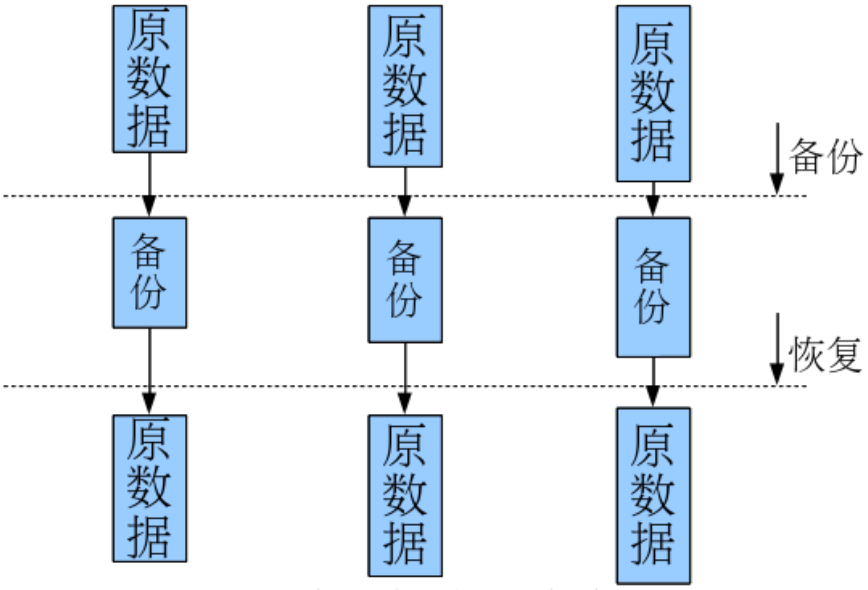


图 5. 3 完全备份示意图

5.6.4.2 增量备份

增量备份只能用于挂载了分区的目录。

增量备份是在依次完全备份或上一次增量备份后，以后每次的备份只需备份与前一次相比增加或者被修改的文件。这意味着，第一次增量备份的对象是进行完全备份后产生的增加

和修改的文件，后续的增量备份则是对前一次增量备份所产生的增加和修改的文件，增量备份级别为 1-9。

增量备份的优点是没有重复的备份数据，因此备份的数据量不大，备份所需的时间很短；缺点是增量备份的数据恢复比较麻烦，必须具有需要恢复的增量备份前面依赖的完全备份和增量备份（一旦丢失或者损坏其中任何一份，就会造成恢复失败），并且恢复是沿着完全备份到依次产生的增量备份按时间顺序进行恢复的，极大地延长了恢复时间。

增量备份的示意图如 5.4 所示。

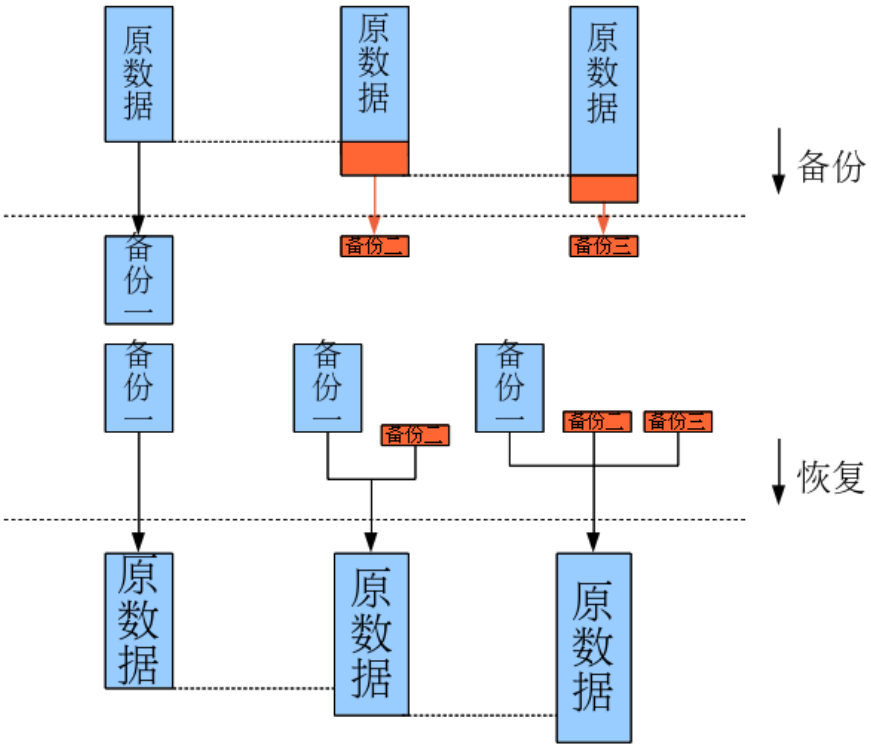


图 5. 4 增量备份示意图

5.6.4.3 差异备份

差异备份只能用于挂载了分区的目录。 差异备份是以完全备份为基准的一种备份方式。例如，第一次对原数据进行完全备份，第二次的备份为原数据从第一次完全备份到第二次备份间的差异，原数据从第三次备份为第 一次完全备份到第三次备份间的差异，依次类推。

差异备份的优点是避免了以上两种备份策略的缺陷，同时由具备有两者的优点，无需每次都对系统做完全备份，因此备份数据量小，备份时间短，其次在恢复数据时也很方面，只需要两个备份，即第一次的完全备份和数据发生灾难前一次的差异备份即可。

差异备份的示意图如 5.5 所示。

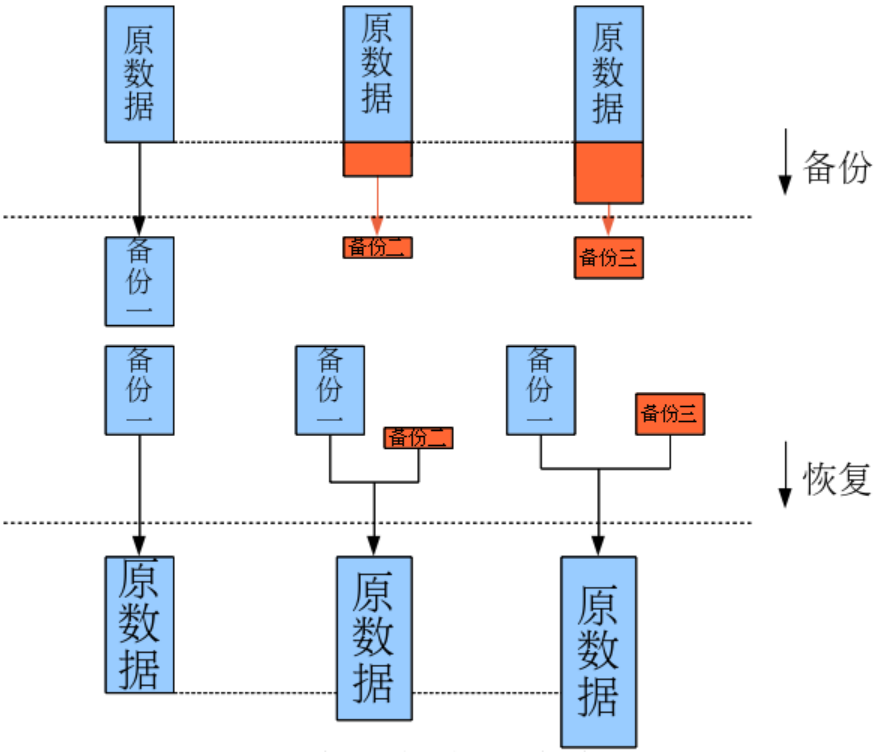


图 5. 5 差异备份示意图

5.6.5 工具参数

表 5. 1 dump

命令格式	Dump[-选项中的备份级别][-选项][-f 备份文件]要备份的数据
命令格式	**选项解析**
-[0-9]	备份级别，普通备份使用 0，分区备份中使用不同的数字进行增量或,差异备份。
-S	后跟需要备份的文件，可以计算出备份需要的空间。
-u	将本次备份时间记入档案，本选项不支持非分区挂载点。
-v	显示详细执行信息。
-j	加入 bzip2 支持，将内容进行压缩。
-f	指定备份文件，也可以接设备文件如/dev/st0 等。
-W	列出在/etc/fstab 里具有 dump 设定的分区是否有进行过备份。

表 5. 2 restore

命令格式	命令解析
restore -t [-f 备份文件]	查看备份文件信息

restore -C [-f 备份文件][-D 挂载点]	比较备份数据和挂载点数据的差别
restore -i [-f 备份文件]	进入互动模式
restore -r [-f 备份文件]	还原系统分区

5.6.6 功能演示

1、以不同策略对分区进行备份

/dev/sda6 的挂载目录为 /tmp/mountpoint，要对该分区备份 3 次。

1）采取完全备份

```
dump -0u -f /tmp/lv00.dmp /tmp/mountpoint
```

第一次完全备份，备份文件为/tmp/lv00.dmp。

```
dump -0u -f /tmp/lv02.dmp /tmp/mountpoint
```

第二次完全备份，备份文件为/tmp/lv02.dmp。

```
dump -0u -f /tmp/lv03.dmp /tmp/mountpoint
```

第三次完全备份，备份文件为/tmp/lv03.dmp。

2）采取增量备份

```
dump -0u -f /tmp/lva0.dmp /tmp/mountpoint
```

第一次为完全备份，备份文件为/tmp/lva0.dmp。

```
dump -0u -f /tmp/lva1.dmp /tmp/mountpoint
```

基于第一次备份的增量备份，备份文件为/tmp/lva1.dmp。

```
dump -0u -f /tmp/lva2.dmp /tmp/mountpoint
```

基于前两次备份的增量备份，备份文件为/tmp/lva2.dmp。

3）采取差异备份

```
dump -0u -f /tmp/lvd00.dmp /tmp/mountpoint
```

第一次为完全备份，备份文件为/tmp/lvd00.dmp。

```
dump -0u -f /tmp/lvd01.dmp /tmp/mountpoint
```

基于第一次备份的差异备份，备份文件为/tmp/lvd01.dmp。

```
dump -0u -f /tmp/lvd02.dmp /tmp/mountpoint
```

基于第二次备份的差异备份，备份文件为 /tmp/lvd02.dmp。

2、普通文件备份

```
dump -0 -f /tmp/bakfile /tmp/nmountpoint
```

把/tmp/nmountpoint 完全备份到/tmp/bakfile。

3、数据恢复

查看备份内容:

```
restore -tf /tmp/bakfile        #查看备份文件 bakfile 中备份的文件
```

通过备份文件交互式地完成数据恢复:

```
restore -if /tmp/bakfile        #进入交互模式，开始恢复 bakfile 备份的内容
restore > help                  #在交互式命令行中输入 help，查看恢复命令
Restore > cd /tmp               #切换目录
Restore > add /nmountpoint      #添加 nmountpoint 到恢复队列
Restore > extract               #开始恢复
Restore > quit                  #退出交互
```

第 6 章 软件包管理

6.1 软件包管理机制

6.1.1 软件包概述

软件包通常包含了实现一系列相关命令或特性所必须的所有的文件。有两种类型的 Deb 软件包:

- Binary packages(二进制软件包),包含可执行文件、配置文件、man/info 页面、版权声明和其它文档。它们通常使用.deb 的扩展名以示区别。这种二进制软件包可使用工具 dpkg 解包。
- Source packages(源码包),包含一个.dsc 文件它用于描述源码包(包括下列文件的名称),一个.orig.tar.gz 文件它是未经修改的原始源代码压缩文件,以及一个.diff.gz 文件它包含了该软件包 Debian 化时所做的修改。

6.1.2 软件包命名约定

软件包命名遵循下列约定:

foo_ver-rev_arch.deb

一般这里的 foo 是软件包的名称,ver 是软件本身的版本号,rev 是修订版本号,arch 是目标架构名称。修订版本号由开发者或创建这个软件包的人指定。通常,包被修改过之后,会把修改版本号加一。

6.1.3 维护脚本

维护脚本是一种可执行脚本,它在软件包安装之前或之后自动运行。它和一个名叫 control 的文件一起组成 Deb 包文件的“管理”部分。这些文件包括:

- preinst

  在软件包(.deb)文件解包之前,运行这个脚本。许多“preins”脚本的任务是停止作用于待升级软件包的服务,直到软件包安装或升级完成。
- postinst

  该脚本的任务是完成软件包(.deb) 文件解包文件的配置工作。通常,“postinst”脚本等待用户输入,或提醒用户,如果他接受当前默认值,要记得软件包安装完后返回重新配置。许多“postinst”脚本负责执行有关命令为新安装或升级的软件重启服务。
- prerm

  该脚本负责停止与软件包关联的 daemon 服务。它在删除软件包关联文件之前执行。
- postrm

  该脚本负责修改软件包链接或文件关联,或删除由它创建的文件。

6.1.4 软件包优先级

每个软件包均被发布者指定了一个优先级,作为软件包管理系统的一个辅助参数,优先级的值有:

- Required(必须)
  该级别软件包是保证系统正常运行必须的。
- Important(重要)
  在任何类 Unix 系统上均安装有该级别软件包。
- Standard(基本)
  该级别软件包是任何 Linux 系统的标准件,它们组成一个小而精的字符模式的系统。
- Optional(推荐)
  该级别软件包包括那些你可能想安装的软件,即使对它们并不熟悉 本 但对它们没有特殊的要求。
- Extra(额外)
  该级别软件包可能与其它高级别软件包冲突,仅当你知道其用途时才会使用它们,或者有运行它们有专门要求。

6.1.5 软件包依赖关系

软件包管理系统依赖声明,它描述了这一事实: 一些软件包需要其它软件包被安装才能正常运行或运行得更好。

- 软件包 A 依赖(depends)软件包 B: 要运行 A 必须安装 B。在有些情况下,A 不仅依赖 B,还要求 B 的特定版本。版本依赖通常有最低版本限制,A 更依赖于 B 的最新版本而非某个特定版本。
- 软件包 A 推荐(recommends)软件包 B: 软件包维护者认为所有用户都不会喜欢缺少某些功能的 A,而这些功能需要 B 来提供。
- 软件包 A 建议(suggests)软件包 B: B 中某些文件与 A 的功能相关,并能够增强 A 的功能。这种关系通过声明软件包 B 增强软件包 A 来表示。
- 软件包 A 与软件包 B 冲突(conflicts):如果系统中安装了 B 那么 A 无法运行。“conflicts”常和“replaces”同时出现。
- 软件包 A 替换(replaces)软件包 B: B 安装的文件被 A 中的文件移除和覆盖了。
- 软件包 A 提供(provides)软件包 B: A 包含了 B 中的所有文件和功能。

6.2 软件包管理工具

6.2.1 常用的包管理工具

- dpkg 底层软件包管理工具
- apt 高级软件包管理工具集
- aptitude apt 命令行前端

6.2.2 dpkg

dpkg 是软件包管理器的基础,被用于安装、卸载和查看.deb 软件包相关的信息。

常用的 dpkg 命令参数如下:

- dpkg -i <package-file> 安装软件包
- dpkg -r [--purge] <package> 卸载软件包
- dpkg -l <package-pattern> 列出软件包名称和安装状态
- dpkg -s <package> 查看软件包详细状态
- dpkg -L <package> 软件包安装文件列表
- dpkg -c <package-file> 列出软件包文件.deb 内容
- dpkg -x <package-file> <dir> 解压软件包文件到目录

6.2.3 apt

apt 是一个软件包管理工具集合,是 dpkg 的高级前端,可以解决软件包依赖问题。apt 工具集使用时通常需要通过网络访问软件源,最常用的工具包括 apt-get 和 apt-cache。

6.2.3.1 设置软件源

软件源指的是提供各种软件包下载的服务器。apt 从指定的软件源获取软件包列表(/var/lib/apt/lists/*),在需要时下载安装。

您可以修改 /etc/apt/sources.list 来设置软件源,例如:

```
deb http://ftp.debian.org/debian/squeeze main non-free contrib
```

6.2.3.2 apt-get

apt-get 是一个命令行方式的软件包处理工具,常用的命令参数如下:

- apt-get update 更新软件包列表
- apt-get install <package> 安装软件包
- apt-get install <package> --reinstall 重新安装软件包
- apt-get remove <package> 卸载软件包
- apt-get remove <package> --purge 卸载软件包(删除配置文件)
- apt-get upgrade 升级所有软件包
- apt-get build-dep <package> 安装软件包编译环境
- apt-get source <package> 下载源码包

- apt-get clean 清理软件包缓存
- apt-get check 检查是否有损坏的依赖

6.2.3.3 apt-cache

apt-cache 用于查询软件包的相关信息,常用的命令参数如下:

- apt-cache search <package> 搜索软件包
- apt-cache show <package> 显示软件包相关信息
- apt-cache showsrc <package> 显示源码包相关信息
- apt-cache depends <package> 显示软件包的依赖信息
- apt-cache rdepends <package> 显示依赖<package>的软件包的信息

6.2.4 aptitude

6.2.4.1 介绍

aptitude 与 apt-get 一样,是功能强大的包管理工具。与 apt-get 不同的是,aptitude 在处理依赖问题上更佳一些。举例来说, aptitude 在删除一个包时,会同时删除本身所依赖的包。这样,系统中不会残留无用的包,整个系统更为干净。

常用 aptitude 命令如下:

- aptitude update 更新可用的软件包列表
- aptitude upgrade 升级可用的软件包
- aptitude dist-upgrade 将系统升级到新的发行版
- aptitude install <package> 安装软件包
- aptitude remove <package> 删除软件包
- aptitude purge <package> 删除软件包及其配置文件
- aptitude search <string> 搜索名称中包含该字符串的软件包
- aptitude show <package> 显示软件包的详细信息
- aptitude clean 删除下载的软件包文件
- aptitude autoclean 仅删除过期的软件包文件

除此之外,您还可以执行命令 aptitude 在文本界面模式中使用 aptitude,如图 6.1 所示。

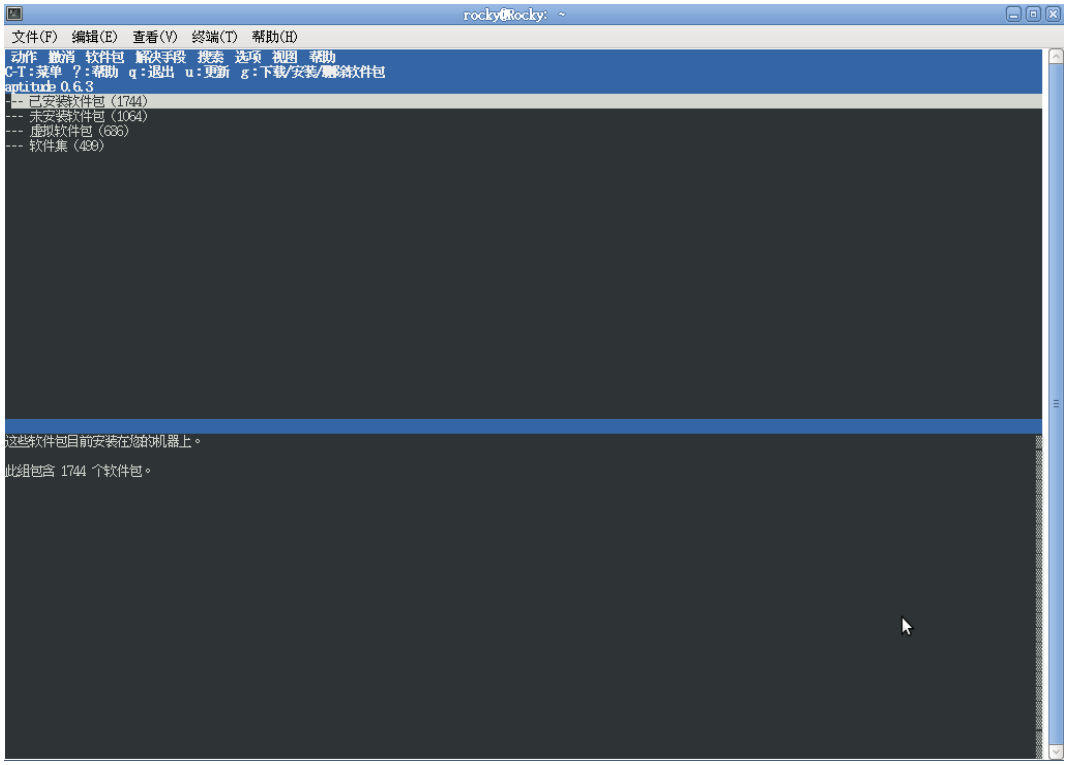


图 6. 1 aptitude 文本界面模式

6.2.4.2 aptitude 安装

1、将 iso 镜像启动盘插入服务器，将启动盘挂载到/mnt 目录下，执行以下命令：

sudo mount /dev/sdb /mnt

2、添加源至/etc/apt/sources.list，执行以下命令：

sudo apt-cdrom --cdrom /mnt add

3、更新源列表，执行以下命令：

sudo apt-get update

4、使用 apt-get 安装，执行以下命令：

sudo apt-get install aptitude

6.2.5 安装软件包

请先确认系统中有 dpkg、apt、aptitude 等包管理工具,并且已将软件源写入到/etc/apt/sources.list 文件中。

然后执行以下命令:

apt-get update

或:

aptitude update

更新源信息将写到 /var/lib/apt/lists/目录下。至少包括以 Packages 、Release、Release.gpg 为结尾的三个文件。

再执行以下命令:

apt-get install <package> 或 aptitude install <package>

安装或更新指定软件包

或者:

apt-get upgrade 或 aptitude upgrade

更新所有已安装的软件包。

更新软件包将被下载到 /var/cache/apt/archives/ 中暂存,然后再进行安装。

升级的记录会写在日志文件 /var/log/dpkg.log 、 /var/log/apt/term.log 中。

第 7 章 用户管理

7.1 添加用户

以 root 或授权管理员登录，执行 useradd testuser1，将添加用户 testuser1，如果需要同时建立用户目录，则执行 useradd -m testuser1。

重要

添加用户时会随机生成用户口令，请记录该口令或修改口令。

提示

执行 less /etc/passwd 可以看到新增加用户 testuser1 的信息。

7.2 删除用户

以 root 或授权管理员登录，执行 rmuser testuser1，testuser1 的用户信息将会被从各个相关的用户数据文件（如/etc/passwd、/etc/group、/etc/shadow 等）中删除，如果需要删除用户目录及其它相关文件，则执行 rmuser -r testuser1。

7.3 添加用户组

以 root 或授权管理员登录，执行 groupadd test，将添加用户组 test。

提示

执行 useradd -g test testuser1 可创建用户 testuser1 并添加到 test 用户组，但 test 组必须已存在。

7.4 删除用户组

以 root 或授权管理员登录，执行 groupdel test，将删除用户组 test。

警告

只有某用户组是空组（没有用户属于该组），或该用户组不是任何用户的 primary group 的情况下，才可以删除该用户组，其它情况下均不能删除该用户组。

7.5 将用户添加到用户组

以 root 或授权管理员登录，执行 gpasswd -a testuser1 test，将用户 testuser1 添加到用户组 test。

7.6 将用户从用户组中删除

以 root 或授权管理员登录，执行 gpasswd -d testuser1 test，将用户 testuser1 从用户组 test 中删除。

> 注意
> 用户必须属于至少一个用户组，因此，当用户只属于一个用户组时，将不能使用上述命令将用户从该用户组中删除。

7.7 改变用户当前所在组

如果用户同时属于几个用户组，可使用 newgrp 命令来改变当前所在组，之后将使用新的组 ID 来进行文件存取权限的管理。

如：用户 testuser 同时属于 users 和 test 用户组，当前组别为 users，执行 touch test1.txt 新建文件 test1.txt，再执行 newgrp test 改变登录的用户组到 test，执行 touch test2.txt 新建文件 test2.txt。查看文件 test1.txt 和 test2.txt 的信息，属主都是 testuser，但分别属于 users 和 test 用户组。

7.8 修改用户口令

使用 passwd 命令可修改当前用户自己的口令。

如：用户 testuser1 使用 passwd 命令，若旧口令输入错误，则不能修改口令。按要求输入正确的旧口令后，输入新口令并再次确认，口令修改成功。

```
$ passwd
为 testuser1 更改 STRESS 密码
（当前）UNIX 密码:                    # 输入错误的旧口令
passwd: 鉴定令牌操作错误
passwd: password unchanged

$ passwd
为 testuser1 更改 STRESS 密码
（当前）UNIX 密码:                    # 输入正确的旧口令
输入新的 UNIX 密码:
重新输入新的 UNIX 密码
passwd: 已成功更新密码
```

若输入的新口令不符合要求（如：与旧口令相同），则会要求重新输入新口令。

```
密码未更改
输入新的 UNIX 密码:
```

重新输入的次数有限制（由授权管理员设置），当尝试次数达到限制次数后，系统返回

提示并退出口令修改：

```
passwd: 鉴定令牌操作错误
passwd: password unchanged
```

输入新口令需要注意以下几点：

- 口令必须为 6~127 位的英文、数字和特殊字符的组合，若输入的新口令不足 5 位，系统将提示要求重新输入：

```
输入新的 UNIX 密码:                    # 输入不足 5 位的口令
必须选择更长的密码:
输入新的 UNIX 密码:
```

- 输入新口令时，若输入的口令保密性不好，比如与之前的口令相似、是某个英文单词或过于简单，系统也会提示，并要求重新输入：

```
输入新的 UNIX 密码:                    # 输入与之前相似的口令，如原口令
                                       # 12345678，输入新口令为 1234567
Bad:new and old password are too similar
输入新的 UNIX 密码:                    # 输入某个英文单词或过于简单
                                       # 如：rocky1
Bad:new password is too simple
输入新的 UNIX 密码:
```

> **重要**
> 口令位数越少时，就要求越复杂的字符组合。

- 若确认新口令（即第二次输入新口令）时错误，系统将提示，并要求重新输入新口令：

```
Sorry, passwords do not match
passwd:无法恢复鉴定信息
passwd:password unchanged
```

> **重要**
> 用户口令的修改也可能被限制。授权管理员可以设定两次修改用户口令之间间隔的最长和最短时间、口令作废的日期、在用户口令作废前多少天警告用户、在账号自动锁定之前容许用户多少天不活动等等。

以 root 或授权管理员登录，执行 passwd 命令，可修改任意用户的口令。

小心

以 root 的身份修改用户口令为强制性修改，不需要验证旧口令，请谨慎使用。

7.9 修改用户口令时限

以 root 或授权管理员登录，使用 passwd 或 chage 命令可以设定两次修改口令之间间隔的最长和最短时间，口令作废的日期，在用户口令作废前多少天警告用户，在账户自动锁定之前容许用户多少天不活动等。

passwd [Options] username

Options：

- -a, --all
  显示所有帐户的信息。仅能与“-S”选项一起使用。
- -d, --delete
  删除用户帐户的口令。
- -e, --expire
  用户帐号到期的日期。过了这天，此用户帐号将不可用。
- -k, --keep-tokens
  只更改到期的口令。
- -i, --inactive INACTIVE
  停滞时期。如果一个口令已过期这些天，那么此用户帐号将不可用。
- -l, --lock
  锁定指定的帐号。锁定后，此用户帐号将不可用。
- -n, --mindays MIN_DAYS
  可更改口令的最小天数，为 0 时代表任何时候都可以更改口令。
- -S, --status
  显示用户帐号的口令状态信息。
- -u, --unlock
  为指定的帐号解锁。解锁后，此用户帐号将变为可用。
- -w, --warndays WARN_DAYS
  用户口令到期前，提前收到警告信息的天数。
- -x, --maxdays MAX_DAYS
  口令保持有效的最大天数。

chage [Options] username

Options：

- -d, --lastday LAST_DAY
  设置上一次更改口令的日期。
- -E, --expiredate EXPIRE_DATE
  用户帐号到期的日期。过了这天，此用户帐号将不可用。
- -I, --inactive INACTIVE
  停滞时期。如果一个口令已过期这些天，那么此用户帐号将不可用。
- -l, --list
  列出当前的设置。由非特权用户来确定他们的密码或用户帐号何时过期。

- -m, --mindays MIN_DAYS
  可更改口令的最小天数，为 0 时代表任何时候都可以更改口令。
- -M, --maxdays MAX_DAYS
  口令保持有效的最大天数。
- -W, --warndays WARN_DAYS
  用户口令到期前，提前收到警告信息的天数。

例如，将用户 testuser1 两次修改口令之间的最小天数设为 2、最大天数设为 90、作废日期为 2009 年 1 月 1 日，并且在接近作废日期的 14 天里警告用户：

```
$ chage -m 2 -M 90 -E 2009-01-01 -W 14 testuser1
```

7.10 修改用户信息

以 root 或授权管理员登录，使用 usermod 命令可修改用户信息。

```
usermod [-c comment] [-d home_dir [-m]]
[-e expire_date] [-f inactive_days] [-g initial_group]
[-G groups ...] [-l name] [-s shell] [-u uid [-o]] name
```

在命令行中使用一个 name 参数指定要改变信息的用户（指定的用户必须存在，且不能是当前在线的用户），并且利用它所提供的各个选项来指定改变字段以及值。

- -c comment
  更改用户的注释。
- -d home_dir
  更改用户主目录名称。
- -e expire_date
  更改用户的到期日，到期日使用 mm/dd/yy 格式来设定。
- -f inactive_days
  更改用户到期几天后停止使用该用户，设定为 0 表示到期后立即停用，预设为-1 表示要关闭这个功能。
- -g initial_group
  更改用户的所属组，initial_group 参数可以使用组的名称或编号来指定组，不过指定的组必须是已存在的。
- -G group,[...]
  更改该用户属于哪些组的成员，可以指定多个组，组间以逗号分隔。可以使用组的名称或编号来指定组，不过指定的组必须是已存在的。
- -l name
  更改用户的名称，该用户的主目录名称需要手动修改。
- -s shell
  更改用户登录系统后执行的 shell，如果指定空白，将会执行系统预设的 shell。
- -u uid
  更改用户的 UIS。指定的 UID 不能与其它用户的 UID 重复，如果 UID 不唯一，则必须加上-o 选项。UID 不可为负值。

7.11 修改口令文件

为了保证系统的安全性，系统通常对用户的口令进行 shadow 处理，并把用户口令保存到只有 root 可读的 /etc/shadow 文件中。该文件包含了系统中所有用户和用户口令等相关信息。每个用户在该文件中对应一行，并且用冒号分成九个域，每一个域包括以下内容:

1、用户登录名
2、用户加密后的口令（若为空，表示该用户不需口令即可登录，若为 * 号，表示该帐号被禁止）
3、从 1970 年 1 月 1 日至口令最近一次被修改的天数
4、口令在多少天内不能被用户修改
5、口令在多少天后必须被修改
6、口令过期多少天后用户帐号被禁止
7、口令在到期多少天内给用户发出警告
8、口令自 1970 年 1 月 1 日被禁止的天数
9、保留域

以 root 登录，可对口令文件 /etc/shadow 进行修改。

7.12 定义鉴别阈值

用户登录系统时，若输入的新口令不符合要求，则会被要求重新输入新口令。重新输入的次数也有限制，当尝试次数达到限制次数后，系统返回提示并退出口令修改，并在一定的时间内不能再次尝试登录系统。

管理员可以修改/etc/login.defs 文件来自定义这些阈值。

如：

```
LOGIN_RETRIES 3         # 输入口令的尝试次数限制为 3 次
LOGIN_TIMEOUT 60 # 登录超时限制为 60 秒
PASS_MIN_LEN 5          # 口令必须为 5 位以上
```

第 8 章 网络管理

8.1 网络参数配置

8.1.1 interfaces 文件

- 路径

  /etc/network/interfaces

- 作用

  设置网络接口参数。

- 例样

  下面是一个配置示例，它在一个网络接口中配置了两个静态 IP 地址：

```
# This file describes the network interfaces available on your
# system and how to activate them. For more information, see
# interfaces(5).

# The loopback interface
auto lo
iface lo inet loopback

# The primary network interfaces
auto eth0
iface eth0 inet static
        address 192.168.1.1
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1

auto eth0:0
iface eth0:0 inet static
        address 192.168.1.2
        netmask 255.255.255.0
        network 192.168.1.0
        broadcast 192.168.1.255
        gateway 192.168.1.1
```

下面是一个从 DHCP 服务器自动获取 IP 地址的实例：

```
# This file describes the network interfaces available on your
# system and how to activate them. For more information, see
# interfaces(5).

# The loopback interface
auto lo
iface lo inet loopback

# The primary network interfaces
allow-hotplug eth0
iface eth0 inet dhcp
```

8.1.2 resolv.conf 文件

- 路径

  /etc/resolv.conf

- 作用

  DNS 域名解析的配置文件。

- 参数设置

  nameserver # 定义 DNS 服务器的 IP 地址
  domain # 定义本地域名
  search # 定义域名的搜索列表
  Sortlist # 对返回的域名进行排序

  最主要是 nameserver 关键字，如果没指定 nameserver 找不到 DNS 服务器，其它关键字是可选的。

- 例样

```
nameserver 192.168.0.1
domain linx-info.com
search www.linx-info.com
```

8.1.3 hostname 文件

- 路径

  /etc/hostname

- 作用

  记录本机的主机名。

- 例样

该文件只有一行，如：

```
Rocky
```

8.1.4 services 文件

- 路径

  /etc/services

- 作用

  Internet 网络服务文件，记录网络服务名和它们对应使用的端口号及协议。文件中的每一行对应一种服务，它由 4 个字段组成，中间用 TAB 或空格分隔，分别表示“服务名称”、“使用端口”、“协议名称”以及“别名”。

- 例样

  下面是这个文件的节选内容：

  ```
  Tcpmux    1/tcp     #     TCP port service multiplexer
  echo   7/tcp
  echo   7/udp
  discard     9/tcp     sinknull
  discard     9/udp     sinknull
  systat 11/tcp    users
  daytime    13/tcp
  daytime    13/udp
  netstat      15/tcp
  qotd   17/tcp    quote
  msp    18/tcp    # message send protocol
  ```

```
msp    18/udp
chargen    19/tcp    ttytst source
chargen    19/udp    ttytst source
ftp-data    20/tcp
ftp    21/tcp
fsp    21/udp    fspd
ssh    22/tcp    # SSH Remote Login Protocol
ssh    22/udp
telnet 23/tcp
smtp   25/tcp    mail
time   37/tcp    timserver
```

提示

一般情况下，不要修改该文件的内容，因为这些设置都是 Internet 标准的设置。一旦修改，可能会造成系统冲突，使用户无法正常访问资源。

Linux 系统的端口号的范围为 0~65535，不同范围有不同的意义。

0	不使用
1~1023	系统保留，只能由 root 用户使用
1024~4999	由客户端程序自由分配
5000~65535	由服务器端程序自由分配

8.1.5 hosts.allow 文件

- 路径

  /etc/hosts.allow

- 作用

  设置系统的外部连接许可

- 参数设置

  <service>: <ip>

  注意

  hosts.allow 优先程度高于 hosts.deny

- 例样

sshd : 192.168.0.0/255.255.255.0
允许~192.168.0~网段的客户端通过~sshd~服务访问服务器

8.1.6 hosts.deny 文件

- 路径
  /etc/hosts.deny
- 作用
  设置系统的外部连接拒绝
- 参数设置
  <service>: <ip>
  特殊参数与 hosts.allow 相同

> 注意
> hosts.deny 优先程度低于 hosts.allow

- 例样

  ALL: ALL: DENY

> 注意
> 为了保持系统的安全性，请尽量不要修改本文件，如需设置其他客户端通过服务访问本机，只需在 /etc/hosts.allow 文件中设置允许即可。

8.1.7 host.conf 文件

- 路径
  /etc/host.conf
- 作用
  设置系统的配置信息。当系统中同时存在 DNS 域名解析和/etc/hosts 主机表机制时，由/etc/host.conf 确定主机名解释顺序。
- 参数设置
  文件包含了为解析库声明的配置信息，它应该每行含一个配置关键字，其后跟着合适的配置信息。
  系统识别的关键字有 order、trim、multi、nospoof 和 reorder。
  - Order
    确定了主机查询是如何执行的。它后面应该跟随一个或者更多的查询方式，这些查询方式用逗号分隔。有效的方式有：bind、hosts 和 nis。
  - trim
    可以多次出现。每次出现其后应该跟随单个的以句点开头的域名。如果设置

了它，resolver 库会自动截去任何通过 DNS 解析出来的主机名后面的域名。这个选项用于本地主机和域。

- multi

  有效的值为 on 和 off。如果设置为 on，resolver 库会返回一台主机在/etc/hosts 文件中出现的的所有有效地址，而不只是第一个。默认情况下设为 off。

- nospoof

  有效的值为 on 和 off。如果设置为 on，resolver 库会尝试阻止主机名欺骗以提高使用 rlogin 和 rsh 的安全性。在执行了一个主机地址的查询之后，resolver 会对该地址执行一次主机名的查询；如果两者不匹配，查询即失败。

- spoofalert

  如果该选项设为 on，同时也设置了 nospoof 选项，resolver 会通过 syslog 设施记录错误报警信息，默认的值为 off。

- reorder

  有效的值为 on 和 off。如果设置为 on，resolver 会试图重新排列主机地址，以便执行 gethostbyname(3)时，首先列出本地地址木即在同一子网中的地址朩，重新排序适合于所有查询方式，默认的值为 off。

- 例样

```
order hosts,bind
multi on
```

8.1.8 ifconfig 命令

您还可以使用 ifconfig 命令来配置和查看网卡参数。

当有多块网卡时，系统中网卡命名规律：eth0,eth1,eth2...,另外，lo 为环回接口，它的 IP 地址固定为 127.0.0.1，掩码 8 位。它代表你的机器本身。

1、查看网卡信息

ifconfig [Interface]

Interface 是可选项，如果不加此项，则显示系统中所有网卡的信息。如果添加此选项则显示所指定的网卡信息。

如:

```
$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:11:D8:6D:C5:A1
          inet addr:192.168.0.1 Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::211:d8ff:fe6d:c5a1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST   MTU:1500   Metric:1
          RX packets:9771 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8802 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:958022 (935.5 Kb)  TX bytes:9606400 (9.1 Mb)
          Interrupt:17

lo        Link encap:Local Loopback
          inet addr:127.0.0.1    Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436 Metric:1
          RX packets:824 errors:0 dropped:0 overruns:0 frame:0
          TX packets:824 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:374847 (366.0 Kb)  TX bytes:374847 (366.0 Kb)
```

- Link encap
  连接类型，Ethernet 指以太网。
- HWaddr
  硬件 mac 地址。
- inet addr
  网卡的 IP 地址。
- Bcast
  网卡的广播地址。
- Mask
  网卡的子网掩码。
- inet6
  网卡的 IPv6 地址。
- Scope
  范围。
- UP BROADCAST RUNNING MULTICAST MTU:1500

接口的活动类型。

- UP 网卡开启状态
- BROADCAST 网卡支持广播
- RUNNING 网卡的网线被接上
- MULTICAST 网卡支持组播
- MTU:1500 最大传输单元为 1500 字节

- RX packets
  接收数据包情况统计。
- TX packets
  发送数据包情况统计。
- RX bytes
  接收数据字节数统计信息。
- TX bytes
  发送数据字节数统计信息。

2、选项

在 ifconfig 命令中可以指定许多选项以改变其行为:

- -a
  该选项告诉 ifconfig 显示所有接口信息，包括活动的和非活动的。
- -s
  这是一个“短列表”选项，它为每个接口显示一行摘要数据。该返回信息是有关接口活动性的，并且没有配置。该输出和 netstat -i 命令的返回内容是一样的。
- -v
  这个“详细的”选项在满足某些类型的错误条件时返回额外信息以帮助发现并处理故障。

3、配置网卡参数

除此之外，还有几个选项可以配置网卡参数。

- 配置网卡的 IP 地址
  ifconifg [Interface] [addr]
  只在接口名称后指定一个地址，将设置该接口的 IP 地址。
  如，在 eth0 上配置 192.168.0.1 的 IP 地址，使用以下命令:

```
$ ifconfig eth0 192.168.0.1
```

若想在 eth0 上再配置一个 192.168.1.1/24 的 IP 地址，使用以下命令:

```
$ ifconfig eth0:0 192.168.1.1
```

这时再用 ifconifg 命令查看，就可以看到两个网卡的信息了，分别为 eth0 和 eth0:0。若还想再增加 IP，那网卡的命名就接着是 eth0:1、eth0:2...

- 配置网卡的子网掩码

  ifconifg [Interface] netmask [addr]

  使用“netmask”选项使得你可以为一个给定接口设置网络掩码。

  如，在 eth0 上配置 24 位掩码，使用以下命令：

  ```
  $ ifconfig eth0 netmask 255.255.255.0
  ```

- 配置网卡的广播地址

  ifconifg [Interface] broadcast [addr]

  当“broadcast”选项后面跟随一个地址参数，那么将设置指定接口的广播地址。

  如，在 eth0 上配置广播地址为 192.168.0.255，使用以下命令：

  ```
  $ ifconfig eth0 broadcast 192.168.0.255
  ```

- 配置网卡的硬件地址

  ```
  $ ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx
  ```

- 启用网卡

  ifconfig [Interface] up

  如果一个接口不是活动的，该选项将激活它。

  如，启动网卡 eth0：

  ```
  $ ifconfig eth0 up
  ```

- 禁用网卡

  ifconfig [Interface] down

  与 up 相反，它使指定接口无效。

  如，禁用网卡 eth0：

  ```
  $ ifconfig eth0 down
  ```

8.2 网络服务

凝思安全操作系统提供下列网络服务:

- Apache
- BIND
- Samba
- SSH

所有服务都通过适当修改和配置，充分发挥了凝思安全操作系统的安全机制，能够完全防止缓冲器溢出攻击，大大增强了网络服务的安全性。

1、运行服务

以管理员用户登录，对所有网络服务可以进行配置、启动、停止。各服务的启动脚本位于/etc/init.d/下。

1）启动服务

例如，要启动 apache2 服务，以管理员登录，执行命令:

```
/etc/init.d/apache2 start
```

2）停止服务

例如，要停止 apache2 服务，以管理员登录，执行命令:

```
/etc/init.d/apache2 stop
```

2、设置开机自动启动

下面以设置 apache2 服务开机自动启动为例。

将 apache2 启动脚本复制到 /etc/init.d 目录下（一般安装软件时会自动装入，也可以手动复制），并命名为 apache2。

```
cp /usr/sbin/apachectl /etc/init.d/apache2
```

查看系统的当前启动级别（系统默认为“N 2”）。

```
runlevels
```

进入系统开机启动级别（例如“N 2”）目录，建立软件启动文件的符号链接。S 表示 start（启动），K 表示 kill（关闭），数字大小代表启动顺序的先后，数字大的启动顺序排在后面。

```
cd /etc/rc2.d
ln -s /etc/init.d/apache2 S20apache2
```

这样，apache2 服务将在开机时自动启动。

您还可以使用 update-rc.d 命令设置启动级别:

1）从（所有的）运行级别配置目录中是删除指定的服务（K 和 S）

update-rc.d -f <service> remove

2）配置服务在运行级别列表中按指定的顺序启动

update-rc.d -f <service> start <order> <runlevels>

3）配置服务在运行级别列表中指定的顺序停止

update-rc.d -f <service> stop <order> <runlevels>

例如：

$ update-rc.d -f apache2 remove
$ update-rc.d apache2 stop 91 0 1 2 3 4 5 6 .
（注意带着“.”）

第一个命令移除了所有的指向 /etc/init.d/apache2 服务脚本的运行级别链接，-f 选项将会使得 update-rc.d 在 apache2 脚本本身已经存在的情况下仍然进行相应的处理。

第二个命令在每一个运行级别创建了一个服务顺序为 91 的停止脚本。例如，这会创建一个/etc/rc3.d/K91apache2 的符号链接来禁止运行级别 3 中 apache2 的运行。（这里要注意的是要添加 reboot 与 shutdown 运行级别，即 0 和 6，尽管其中所有的服务都会被设置成为“stop”）。

作为另外的一个例子，下面所显示的命令将会在运行级别 3、4、5 允许 apache2 的运行，而在其他的运行级别中则被禁止：

$ update-rc.d -f apache2 remove
$ update-rc.d apache2 start 91 3 4 5 . stop 91 0 1 2 6 .
（注意带着“.”）

提示

每个运行级别中，同一个服务的符号链接只能有一个要么 K 开头的要么 S 开头的，不能同时存在。

下面逐一说明各个网络服务的配置。

8.2.1 Apache

8.2.1.1 简介

Apache 服务用于实现 WEB 服务，它是 Internet 网上应用最为广泛的 Web 服务器软件之一。Apache 服务器源自美国国家超级技术计算应用中心（NCSA）的 Web 服务器项目中。 目前已在互联网中占据了领导地位。Apache 服务器经过配置之后，能适应高负荷，大吞吐量的互联网工作。

8.2.1.2 配置文件

Apache 的主配置文件 apache2.conf 位于/etc/apache2 目录下，提供了最基本的服务器配置。这些配置文件控制着服务器各个方面的特性，因此，为了正常运行服务器便需要设置好这些文件。

除了配置文件之外，Apache 还使用 mime.types 文件用于标识不同文件对应的 MIME 类型，magic 文件设置不同 MIME 类型文件的特殊标识，当 Apache 服务器从文档后缀不能判断出文件的 MIME 类型时，能通过文件内容中的这些特殊标记来判断文档的 MIME 类型。

在主配置文件 apache2.conf 中包含了以下文件：

- 动态模块的配置：
  /etc/apache2/mods-enabled/*.load
  /etc/apache2/mods-enabled/*.conf
- 端口监听的配置：
  /etc/apache2/ports.conf
- 虚拟主机的配置指令：
  /etc/apache2/sites-enabled/

8.2.1.3 配置文件参数

8.2.1.3.1 /etc/apache2/apache2.conf

1、安装服务器的基础目录。

格式：ServerRoot directory-path

```
ServerRoot “/etc/apache2”
```

ServerRoot 用于指定守护进程 apache2 的运行目录，apache2 启动之后会自动将进程的根目录改变为这个目录，因此如果设置文件中指定的文件或目录是相对路径，那么真实路径就位于这个 ServerRoot 定义的路径之下。

2、设置用户和组

```
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
```

User 和 Group 配置 Apache 的安全保证，Apache 在打开端口之后，就将其本身设置为这两个选项设置的用户和组权限进行运行，这样就降低了服务器的危险性。凝思安全操作系统通过能力机制允许任何用户在任何端口上（可以小于 1024）运行 Apache。

APACHE_RUN_USER 和 APACHE_RUN_GROUP 定义在配置文件 envvars 中，Apache2 的缺省设置为 www-data 和 www-data。一般情况下要为 Web 服务设定一个特定的用户和组，同

时在这里更改用户和组设置。凝思安全操作系统缺省的 Apache 用户和组是 netadmin 和 netadmin。

3、对根目录访问控制

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Require all denied
</Directory>
```

Apache 服务器可以针对目录进行文档的访问控制，然而访问控制可以通过两种方式来实现，一个是在配置文件 apache2.conf 中针对每个目录进行设置，另一个方法是在每个目录下设置访问控制文件，通常访问控制文件名字为.htaccess。虽然使用这两个方式都能用于控制浏览器的访问，然而第一种方法要求每次改动后重新启动 apache2 守护进程，比较不灵活，因此主要用于配置服务器系统的整体安全控制策略，而使用每个目录下的.htaccess 文件设置具体目录的访问控制更为灵活方便。

Directory 语句就是用来定义目录的访问限制，这里可以看出它为一个目录定义访问限制的标准语法。上例的这个设置是针对系统的根目录进行的，设置了允许符号连接的选项 FollowSymLinks，以及使用 AllowOverride None 表示不允许这个目录下的访问控制文件改变这里进行的配置，这也意味着不用查看这个目录下的相应访问控制文件。

由于 Apache 对一个目录的访问控制设置是能够被下一级目录继承的，因此对根目录的设置将影响到它的下级目录。注意由于 AllowOverride None 的设置，使得 Apache 服务器不需要查看根目录下的访问控制文件，也不需要查看以下各级目录下的访问控制文件，直至 apache2.conf 中为某个目录指定了允许 Alloworride，即允许查看访问控制文件。由于 Apache 对目录访问控制是采用的继承方式，如果从根目录就允许查看访问控制文件，那么 Apache 就必须一级一级的查看访问控制文件，对系统性能会造成影响。而缺省关闭了根目录的这个特性，就使得 Apache 从 apache2.conf 中具体指定的目录向下搜寻，减少了搜寻的级数，增加了系统性能。因此对于系统根目录设置 AllowOverride None 不但对于系统安全有帮助，也有益于系统性能。

4、对文档根目录访问控制

```
<Directory /var/www/html>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow, deny
    Allow from all
</Directory>
```

这里定义的是系统对外发布文档的目录的访问设置，设置不同的 AllowOverride 选项，以定义配置文件中的目录设置和用户目录下的安全控制文件的关系，而 Options 选项用于定义该目录的特性。

4.1、AllowOverride 的设置

对每个目录访问控制文件作用的影响。

配置文件和每个目录下的访问控制文件都可以设置访问限制，配置文件是由管理员设置的，而每个目录下的访问控制文件是由目录的属主设置的，因此管理员可以规定目录的属主是否能覆盖系统在设置文件中的设置，这就需要使用 AllowOverride 参数进行设置，通常可以设置的值为：

- All
  缺省值，使访问控制文件可以覆盖系统配置。
- None
  服务器忽略访问控制文件的设置。
- Options
  允许访问控制文件中可以使用 Options 参数定义目录的选项。
- FileInfo
  允许访问控制文件中可以使用 AddType 等参数设置。
- AuthConfig
  允许访问控制文件使用 AuthName，AuthType 等针对每个用户的认证机制，这使目录属主能用口令和用户名保护目录。
- Limit
  允许对访问目录的客户机的 IP 地址和名字进行限制。

4.2、Options 的设置

服务器特性设置。

每个目录具备一定属性，可以使用 Options 来控制这个目录下的一些访问特性设置，以下为常用的特性选项：

- All
  所有的目录特性都有效，这是缺省状态。
- None
  所有的目录特性都无效。
- FollowSymLinks
  允许使用符号连接，这将使浏览器有可能访问文档根目录（DocumentRoot）之外的文档。
- SymLinksIfOwnerMatch
  只有符号连接的目的与符号连接本身为同一用户所拥有时，才允许访问，这个设置将增加一些安全性。
- ExecCGI
  允许这个目录下可以执行 CGI 程序。
- Indexes
  允许浏览器可以生成这个目录下所有文件的索引，使得在这个目录下没有 index.html（或其它索引文件）时，能向浏览器发送这个目录下的文件列表。

此外，上例中还使用了 Order，Allow，Deny 等参数，这是 Limit 语句中用来根据浏览器的域名和 IP 地址来控制访问的一种方式。其中 Order 定义处理 Allow 和 Deny 的顺序，而

Allow，Deny 则针对名字或杉材进行访问控制设置，上例使用 allow from all，表示允许所有的客户机访问这个目录，而不进行任何限制。

5、对 URL 的访问控制

```
<Location /server-status>
    SetHandler server-status
    Order deny, allow
    Deny from all
    Allow from .your_domain.com
</Location>

<Location /server-info>
    SetHandler server-info
    Order deny，allow
    Deny from all
    Allow from .your_domain.com
</Location>
```

用于设置访问控制的设置主要是针对目录和文件进行设置的，然而也可以针对不同的 URL 进行访问控制的设置，这样就不必担心 ScriptAlias，Alias 是否将路径设置到了受控制的目录之外。针对 URL 进行控制的语句为 Location，这样不但能对服务器上的文件，CHI 提供保护，此外，它还能保护不能找到对应文件，而是由服务器本身提供的特殊功能 URL。http://servername/server-status 用于报告当前 Apache 服务器的状态，http://servername/server-info 用于报告 Apache 服务器的统计信息。与此相关的设置还有 ExtendedStatus，可以让服务器输出更详细的报告。

6、访问控制文件

```
AccessFileName .htaccess
```

AccessFileName 定义每个目录下的访问控制文件的文件名，缺省为.htaccess，可以通过更改这个文件，来改变不同目录的访问控制限制。

```
<FilesMatch "^\.ht">
    Order allow，deny
    Deny from all
    Satisfy All
</FilesMatch>
```

除了可以针对目录进行访问控制之外，还可以根据文件来设置访问控制，这就是 File 语句的任务。使用 File 语句，不管文件处于哪个目录，只要名字匹配，就必须接受相应的访问控制。这个语句对于系统安全比较重要，例如上例将屏蔽所有的使用者不能访问.htaccess 文件，这样.htaccess 中的安全信息就不能被客户获取。

7、日志记录

```
    ErrorLog "/var/log/apache2/error_log"
    LogLevel warn
    LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
vhost_combined
    LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
    LogFormat "%h %l %u %t \"%r\" %>s %O" common
    LogFormat "%{Referer}i -> %U" referer
    LogFormat "%{User-agent}i" agent
```

这里定义了系统日志的形式。对于服务器错误记录，由 ErrorLog 指定记录错误日志的文件，LogLevel 指定要记录的内容。

对于系统的访问日志，缺省使用 CustomLog 参数定义日志的位置。可以使用 combined 参数指定将所有的访问日志放在一个文件中，也可以将不同种类的访问日志放在不同的日志记录文件中，这是通过在 CustomLog 中指定不同的记录类型来完成的。

- common 表示普通的对单页面请求访问记录；
- referer 表示每个页面的引用记录，可以看出一个页面中包含的请求数；
- agent 表示对客户机的类型记录。

显然可以将现有的 combined 定义的设置行注释掉，并使用 common、referer 和 agent 作为 LogFormat 的参数，来为不同种类的日志分别指定日志记录文件。

显然，LogFormat 是用于定义不同类型的日志进行记录时使用的格式，这里使用了以%开头的宏定义，以记录不同的内容。如果这些参数指定的文件使用的是相对路径，那么就是相对于 ServerRoot 的路径。

8、在服务器配置文件中包含其它配置文件

格式：Include file-path|directory-path

这个参数允许在服务器配置文件中加入其它配置文件。

shell 风格（fnmatch()）的通配符可以用于按照字母顺序一次包含多个文件。另外，如果 Include 指向了一个目录而不是一个文件，Apache 将读入该目录及其子目录下的所有文件，并依照字母顺序将这些文件作为配置文件进行解析。但是并不推荐这么做，因为偶尔会有临时文件在这个目录中生成，从而导致 Apache 启动失败。

文件的路径可以是一个完整的绝对路径（以一个斜杠开头），如：

```
Include /etc/apache2/mods-available/ssl.conf
Include /etc/apache2/mods-enabled/*.conf
```

或是相对于 ServerRoot 目录（假设为/etc/apache2/）的相对路径，如：

```
Include mods-available/ssl.conf
Include mods-enabled/*.conf
```

请确保包含的目录中不包含任何诸如编辑器临时文件等引起误导的文件，因为 Apache 会尝试读取它们并把其中的内容作为配置参数来处理，这样可能会导致启动过程的失败。运行 apachectl configtest 将会把配置检查时所使用的所有文件列出来以供参考。这将有助于检验配置中是否仅包含了您所希望出现那些文件。

这个参数 Include 可用于与使用 srm.conf 和 access.conf 设置文件的老版本 Apache 兼容。如果没有兼容的需要，可以将设置文件指定为/dev/null，即 Include /dev/null，这将表示不存在其

它设置文件，而仅使用 apache2.conf 一个文件来保存所有的设置选项。

9、指定记录 apache2 服务进程号的文件

PidFile ${APACHE_PID_FILE}

PidFile 指定的文件将记录 apache2 守护进程的进程号，由于 apache2 能自动复制其自身，因此系统中有多个 apache2 进程，但只有一个进程为最初启动的进程，它是其它进程的父进程，对这个进程发送信号将影响所有的 apache2 进程。PidFile 定义的文件中就记录 apache2 父进程的进程号。

APACHE_PID_FILE 定义在/etc/apache2/envvars 中。

10、指定锁文件位置

Mutex file:${APACHE_LOCK_DIR} default

接受串行锁文件的位置。

由于 apache2 会经常进行并发的文件操作，就需要使用加锁的方式来保证文件操作不冲突，由于 NFS 文件系统在文件加锁方面能力有限，因此这个目录应该是本地磁盘文件系统，而不应该使用 NFS 文件系统。

LockFile 指定了 apache2 守护进程的加锁文件，一般不需要设置这个参数，Apache 服务器将自动在 ServerRoot 下面的路径中进行操作。但如果 ServerRoot 为 NFS 文件系统，便需要使用这个参数指定本地文件系统中的路径。

APACHE_LOCK_DIR 定义在/etc/apache2/envvars 中。

重要

最好不要将此文件放在任何人都可以具有写权限的目录（比如 /var/tmp ）中，因为别人可以通过建立一个与服务器企图建立的锁文件同名的文件，来阻止服务器启动，从而造成一个拒绝服务攻击。

11、超时间隔

Timeout 300

Timeout 定义客户程序和服务器连接的超时间隔，超过这个时间间隔（秒）后服务器将断开与客户机的连接。

12、保持连接

KeepAlive On

在 HTTP 1.0 中，一次连接只能作传输一次 HTTP 请求，而 KeepAlive 用于支持 HTTP 1.1 版本的一次连接，多次传输功能，这样就可以在一次连接中传递多个 HTTP 请求。虽然只有

较新的浏览器才支持这个功能，但还是打开这个选项。打开的结果是允许用户建立永久连接。

13、一次连接可请求的最大次数

MaxKeepAliveRequests 100

MaxKeepAliveRequests 为一次连接可以进行的 HTTP 请求的最大次数。将其值设为 0 将支持在一次连接内进行无限次的传输请求。事实上没有客户程序在一次连接中请求太多的页面，通常达不到这个上限就完成连接了。

14、多次请求间隔

KeepAliveTimeout 15

KeepAliveTimeout 测试一次连接中的多次请求传输之间的时间，如果服务器已经完成了一次请求，但一直没有接收到客户程序的下一次请求，在间隔超过了这个参数设置的值之后，服务器就断开连接。

15、主机名记入日志

HostnameLookups Off

此参数启用了 DNS 查询，使得主机名能被记入日志（并用 REMOTE_HOST 变量传递给 CGI/SSI）。参数 Double 指定进行一次双向 DNS 查询。也就是说在一次反向查询之后，再对返回的结果进行一次正向查询。在正向查询结果中至少应该有一个 IP 地址与初始的地址相符。

不论此处如何设置，当您使用 mod_authz_host 来根据主机名控制访问的时候，就会执行一次双向查询。这对安全来说非常必要。请注意如果您没有设置 HostnameLookups Double，这种双向查询的结果不是自动生成的。比如说：如果仅仅设置了 HostnameLookups On 而且请求是针对一个根据主机名做了限制的对象，不论双向查询是否失败，CGI 还是会把单向查询的结果用 REMOTE_HOST 来传送。

默认值设置为 Off 是为了那些不需要进行反向查询的站点节约网络带宽考虑的。这对最终用户也是有益的，因为这样他们就不用忍受查询造成的延迟了。高访问量的网站应该将此参数设置为 Off 因为 DNS 查询会造成明显的时间消耗。在 bin 目录下的 logresolve 工具可以在离线的情况下对已经记入日志的 IP 地址进行主机名的查询。

8.2.1.3.2 /etc/apache2/ports.conf

1、服务器监听的 IP 地址和端口。

```
Listen 80
#Listen 12.34.56.78:80
```

格式：Listen [IP-address:]portnumber [protocol]

Listen 指示 Apache 只在指定的 IP 地址和端口上监听；默认情况下 Apache 会在所有 IP 地址上监听。Listen 是必须设置的。如果在配置文件中找不到这个参数，服务器将无法启动。如果只指定一个端口，服务器将在所有地址上监听该端口。如果指定了地址和端口的组合，服务器将在指定地址的指定端口上监听。

使用多个 Listen 可以指定多个不同的监听端口或地址端口组合。服务器将会对列出的所有端口和地址端口组合上的请求作出应答。

例如，想要服务器接受 80 和 8000 端口上的请求，可以这样设置：

```
Listen 80
Listen 8000
```

为了让服务器在两个确定的地址端口组合上接受请求，可以这样设置：

```
Listen 192.170.2.1:80
Listen 192.170.2.5:8000
```

IPv6 地址必须用方括号括起来：

```
Listen [2001:db8::a00:20ff:fea7:ccea]:80
```

可选的 protocol 参数在大多数情况下并不需要。若未指定该参数，则将为 443 端口使用默认的 https 协议，为其它端口使用 http 协议。在这里指定协议是为了确定使用哪个模块来处理请求，以及根据 Accept Filter 根据不同的协议有针对性的进行优化。

仅在使用非标准端口时才需要指定 protocol 参数。比如在 8443 端口运行 https 协议：

```
Listen 192.170.2.1:8443 https
```

> **注意**
>
> 多个 Listen 指定了同一个地址和端口的组合后，会导致“ Address already in use” 错误。

8.2.1.3.3 /etc/apache2/sites-available/000-default.conf

1、管理员 email 地址

```
ServerAdmin webmaster@localhost
```

配置文件中应该改变的也许只有 ServerAdmin，这一项用于配置 web 服务器的管理员的 email 地址，这将在 HTTP 服务出现错误的条件下返回给浏览器，以便让 Web 使用者和管理员联系，报告错误。习惯上使用服务器上的 webmaster 作为 web 服务器的管理员，通过邮件服务器的别名机制，将发送给 webmaster 的电子邮件转发给真正的 Web 管理员。

2、服务器用于辨识自己的主机名和端口号。

```
ServerName new.host.name:80
```

格式：ServerName [scheme://]fully-qualified-domain-name[:port]

ServerName 设置了服务器用于辨识自己的主机名和端口号。这主要用于创建重定向 URL。

通常一个 Web 服务器可以具有多个名字，客户浏览器可以使用所有这些名字或 IP 地址访问这台服务器，但在没有定义虚拟主机的情况下，服务器总是以自己的正式名 字回应浏览器。ServerName 就定义了 Web 服务器自己承认的正式名字，例如一台服务器名字（在 DNS 中定义了 A 类型）为 exmaple.org.cn，同时为了方便记忆还定义了一个别名（CNAME 记录）为 www.exmaple.org.cn，那么 Apache 自动解析得到的名字就为 exmaple.org.cn，这样不管客户浏览器使用哪个名字发送请求，服务器总是告诉客户程序自己为 exmaple.org.cn。虽然这一般并不会造成什么问题，但是考虑到某一天服务器可能迁移到其它计算机上，而只想通过更改 DNS 中的 www 别名配置就完成迁移任务，所以不想让客户在其书签中使用 Linux 记录下这个服务器的地址，就必须使用 ServerName 来重新指定服务器的正式名字。

当没有指定 ServerName 时，服务器会尝试对 IP 址进行反向查询来推断主机名。但如果服务器的名字解析有问题（通常为反向解析不正确），或者没有正式的 DNS 名字，也可以在这里指定 IP 地址。当 ServerName 设置不正确的时候，服务器不能正常启动。如果在 ServerName 中没有指定端口号，服务器会使用接受请求的那个端口。为了加强可靠性和可预测性，您应该使用 ServerName 显式的指定一个主机名和端口号。

如果使用的是基于域名的虚拟主机，在<VirtualHost>段中的 ServerName 将是为了匹配这个虚拟主机，在“Host:”请求头中必须出现的主机名。

3、文档根目录

DocumentRoot /var/www/html

DocumentRoot 定义这个服务器对外发布的超文本文档存放的路径，客户程序请求的 URL 就被映射为这个目录下的网页文件。这个目录下的子目录，以及使用符号连接指出的文件和目录都能被浏览器访问，只是要在 URL 上使用同样的相对目录名。

注意：符号连接虽然逻辑上位于根文档目录之下，但实际上可以位于计算机上的任意目录中，因此可以使客户程序能访问那些根文档目录之外的目录，这在增加了灵活性的同时减少了安全性。Apache 在目录的访问控制中提供了 FollowSymLinks 选项来打开或关闭支持符号连接的特性。

8.2.1.3.4 /etc/apache2/mods-available/dir.conf

1、网站首页文件名

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.php
</IfModule>
```

很多情况下，URL 中并没有指定文档的名字，而只是给出了一个目录名。那么 Apache 服务器就自动返回这个目录下由 DirectoryIndex 定义的文件，当然可以指定多个文件名字，系统会在这个目录下顺序搜索。当所有由 DirectoryIndex 指定的文件都不存在时，Apache 服务器可以根据系统设置，生成这个目录下的所有文件列表，提供用户选择。此时该目录的访问控制选项中的 Indexes 选项（Options Indexes）必须打开，使得服务器能够生成目录列表，否则 Apache 将拒绝访问。

8.2.1.3.5 /etc/apache2/mods-available/alias.conf：

1、别名映射

```
<IfModule alias_module>
    Alias /test "/var/www/html/test"
    <Directory "/var/www/html/test">
        Options Indexes MultiViews
        AllowOverride AuthConfig
        Order allow,deny
        Allow from all
    </Directory>

    ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
    <Directory "/var/www/cgi-bin">
        AllowOverride None
        Options None
        Order allow,deny
        Allow from all
    </Directory>

    Alias /icons/ "/var/www/icons/"
    <Directory "/var/www/icons">
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</IfModule>
```

Alias 用于将 URL 与服务器文件系统中的真实位置进行直接映射，一般的文档将在 DocumentRoot 中进行查询，然而使用 Alias 定义的路径将直接映射到相应目录下，而不再到 DocumentRoot 下面进行查询。因此 Alias 可以用来映射一些公用文件的路径，例如保存了各种常用图标的 icons 路径。这样使得除了使用符号连接之外，文档根目录（DocumentRoot）

外的目录也可以通过使用了 Alias 映射，提供给浏览器访问。

定义好映射的路径之后，应该需要使用 Directory 语句设置访问限制。

ScriptAlias 也是用于 URL 路径的映射，但与 Alias 的不同在于，ScriptAlias 是用于映射 CGI 程序的路径，这个路径下的文件都被定义为 CGI 程序，通过执行它们获得结果，而不是由服务器直接返回其内容。缺省情况下 CGI 程序使用 cgi-bin 目录作为虚拟路径。

8.2.1.3.6 /etc/apache2/mods-available/mime.conf

1、设置数据类型

```
<IfModule mod_mime.c>
    TypesConfig /etc/mime.types
    #AddType application/x-gzip .tgz
    #AddEncoding x-compress .Z
    #AddEncoding x-gzip .gz .tgz
    AddType application/x-compress .Z
    AddType application/x-gzip .gz .tgz
    AddType application/x-httpd-php .php
    #AddHandler cgi-script .cgi
    #AddHandler type-map var
    #AddType text/html .shtml
    #AddOutputFilter INCLUDES .shtml
</IfModule>
```

TypeConfig 用于设置保存不同的 MIME 类型数据的文件名，缺省设置为/etc/mime.types。

AddEncoding

AddType 可以为特定后缀的文件指定 MIME 类型，这里的设置将覆盖 mime.types 中的设置。

AddHandler 是用于指定非静态的处理类型，用于定义文档为一个非静态的文档类型，需要进行处理，再向浏览器返回处理结果。例如上面注释中的设置是将以.cgi 结尾的文件设置为 cgi-script 类型，那么服务器将启动这个 CGI 程序以进行处理。如果需要在前面 AliasScript 定义的路径之外执行 CGI 程序，就需要使用这个参数进行设置，此后以.cgi 结尾的文件将被当作 CGI 程序执行。

AddOutputFilter

2、语言设置

```
AddLanguage ca .ca
AddLanguage cs .cz .cs
AddLanguage da .dk
AddLanguage de .de
AddLanguage el .el
AddLanguage en .en
AddLanguage eo .eo
AddLanguage es .es
AddLanguage et .et
AddLanguage fr .fr
AddLanguage he .he
AddLanguage hr .hr
AddLanguage it .it
AddLanguage ja .ja
AddLanguage ko .ko
AddLanguage ltz .ltz
AddLanguage nl .nl
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pl .po
AddLanguage pt .pt
```

```
AddLanguage zh-CN .zh-cn
    AddLanguage zh-TW .zh-tw
    LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl          pt pt-BR ru sv
    tr zh-CN zh-TW
    ForceLanguagePriority Prefer Fallback
    AddCharset us-ascii.ascii .us-ascii
    AddCharset ISO-8859-1 .iso8859-1      .latin1
    AddCharset ISO-8859-2 .iso8859-2      .latin2 .cen
    AddCharset ISO-8859-3 .iso8859-3      .latin3
    AddCharset ISO-8859-4 .iso8859-4      .latin4
    AddCharset ISO-8859-5 .iso8859-5      .cyr .iso-ru
    AddCharset ISO-8859-6 .iso8859-6      .arb .arabic
    AddCharset ISO-8859-7 .iso8859-7      .grk .greek
    AddCharset ISO-8859-8 .iso8859-8      .heb .hebrew
    AddCharset ISO-8859-9 .iso8859-9      .latin5 .trk
    AddCharset ISO-8859-10      .iso8859-10     .latin6
    AddCharset ISO-8859-13      .iso8859-13
    AddCharset ISO-8859-14      .iso8859-14     .latin8
    AddCharset ISO-8859-15      .iso8859-15     .latin9
    AddCharset ISO-8859-16      .iso8859-16     .latin10
    AddCharset ISO-2022-JP .iso2022-jp .jis
    AddCharset ISO-2022-KR .iso2022-kr .kis
    AddCharset ISO-2022-CN .iso2022-cn .cis
    AddCharset Big5.Big5    .big5 .b5
    AddCharset cn-Big5 .cn-big5
    AddCharset WINDOWS-1251 .cp-1251       .win-1251
    AddCharset CP866 .cp866
    AddCharset KOI8   .koi8
    AddCharset KOI8-E       .koi8-e
    AddCharset KOI8-r.koi8-r .koi8-ru
    AddCharset KOI8-U       .koi8-u
    AddCharset KOI8-ru .koi8-uk .ua
    AddCharset ISO-10646-UCS-2 .ucs2
    AddCharset ISO-10646-UCS-4 .ucs4
    AddCharset UTF-7 .utf7
    AddCharset UTF-8 .utf8
    AddCharset UTF-16       .utf16
    AddCharset UTF-16BE .utf16be
    AddCharset UTF-16LE .utf16le
    AddCharset UTF-32       .utf32
    AddCharset UTF-32BE .utf32be
    AddCharset UTF-32LE .utf32le
    AddCharset euc-cn .euc-cn
    AddCharset euc-gb .euc-gb
    AddCharset euc-jp  .euc-jp
    AddCharset euc-kr  .euc-kr
    AddCharset EUC-TW     .euc-tw
    AddCharset gb2312       .gb2312 .gb
    AddCharset iso-10646-ucs-2 .ucs-2 .iso-10646-ucs-2
    AddCharset iso-10646-ucs-4 .ucs-4 .iso-10646-ucs-4
    AddCharset shift_jis .shift_jis .sjis
```

AddLanguage 在文件扩展名与特定的语言之间建立映射。定义以 extension 为扩展名的文件是以 MIME-lang 语言写成的。这个映射关系会添加在所有有效的映射关系上，并覆盖所有相同的 extension 扩展名映射。虽然内容的语言会返回给客户端，但浏览器一般未必会使用这一信息。AddLanguage 更多的用于内容协商，以决定哪个文档应当被返回给用户。如果同一个扩展名被赋予多个语言，那么使用最后出现的那个。

在处理 MultiViews 请求时，LanguagePriority 在客户没有指示语言偏爱的情况下，设定语言变体的优先级列表。这个 MIME-lang 列表是按优先级降序排列的。

> 注意
>
> 本参数只在根据其它信息无法决定最好的语言或者 ForceLanguagePriority 不是 None 时才有效。对正确实现的 HTTP/1.1 请求，本参数没有任何作用。

ForceLanguagePriority 使用 LanguagePriority 的设置，在服务器无法返回单个匹配文档的情况下，指定完成协商过程的方法。ForceLanguagePriority Prefer 在有几个等价选择的情况下，使用 LanguagePriority 的设定以提供一个有效的结果，而不是返回 HTTP 结果 300（多重选择）。ForceLanguagePriority Fallback 使用 LanguagePriority 在无法找到合适结果的情况下，指定一个有效的结果，而不是返回 HTTP 结果 406（不可接受）。Prefer 和 Fallback 两个选项可以同时指定，这样在有一个以上有效变体的情况下，返回 LanguagePriority 列表中第一个匹配的变体文档，而在没有变体能够匹配客户可接受的语言的情况下，返回第一个可用的变体文档。

AddCharset 在特定的文件扩展名与特定的字符集之间建立映射。charset 是以 extension 为扩展名的文件的 MIME 字符集参数。这个映射关系会强制添加在所有现存的映射关系上，并覆盖所有现存的 extension 扩展名映射。AddCharset 除了用于通知客户端文档的字符集编码方式以便正确地翻译和显示以外，还用于内容协商（根据用户的优先选择信息，从几个文档中选择一个返回给用户）。extension 参数是大小写无关的，并且可以带或不带前导点。

8.2.1.3.7 /etc/apache2/mods-available/mime_magic.conf

1、文件特性

```
<IfModule mod_mime_magic.c>
    MIMEMagicFile /etc/apache2/magic
</IfModule>
```

除了从文件的后缀出发来判断文件的 MIME 类型之外，Apache 还可以进一步分析文件的一些特征，来判断文件的真实 MIME 类型。这个功能是由 mod_mime_magic 模块实现的，它需要一个记录各种 MIME 类型特征的文件，以进行分析判断。上面的设置 是一个条件语句，

如果载入了这个模块，就必须指定相应的标志文件 magic 的位置。

8.2.1.3.8 /etc/apache2/conf-available/localized-error-pages.conf：

1、错误代码

```
#ErrorDocument 500  "The server made a boo boo."
#ErrorDocument 404  /missing.html
#ErrorDocument 404  /cgi-bin/missing_handler.pl
#ErrorDocument     402 http://some.other_server.com/subscription_info.html
```

如果客户请求的网页不存在，或者没有访问权限等情况发生时，服务器将产生一个错误代码，同时也将回应客户浏览器一个标识错误的网页。ErrorDocument 就用于设置当出现哪个错误时应该回应客户浏览器哪些内容，ErrorDocument 的第一个参数为错误的序号，第二个参数为回应的数据，可以为简单的文本，本地网页，本地 CGI 程序，以及远程主机上的网页。

8.2.1.3.9 /etc/apache2/mods-available/mpm_prefork.conf

1、启动时子进程数

```
StartServers 5
```

StartServers 就是用来设置 apache2 启动时产生的子进程副本数量，这个参数与上面定义的 MinSpareServers 和 MaxSpareServers 相关，都是用于启动空闲子进程以提高服务器的反应速度的。这个参数应该设置为前两个值之间的一个数值，小于 MinSpareServers 和大于 MaxSpareServers 都没有意义。

2、最大/最小子进程数

```
MinSpareServers 5
MaxSpareServers 10
```

在使用子进程处理 HTTP 请求的 Web 服务器上，由于要首先生成子进程才能处理客户的请求，因此反应时间就有一些延迟。Apache 服务器使用了一个特殊技术解决这个问题，这就是预先生成多个空余的子进程驻留在系统中，一旦有请求出现，就立即使用这些空余的子进程进行处理，这样就不存在生成子进程造成的延迟了。在运行中随着客户请求的增多，启动的子进程会随之增多，但这些服务器副本在处理完一次 HTTP 请求之后并不立即退出，而是停留在计算机中等待下次请求。但是空余的子进程副本不能只增加不减少，太多的空余子进程没有处理任务，也占用服务器的处理能力，因此就要限制空余副本的数量，使其保持一个

合适的范围，这样既能及时回应客户请求，又能减少不必要的进程数量。

因此就可以使用 MinSpareServers 来设置最少的空余子进程数量，以及使用 MaxSpareServers 来限制最多的空闲子进程数量，多余的服务器进程副本就会退出。根据服务器的实际情况来进行设置，如果服务器性能较高，并且也被频繁访问，就应该增大这两个参数的设置。对于高负载的专业网站，这两个值应该大致相同，并且等同于系统支持的最多服务器副本数量，也减少不必要的副本退出。

3、最大连接请求

MaxRequestWorkers 150

在另一方面，服务器的能力毕竟是有限的，不可能同时处理无限多的连接请求，因此 MaxRequestWorkers 就用于规定服务器支持的最多并发访问的客户数，如果这个值设置得过大，系统在繁忙时不得不在过多的进程之间进行切换，这样就会减慢对每个客户的反应，并降低了整体的效率。如果这个值设置的较小，那么系统繁忙时就会拒绝一些客户的连接请求，并将其放入请求队列。当服务器性能较高时，就可以适当增加这个值的设置。对于专业网站，应该使用提高服务器效率的策略，因此这个参数不能超过硬件本身的限制，如果频繁出现拒绝访问现象，就说明需要升级服务器硬件了。对于非专业网站，不太在意对客户浏览器的反应速度，或者认为反应速度较慢也比拒绝连接好，就也可以略微超过硬件条件来设置这个参数。

4、子进程副本最大处理次数

MaxConnectionsPerChild 0

使用子进程的方式提供服务的 Web 服务器，常用的方式是一个子进程为一次连接服务，这样造成的问题就是每次连接都需要进行生成和退出子进程的系统操作，使得这些额外的处理过程占据了计算机的大量处理能力。因此最好的方式是一个子进程可以为多次连接请求服务，这样就不需要这些生成和退出进程的系统消耗，Apache 就采用了这样的方式，一次连接结束后，子进程并不退出，而是停留在系统中等待下一次服务请求，这样就极大的提高了性能。

但由于在处理过程中子进程要不断的申请和释放内存，次数多了就会造成一些内存垃圾，从而影响系统的稳定性，浪费不必要的系统资源。因此在一个副本处理过一定次数的请求之后，就可以让这个子进程副本退出，再从原始的 apache2 进程中重新复制一个干净的副本，这样就能提高系统的稳定性。每个子进程处理服务请求次数由 MaxConnectionsPerChild 定义。缺省的设置值为 0，对于具备高稳定性特点的 Linux 系统来讲可以设置为 1000 甚至更高，设置为 0 支持每个副本进行无限次的服务处理。

8.2.1.3.9 /etc/apache2/mods-available/autoindex.conf

1、索引和图标

```
    IndexOptions FancyIndexing VersionSort HTMLTable NameWidth=* DescriptionWidth=*
Charset=UTF-8
    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
    AddIconByType (VID,/icons/movie.gif) video/*
    AddIcon /icons/binary.gif .bin .exe
    AddIcon /icons/binhex.gif .hqx AddIcon /icons/tar.gif .tar
    AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
    AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
    AddIcon /icons/a.gif .ps .ai .eps
    AddIcon /icons/layout.gif .html .shtml .htm .pdf
    AddIcon /icons/text.gif .txt
    AddIcon /icons/c.gif .c
    AddIcon /icons/p.gif .pl .py
    AddIcon /icons/f.gif .for
    AddIcon /icons/dvi.gif .dvi
    AddIcon /icons/uuencoded.gif .uu
    AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
    AddIcon /icons/tex.gif .tex
    AddIcon /icons/bomb.gif core
    AddIcon /icons/back.gif ..
    AddIcon /icons/hand.right.gif README
    AddIcon /icons/folder.gif ^^DIRECTORY^^
    AddIcon /icons/blank.gif ^^BLANKICON^^
    DefaultIcon /icons/unknown.gif
    #AddDescription "GZIP compressed document" .gz
    #AddDescription "tar archive" .tar
    #AddDescription "GZIP compressed tar archive" .tgz
    ReadmeName README.html HeaderName HEADER.html
    IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

当一个 HTTP 请求的 URL 为一个目录的时候，服务器返回这个目录中的索引文件。但如果一个目录中不存在缺省的索引文件，并且该服务器又许可显示目录文件列表的时候，就会显示出这个目录中的文件列表，为了使得这个文件列表能具有可理解性，而不仅仅是一个简单的列表，就需要前面的这些设置参数。

如果使用了 IndexOptions FancyIndexing 选项，可以让服务器产生的目录列表中针对各种不同类型的文档引用各种图标。而哪种文件使用哪种图标，则使用下面的 AddIconByEncoding，AddIconByType 以及 AddIcon 来定义，分别依据 MIME 的编码，类型以及文件的后缀来判断使用何种图标。如果不能确定文档使用的图标，就使用 DefaultIcon 定义的缺省图标。

同样，使用 AddDescription 可以为不同类型的文档加入不同的描述。服务器还在目录下，查询使用 ReadmeName 和 HeaderName 定义的文件（自动加上.html 后缀，如果没有发现，再使用.txt 后缀进行搜索），如果发现了这些文件，就在文件列表之前首先显示这些文件的内容，以使得普通目录列表具备更大的可理解性。

IndexIgnore 让服务器在列出文件列表时忽略相应的文件，这里使用模式配置的方式定义文件名。

8.2.1.3.10 /etc/apache2/mods-available/userdir.conf

1、多用户主页设置

```
UserDir public_html
```

当在一台 Linux 上运行 Apache 服务器时，这台计算机上的所有用户都可以有自己的网页路径，形如 http://example.org.cn/~user，使用波浪符号加上用户名就可以映射到用户自己的网页目录上。映射目录为用户个人主目录下的一个子目录，其名字就用 UseDir 这个参数进行定义，缺省为 public_html。如果不想为正式的用户提供网页服务，使用 DISABLED 作 UserDir 的参数即可。

```
<Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS>
        Order allow, deny
        Allow from all
    </Limit>
    <LimitExcept GET POST OPTIONS>
        Order deny, allow
        Deny from all
    </LimitExcept>
</Directory>
```

这里可以看到 Directory 的另一个用法，即可以通过简单的模式匹配方法，针对分布在不同目录下的子目录定义访问控制权限。这样设置需要 Apache 服务器对每个路径进行额外的处理，因此就会降低服务器的性能，所以缺省情况并没有打开这种访问限制。

这里可以看到另外一个语句 Limit，Limit 语句就是用来针对具体的请求方法来设定访问控制的，其中可以使用 GET，POST 等各种服务器支持的请求方法做 Limit 的参数，设定不同请求方法的访问限制。一般可以打开对 GET，POST，HEAD 三种请求方法，而屏蔽其它的请求方法，以增加安全性。Limit 语句中，可以用 Order、Allow、Deny，Allow 和 Deny 中可以使用匹配的方法针对域名和 IP 进行限制，只是对于域名是从后向前匹配，对于 IP 地址则从前向后匹配。

8.2.1.3.11 /etc/apache2/mods-available/status.conf

1、运行状态

ExtendedStatus On

Apache 服务器可以通过特殊的 HTTP 请求报告自身的运行状态，打开这个 ExtendedStatus 可以让服务器报告更全面的运行状态信息。

8.2.1.3.12 /etc/apache2/conf-available/security.conf：

1、描述信息

ServerTokens Full

这个参数控制了服务器回应给客户端的“Server:”应答头是否包含关于服务器操作系统类型和编译进的模块描述信息。

- ServerTokens Prod[uctOnly]
  服务器会发送(比如)：Server: Apache
- ServerTokens Major
  服务器会发送(比如)：Server: Apache/2
- ServerTokens Minor
  服务器会发送(比如)：Server: Apache/2.0
- ServerTokens Min[imal]
  服务器会发送(比如)：Server: Apache/2.0.41
- –ServerTokens OS
  服务器会发送(比如)：Server: Apache/2.0.41 (Unix)
- ServerTokens Full（或未指定）
  服务器会发送(比如)：Server: Apache/2.0.41 (Unix) PHP/4.3.2 MyMod/1.2 此设置将作用于整个服务器，而且不能用在虚拟主机的配置段中。这个参数还控制着 ServerSignature 的显示内容。

2、错误文档

ServerSignature On

一些情况下，例如当客户请求的网页并不存在时，服务器将产生错误文档，缺省情况下由于打开了 ServerSignature 选项，错误文档的最后一行将包含服务器的名字，Apache 的版本等信息。有的管理员更倾向于不对外显示这些信息，就可以将这个参数设置为 Off，或者设置为 Email，最后一行将替换为对 ServerAdmin 的 Email 提示。

8.2.1.3.13 /etc/apache2/mods-available/proxy.conf

1、浏览器代理设置

```
#<IfModule mod_proxy.c>
    #    ProxyRequests On
    #    <Proxy>
        #    Allow from .your_domain.com
    #    </Proxy>
    #    <Directory proxy:*>
        #    Order deny，allow
        #    Deny from all
        #    Allow from .your_domain.com
    #    </Directory>
    # Enable/disable the handling of HTTP/1.1 "Via:" headers.
    # ("Full" adds the server version;
    # "Block" removes all outgoing Via: headers)
    # Set to one of: Off | On | Full | Block
    #    ProxyVia On
    # To enable the cache as well，edit and uncomment the following lines:
    #（no cacheing without CacheRoot）
    #    CacheRoot "/var/www/cache"
    #    CacheSize 5
    #    CacheGcInterval 4
    #    CacheMaxExpire 24
    #    CacheLastModifiedFactor 0.1
    #    CacheDefaultExpire 1
    #    NoCache a_domain.com another_domain.edu
        joes.garage_sale.com
#</IfModule>
```

Apache 服务器本身就具备代理的功能，然而这要求加载入 mod_proxy 模块。这能使用 IfModule 语句进行判断，如果存在 mod_proxy 模块，就使用 ProxyRequests 打开代理支持。此后的 Directory 用于设置对 Proxy 功能的访问权限设置，以及用于设置缓冲的各个参数设置。

8.2.1.4 日志

Apache 内建了记录服务活动的功能，这就是它的日志。

- /var/log/apache2/access_log

  访问日志，记录了所有对 Web 服务器的访问活动。

  访问日志文件的位置实际上是一个配置选项。在/etc/apache2/apache.conf 配置文件中，有如下这行内容：

  CustomLog "/var/log/apache2/access_log" common

  CustomLog 指定了保存日志文件的具体位置以及日志的格式。上面这行指定的是 common 日志格式。

  CustomLog 中的路径是日志文件的路径。注意，由于日志文件是由 HTTP 用户打开的

（用 User 指定），因此必须注意这个路径要有安全保证，防止该文件被随意改写。

- /var/log/apache2/error_log

错误日志，记录了 Apache 发生的错误。错误日志和访问日志一样也是 Apache 服务的标准日志。

错误日志无论在格式上还是在内容上都和访问日志不同。然而，错误日志和访问日志一样也提供丰富的信息，我们可以利用这些信息分析服务器的运行情况，哪里出现了问题。

错误日志的位置可以通过/etc/apache2/httpd.conf 文件中的 ErrorLog 选项进行设置：

```
ErrorLog "/var/log/apache2/error.log"
```

除非文件位置用“/”开头，否则这个文件位置是相对于 ServerRoot 目录的相对路径。

错误日志记录了服务器运行期间遇到的各种错误，以及一些普通的诊断信息，比如服务器何时启动，何时关闭等。

我们可以设置日志文件记录信息级别的高低，控制日志文件记录信息的数量和类型。这是通过 LogLevel 设置的，该参数默认设置的级别是 error，即记录称得上错误的事件。

大多数情况下，我们在日志文件中见到的内容分属两类：文档错误和 CGI 错误。但是，错误日志中偶尔也会出现配置错误，另外还有前面提到的服务器启动和关闭信息。

1、解读日志

下面是访问日志中一个典型的记录：

```
216.35.116.91 - - [19/Aug/2000:14:47:37 -0400] "GET / HTTP/1.0" 200 654
```

这行内容由 7 项构成，上面的例子中有两项空白，但整行内容仍旧分成了 7 项。

第一项信息是远程主机的地址，即它表明访问网站的究竟是谁。在上面的例子中，访问网站的主机是 216.35.116.91。随便说一句，这个地址属于一台名为 si3001.inktomi.com 机器（要找出这个信息，可以使用 nslookup 工具查找 DNS），inktomi.com 是一家制作 Web 搜索软件的公司。可以看出，仅仅从日志记录的第一项出发，我们就可以得到有关访问者的不少信息。

默认情况下，第一项信息只是远程主机的 IP 地址，但我们可以要求 Apache 查出所有的主机名字，并在日志文件中用主机名字来替代 IP 地址。然而，这种做法通常不值得推荐，因为它将极大地影响服务器记录日志的速度，从而也就减低了整个网站的效率。另外，有许多工具能够将日志文件中的 IP 地址转换成主机名字，因此要求 Apache 记录主机名字替代 IP 地址是得不偿失的。

然而，如果确实有必要让 Apache 找出远程主机的名字，那么我们可以使用如下命令：

```
HostNameLookups on
```

如果 HostNameLookups 设置成 double 而不是 on，日志记录程序将对它找到的主机名字进行反向查找，验证该主机名字确实指向了原来出现的 IP 地址。默认情况下 HostNameLookups 设置为 off。

上例日志记录中的第二项是空白，用一个占位符“-”替代。实际上绝大多数时候这一项都是如此。这个位置用于记录浏览者的标识，这不只是浏览者的登录名字，而是浏览者的 email 地址或者其它唯一标识符。这个信息由 identd 返回，或者直接由浏览 器返回。

很早的时候，那时 Netscape 0.9 还占据着统治地位，这个位置往往记录着浏览者的 email 地址。然而，由于有人用它来收集邮件地址和发送垃圾邮件，所以它未能保留很长时间，很久之前市场上几乎所有的浏览器就取消了这项功能。因此，到了今天，我们在日志记录的第二项看到 email 地址的机会已经微乎其微了。

日志记录的第三项也是空白。这个位置用于记录浏览者进行身份验证时提供的名字。当然，如果网站的某些内容要求用户进行身份验证，那么这项信息是不会空白的。但是，对于大多数网站来说，日志文件的大多数记录中这一项仍旧是空白的。

日志记录的第四项是请求的时间。这个信息用方括号包围，而且采用所谓的“公共日志格式”或“标准英文格式”。因此，上例日志记录表示请求的时间是 2000 年 8 月 19 日星期三 14:47:37。时间信息最后的“-0400”表示服务器所处时区位于 UTC 之前的 4 小时。

日志记录的第五项信息或许是整个日志记录中最有用的信息，它告诉我们服务器收到的是一个什么样的请求。该项信息的典型格式是“METHOD RESOURCE PROTOCOL”，即“方法 资源 协议”。

在上例中，METHOD 是 GET，其它经常可能出现的 METHOD 还有 POST 和 HEAD。此外还有不少可能出现的合法 METHOD，但主要就是这三种。

RESOURCE 是指浏览者向服务器请求的文档，或 URL。在 这个例子中，浏览者请求的是“/”， 即网站的主页或根。 大多数情况下，“/”指向 DocumentRoot 目录的 index.html 文档，但根据服务器配置的不同它也可能指向其它文件。

PROTOCOL 通常是 HTTP，后面再加上版本号。版本号或者是 1.0，或者是 1.1，但出现 1.0 的时候比较多。我们知道，HTTP 协议是 Web 得以工作的基础，HTTP/1.0 是 HTTP 协议的早期版本，而 1.1 是最近的版本。当前大多数 Web 客户程序仍使用 1.0 版本的 HTTP 协议。

日志记录的第六项信息是状态代码。它告诉我们请求是否成功，或者遇到了什么样的错误。大多数时候，这项值是 200，它表示服务器已经成功地响应浏览器的请求，一切正常。一般地说，以 2 开头的状态代码表示成功，以 3 开头的状态代码表示由于各种不同的原因用户请求被重定向到了其它位置，以 4 开头的状态代码表示客户端存在某种错误，以 5 开头的状态代码表示服务器遇到了某个错误。

日志记录的第七项表示发送给客户端的总字节数。它告诉我们传输被打断（即，该数值是否和文件的大小相同）。把日志记录中的这些值加起来就可以得知服务器在一天，一周或者一月内发送了多少数据。

2、CGI 错误

错误日志最主要的用途可能是诊断行为异常的 CGI 程序。为了进一步分析和处理方便，CGI 程序输出到 STDERR（Standard，标准错误设备）的所有内容都将直接进入错误日志。这意味着，任何编写良好的 CGI 程序，如果出现了问题，错误日志就会告诉我们有关问题的详细信息。

然而，把 CGI 程序错误输出到错误日志也有它的缺点，错误日志中将出现许多没有标准格式的内容，这使得用错误日志自动分析程序从中分析出有用的信息变得相当困难。

由于 CGI 程序运行环境的特殊性，如果没有错误日志的帮助，大多数 CGI 程序的错误都将很难解决。

有不少人在邮件列表或者新闻组中抱怨说自己有一个 CGI 程序，当打开网页时服务器却返回错误，比如“Internal Server Error”。我们可以肯定，这些人还没有看过服务器的错误日志，或者根本不知道错误日志的存在。决多大多数情况下，错误日志能够精确地指出 CGI 错误的所在以及如何修正这个错误。

3、定义日志格式

定制日志文件的格式涉及到两个参数，即 LogFormat 和 CustomLog，默认 apache.conf 文件提供了关于这两个参数的几个示例。

LogFormat 定义格式并为格式指定一个名字，以后我们就可以直接引用这个名字。CustomLog 设置日志文件，并指明日志文件所用的格式（通常通过格式的名字）。

LogFormat 的功能是定义日志格式并为它指定一个名字。例如，在默认的 apache.conf 文件中，我们可以找到下面这行代码：

LogFormat “%h %l %u %t \”%r\“ %>s %b” common

该命令创建了一种名为“common”的日志格式，日志的格式在双引号包围的内容中指定。格式字符串中的每一个变量代表着一项特定的信息，这些信息按照格式串规定的次序写入到日志文件。

Apache 文档已经给出了所有可用于格式串的变量及其含义，下面是其译文：

%...a：远程 IP 地址

%...A：本地 IP 地址

%...B：已发送的字节数，不包含 HTTP 头

%...b：CLF 格式的已发送字节数量，不包含 HTTP 头。例如当没有发送数据时，写入“-”而不是 0。

%...{FOOBAR}e：环境变量 FOOBAR 的内容

%...f：文件名字

%...h：远程主机

%...H：请求的协议

%...{Foobar}i：Foobar 的内容，发送给服务器的请求的标头行

%...l：远程登录名字（自 identd，如提供的话）

%...m：请求的方法

%...{Foobar}n：来自另外一个模块的注解朐 Foobar 胙的内容

%...{Foobar}o：Foobar 的内容，应答的标头行

%...p：服务器响应请求时使用的端口

%...P：响应请求的子进程 ID

%...q：查询字符串（如果存在查询字符串，则包含“？”后面的部分；否则它是一个空字符串）

%...r：请求的第一行

%...s：状态，对于进行内部重定向的请求，这是指“原来”请求的状态。如果

用%...>s，则是指后来的请求

%...t：以公共日志时间格式表示的时间（称为标准英文格式）

%...{format}t：以指定格式 format 表示的时间

%...T：为响应请求而耗费的时间，以秒计

%...u：远程用户（如果返回状态（%s）401 则可能是伪造的）

%...U：用户所请求的 URL 路径

%...v：响应请求的服务器的 ServerName

%...V：依照 UseCanonicalName 设置得到的服务器名字

在所有上面列出的变量中，“...”示一个可选的条件。如果没有指定条件，则变量的值将以“-”取代。分析前面来自默认 apache.conf 文件的 LogFormat 示例，可以看出它创建了一种名为“common”的日志格式，其中包括：远程主机，远程登录名字，远程用户，请求时间，请求的第一行代码，请求状态，以及发送的字节数。

有时候我们只想在日志中记录某些特定的，已定义的信息，这时就要用到“...”。如果在“%”和变量之间放入了一个或者多个 HTTP 状态代码，则只有当请求返回的状态代码属于指定的状态代码之一时，变量所代表的内容才会被记录。例如，如果我们想要记录的是网站的所有无效链接，那么可以使用：

```
LogFormat %404{Referer}i BrokenLinks
```

反之，如果我们想要记录那些状态代码不等于指定值的请求，只需加入一个“!”号即可：

```
LogFormat %!200U SomethingWrong
```

8.2.1.5 参考信息

有关 Apache 的详细信息，请参见位于 http://www.apache.org 的官方主页。

8.2.2 Samba

8.2.2.1 简介

Samba 是一组软件包，运行于 Linux 系统中，用来实现与 Windows 系统之间的通信。它允许 Linux 系统出现在的 Windows 网络邻居中，使 Windows 用户享受由 Linux 主机提供的文件与打印服务。

Samba 由两个主要程序组成，它们是 smbd 和 nmbd。这两个守护进程在服务器启动到停止期间持续运行，Samba 提供了四个服务：文件和打印服务、授权与被授权、名字解析、浏览服务。前两项服务由 smbd 提供，后两项服务则由 nmbd 提供。简单地说，smbd 进程的作用是处理到来的 SMB 软件包，为使用该软件包的资源与 Linux 进行协商，nmbd 进程使主机（或工作站）能浏览 Linux 服务器。

8.2.2.2 配置文件

Samba 的配置文件 smb.conf 位于/etc/samba/目录下，可以设置各种参数，设置哪些资源被共享、以及其它用户对这些资源的操作权限等等。

8.2.2.3 配置文件参数

smb.conf 文件的格式非常容易理解，可以对它进行编辑与修改使它适合自己的需求。

1、三个特殊段

除了控制访问共享的定制外，smb.conf 有三个特殊段:

- [global]段定义了整个过程中的全局参数，并为其它段提供缺省值。
- [homes]段定义远程用户存取在本地 Linux 机的用户主目录的参数。
- [printers]段定义远程用户共享本地 Linux 的打印机。

下面就配置文件中比较重要的几个段分别给与说明。

1）[global]

[global]节是全局参数节，该部分提供了全局参数，对 Samba 的功能具有很大的影响，主要用来设置整个系统规则，其设置直接影响 Samba 系统的行为。

- netbios name
  设置主机名称，即 Windows 系统“网上邻居”中所见的机器名。
- workgroup
  用来指定主机所在网络上所属的域名或者工作组名称。格式是:
  workgroup= Nt Domain-Name or workgroup-name
- server string
  用来设置本机描述，说明服务器用途，缺省是 Samba Server。
- host allow
  允许访问的主机 IP 地址，对安全非常重要。它设置了哪些机器可以访问 Samba 服务器。
- load printers
  允许自动加载打印机列表，而不需要单独设置每一台打印机。
- Interface
  配置 Samba 使用多个网络接口。
- domain controller
  仅当网络中有主域控制器时使用此选项。
- Security
  设置安全参数，定义安全模式。Samba 的安全模式有三种:
  share user server
  其中 share 安全级别最低，user 模式要求连接时输入用户名和口令，server 模式要求用户的认证由 Samba 服务器或 NT 服务器来完成。
- encrypt passwords
  smb passwd file
  设置是否需要加密口令。
  例样:

```
[global]
netbios name=Samba test workgroup = in.linx
smb passwd file = /etc/samba/private/smbpasswd
server string = Samba Server
hosts allow = 192.168.0. 192.168.0. 127.
```

```
dns proxy = no security = user
encrypt passwords = yes
guest account = testuser
interfaces = 192.168.0.1/24
max log size = 50
socket options = TCP_NODELAY SO_RCVBUF=8192
SO_SNDBUF=8192
log file = /var/log/samba/log.%m
load printers = yes
printcap name = /etc/printcap
... ...
```

2）[homes]

[homes]部分指定了 Windows 共享的主目录，如果在 Windows 工作站登录的名字与 Linux 用户名相同，提供的口令也一致，那么打开网络邻居，双击共享目录图标，就可获得访问该目录的权力。从 Windows 访问 Linux 主目录时，用户名作为主目录共享名。该部分通常定义了 Linux 机器上共享的目录资源，其名字可以由用户确定。段中的设置控制了每一个用户目录的共享权限。

当任何一个客户访问 Samba 服务器时，在网络资源中都能出现自己的 home 目录共享。其配置如下:

- Comment
  设定在浏览本机资源时，出现在指定资源旁边的字符串。
- Browseable
  控制一项服务是否能够出现在网上邻居中，no 意味着这个目录将在浏览时显示为要验证的用户名称，yes 则显示 homes 和要验证的用户名称的共享目录。
- Writable
  控制是否允许通过验证的用户对主目录有写入的权限，但最终取决于该目录的权限。设置 writable = yes 后，主目录是可写的。

例样:

```
[homes]
comment = Home Directories
browseable = no
writable = no
```

3）[printers]

[printers]部分用于指定如何共享 Linux 网络打印机，从 windows 系统访问 Linux 网络打印机时，共享应是 printcap 中指定的 Linux 打印机名。其配置如下:

- Comment
  指定为哪些设备进行设置。
- Browseable
  指定是否能够浏览 Windows 客户机的 printcap 文件定义的全部打印机。
- Printable

指定是否允许不打印而去访问与打印服务相关的假脱机目录，例如：/var/spool/lpd/lp。

- Public
  指定未被 Linux 用户名验证的 Windows 客户机是否有权使用 Samba 打印服务。
- writable
  指定是否有权写假脱机目录。

例样:

```
[printers]
comment = All Printers
browseable = no
printable = yes
public = no
writable = no
```

2、其它配置方法

1）用户共享目录[public]

设置公共访问目录。

[public]段提供了所有用户都可以共同访问的目录。除了那些属于维护人员具有读、写、执行权外，用户只具有读取的访问权限。其配置如下:

- Path
  指定公众共享目录路径。
- Public
  取值为 yes 时，允许公众共享，否则，禁止公众共享该目录。
- Writable
  取值为 yes 时，公众对/home/samba 有可写权力。
- Printable
  取值为 yes 时，公众对/home/samba 有打印权力。
- write list
  指定具有可写权力的用户名单。

例样:

```
[public]
path = /home/samba
public = yes
writable = yes
printable = no
writelist = @staff
```

2）用户映射

全局参数“username map”用来控制用户映射，它允许管理员指定一个映射文件，该文件包含了在客户机和服务器之间进行用户映射的信息。

如：username map = /etc/samba/smbuser

用户映射经常在 windows 和 linux 主机间进行。两个系统拥有不同的用户账

号，用户映射的目的是将不同的用户映射成为一个用户，便于共享文件。

下面是一个映射文件的例子:

```
# Map Windows admin to sysadmin
sysadmin = admin administrator
# Map the member of developer to studio
studio = @developer
```

等号左边是单独的 Linux 账号，等号右边是要映射的账号列表。

服务器逐行分析映射文件，如果提供的账号和某行有右侧列表中的账号匹配，就把它替换为等号左边的账号。

3）使用加密口令

新版本的 Windows 95 以及 Windows 98、Windows NT（sp3 以上版本），在网络传输中仅传递加密口令作为用户认证的信息。这类客户机和不支持加密口令并且以 user 安全级运行的 Samba 服务器通讯时，会出现故障。为了正常的通讯，Samba 服务器使用加密口令。下面讨论如何在 Samba 中使用加密口令。

i. 口令文件 /etc/samba/private/smbpasswd

为了使用加密口令，Samba 需要一份口令文件，并且该文件应该和 Linux 的口令文件（/etc/passwd）保持同步。下面是生成文件命令:

```
$ less /etc/password | mksmbpasswd >
                        /etc/samba/private/smbpasswd
```

smbpasswd 是需要的口令文件，其权限是 0600，所有者是 sysadmin。smbpasswd 和 passwd 文件的记录对应，密码部分不同。密码有两部分组成，每部分是 32 个“X”，前部分用于和 Lanman 通讯，后部分和 Windows NT 通讯。sysadmin 用户可以使用 smbpasswd 命令为每个用户设定 Samba 口令。

ii. 修改配置文件 /etc/samba/smb.conf

要使 Samba 使用加密口令，需要在配置文件 smb.conf 中加入如下参数。

```
Encrypt passwords = yes
Smb passwd file = /etc/samba/private/smbpasswd
```

第一行通知 Samba 使用加密口令，第二行给出口令文件的位置。

4）Windows 系统中的明码口令

Samba 系统中使用明码口令作为连接 SMB 的默认设置。当 SMB 服务器对协商协议做出响应时，响应信息包含了一位，以说明服务器是否支持询问或者响应加密。随着 Win95 的网络重定向更新程序的发布，Microsoft 修改了默认值，这样，Windows 客户就不会向不支持加密的服务器发送明码口令了。

在这种情况下，有两种解决办法:

i. 设置 Samba 服务器使用加密口令

ii. 让 Windows 客户使用明码口令

这里选用第 2 种解决办法，通过修改注册表来实现。下面对 Win95/

Win98、Winnt 用户分别给与说明。

i. Wni98/Win95 系统用户

在注册表中加入下列注册字，并重新启动机器:
[HKLM\System\CurrentCntrolSet\Services\VxD\VNETSUP]
"EnablePlainTextPAssword" = dword:00000001

ii. Winnt 系统用户

修改注册表，加入下列注册表项，并重新启动机器:
[HKLM\System\CurrentCntrolSet\Services\Rdr\Parameters]
"EnablePlainTextPAssword" = dword:00000001

5）配置 Samba 实现本地打印机共享

在开始之前，确信 Samba 运行正常，并且 Windows 用户可以在网络邻居的列表中可以看到共享的 Samba 服务器。

为了配置 smb.conf 文件实现共享一个本地打印机，检查 Linux 下的打印机是否工作正常。下一步在 Windows 客户端安装同一个打印机。这样做的话，需要 Windows 下打印机的安装盘。开始安装打印机，点击开始，选择打印机，并且双击添加打印机。

在打印机安装向导中，按照屏幕上的说明一步一步进行操作，直到在列表中选择打印机的名称。要点：写出打印机的准确名称，注意大小写（名称对大小写是敏感的）。如果需要，在安装完成后要重启计算机。

然后在 Samba 服务器上以系统管理员（sysadmin）身份登录，对 Samba 配置文件 /etc/samba/smb.conf 进行相应修改。

在[global] 这一节，找到打印名称这行，如果想使 Windows 用户可以使用所有本地打印机，去掉这一行，并且如果文件中没有这些内容的话，添加上:

```
printcap name = /etc/printcap
load printers = yes
```

如果只想让其它用户用一个打印机，不要删掉那行，也不用添加什么。需要注意的是在大多数从 BSD 发展而来的系统上，Samba 假设使用默认的打印系统。如果用的是其它打印系统，在打印系统配置中，找到相应的行，改变默认设置（bsd）。选择包括 sysv、plp、lprng、aix、hpux 和 qnx。如果不能确认正在使 用的打印系统，可以查看一下相关文件，但大多数情况下都是默认的。

下一步，使所有的本地打印机都实现共享。所有的打印机都在/etc/printcap 文件的打印机列表中，在/etc/samba/smb.conf 文件尾添加以 下服务条目:

```
[printers]
writable = no
path = /tmp
printable = yes
```

如果只要共享一个打印机，在 smb.conf 文件的[global]小节添加一个自动服务行。和在/etc/printcap 中的相同，这行指定想使用的打印机名称。下行中在 lp 填写默认打印机的名称:

```
auto services = lp
```

而且，仿照下面创建一个打印机定义，添加到[services]小节中（在/etc/samba/smb.conf 文件末尾）：

```
[lp]
printable = yes
comment = Epson Stylus（Color）
public = yes
writable = no
browseable = yes
printer driver= Epson Stylus Color 740
```

定义中的打印机名是 lp，这个名称必须和上面自动服务行中的名称相一致，并且也要和在/etc/printcap 文件中定义的打印机名称相同（或打印机的别名）。需要注意的是，打印机驱动行需要正确的输入 Windows 中的打印机信息（在这里，Epson Stylus Color 740，还是需要注意大小写）。

重新启动 Samba 服务后就可以共享使用打印机了。

8.2.2.4 日志

Samba 日志文件可以帮助诊断大多数的管理员在初期可能面对的问题。Samba 在形成日志的时候非常灵活，可以随意设置服务器记录日志的大小。

Samba 日志文件位于/var/log/samba/，其中 log.nmbd 和 log.smbd 日志文件记录了 Samba 服务的开启、关闭以及出错信息。log.nmbd 记录了 nmbd 进程的信息，而 log.smbd 记录的是 smbd 进程的信息。另外，Samba 可以为每一个连接的客户使用唯一的配置和日志文件，并使用这个文件跟踪故障。只要某一个用户出现了问题，就可以利用查看某一个用户的 Samba 配置文件来进行正确的配置。

8.2.2.5 参考信息

有关 Samba 的详细信息，请参见位于 http://www.samba.org/官方主页。

8.2.3 SSH

8.2.3.1 简介

SSH（Secure Shell）是目前较可靠，专为远程登录会话和其它网络服务提供安全性的协议。利用 SSH 协议可以有效防止远程管理过程中的信息泄露问题。用户通过 SSH 可以把所有传输的数据进行加密，使“中间人”的攻击方式不可能实现，而且也能够防止 DNS 和 IP 欺骗。它还有一个额外的好处是传输的数据是经过压缩的，可以加快传输的速度。SSH 作用广泛，既可以代替 Telnet，又可以为 FTP、POP，甚至为 PPP 提供一个安全的“通道”，主要是解决口令在网上明文传输的问题。为了系统安全和用户自身的权益，推广 SSH 是必要的。

8.2.3.2 配置文件

SSH 服务器的配置使用的是/etc/ssh/sshd_config。

8.2.3.3 配置文件参数

1、基本设置

编辑“sshd_config”文件（vi /etc/ssh/sshd_config），加入或改变下面的参数:

```
# This is ssh server systemwide configuration file.
Port   22
ListenAddress 192.168.1.1
HostKey   /etc/ssh/ssh_host_key
ServerKeyBits 1024
LoginGraceTime    600
KeyRegenerationInterval     3600
PermitRootLogin   no
IgnoreRhosts   yes
IgnoreUserKnownHosts      yes
StrictModes          yes
X11Forwarding      no
PrintMotd yes
SyslogFacility AUTH
LogLevel  INFO
RhostsAuthentication    no
RhostsRSAAuthentication          no
RSAAuthentication      yes
PasswordAuthentication      yes
PermitEmptyPasswords      no
AllowUsers     admin
```

下面将逐行说明上面的选项设置:

- Port

  设置 sshd 监听的端口号。

- ListenAddress

  设置 sshd 服务器绑定的 IP 地址。

- HostKey

  设置包含计算机私人密匙的文件。

- ServerKeyBits

  定义服务器密匙的位数。

- LoginGraceTime

  设置如果用户不能成功登录，在切断连接之前服务器需要等待的时间（以秒为单位）。

- KeyRegenerationInterval

  设置在多少秒之后自动重新生成服务器的密匙（如果使用密匙）。重新生成密匙是为了防止用盗用的密匙解密被截获的信息。

- PermitRootLogin

  设置 root 能不能用 SSH 登录。这个选项一定不要设成“yes”。

- IgnoreRhosts
  设置验证的时候是否使用“rhosts”和“shosts”文件。
- IgnoreUserKnownHosts
  设置 sshd 是否在进行 RhostsRSAAuthentication 安全验证的时候忽略用户的“$HOME/.ssh/known_hosts”。
- StrictModes
  设置 sshd 在接收登录请求之前是否检查用户家目录和 rhosts 文件的权限和所有权。这通常是必要的，因为新手经常会把自己的目录和文件设成任何人都有写权限。
- X11Forwarding
  设置是否允许 X11 转发。
- PrintMotd
  设置 sshd 是否在用户登录的时候显示/etc/motd 中的信息。
- SyslogFacility
  设置在记录来自 sshd 的消息的时候，是否给出“facility code”。
- LogLevel
  设置记录 sshd 日志消息的层次。INFO 是一个好的选择。查看 sshd 的 man 帮助页，已获取更多的信息。
- RhostsAuthentication
  设置只用 rhosts 或“/etc/hosts.equiv”进行安全验证是否已经足够了。
- RhostsRSAAuthentication
  设置是否允许用 rhosts 或“/etc/hosts.equiv”加上 RSA 进行安全验证。
- RSAAuthentication
  设置是否允许只有 RSA 安全验证。
- PasswordAuthentication
  设置是否允许口令验证。
- PermitEmptyPasswords
  设置是否允许用口令为空的帐号登录。
- AllowUsers
  “AllowUsers”的后面可以跟着任意的数量的用户名的匹配串（patterns）或 user@host 这样的匹配串，这些字符串用空格隔开。主机名可以是 DNS 名或 IP 地址。

在默认的设置选项中需要注意的有:

- PermitRootLogin
  最好把这个选项设置成“PermitRootLogin without-password”，这样“root”用户就不能从没有密匙的计算机上登录。把这个选项设置成“no”将禁止“root”用户登录，只能用“su”命令从普通用户转成“root”。
- X11Forwarding
  把这个选项设置成“yes”允许用户运行远程主机上的 X 程序。就算禁止这个选项也不能提高服务器的安全因为用户可以安装他们自己的转发器（forwarder），

请参看“man sshd”。

- PasswordAuthentication

  把这个选项设置为“no”只允许用户用基于密匙的 mail。为 POP 加上“加密通道”可以防止 POP 的密码被网络监听器（sniffer）监听到。还有一个好处就是 SSH 的压缩方式可以让邮件传输得更快。

2、配置客户端的软件

OpenSSH 有三种配置方式：命令行参数、用户配置文件和系统级的配置文件/etc/ssh/ssh_config。命令行参数优先于配置文件，用户配置文件优先于系统配置文件。所有的命令行的参数都能在配置文件中设置。因为缺省设置时没有默认的用户配置文件，所以要把朐/etc/ssh/ssh_config 拷贝并重新命名为朐~/.ssh/config。

标准的配置文件：

```
[lots of explanations and possible options listed]
# Be paranoid by default
Host        *
ForwardAgent no
ForwardX11    no
FallBackToRsh      no
```

还有很多选项的设置可以用“man ssh”查看“CONFIGURATION FILES”这一章，配置文件是按顺序读取的。先设置的选项先生效。

假定你在 www.foobar.com 上有一个名为“bilbo”帐号。而且你要“ssh-agent”和“ssh-add”结合起来使用并且使用数据压缩来加快传输速度。 因为主机名太长了，不愿意得输入这么长的名字，用“fbc”作为“www.foobar.com”的简称。你的配置文件可以是这样的：

```
Host   *fbc
HostName      www.foobar.com
User   bilbo ForwardAgent        yes
Compression        yes
# Be paranoid by default
Host   *
ForwardAgent no
ForwardX11         no
FallBackToRsh            no
```

输入“ssh fbc”之后，SSH 会自动地从配置文件中找到主机的全名，用你的用户名登录并且用“ssh-agent”管理的密匙进行安全验证。

用 SSh 连接到其它远程计算机用的还是“paranoid（偏执）”默认设置。如果有些选项没有在配置文件或命令行中设置，那么还是使用默认的“paranoid”设置。

在我们上面举的那个例子中，对于到 www.foobar.com 的 SSH 连接：“ForwardAgent”和“Compression”被设置为“Yes”；其它的设置选项（如果没有用命令

行参数）“ForwardX11”和“FallBackToRsh”都被设置成“No”。

3、其它

还有一些需要注意的设置选项:

- CheckHostIP

  这个选项用来进行 IP 地址的检查以防止 DNS 欺骗。

- CompressionLevel

  压缩的级别从“1”（最快）到“9”（压缩率最高）。默认值为“6”。

- ForwardX11

  为了在本地运行远程的 X 程序必须设置这个选项。

- LogLevel

  当 SSH 出现问题的时候，即可设置该选项。默认值为“INFO”。

8.3 bonding

Linux bonding 驱动提供了一种可以将多个网络接口设备捆绑为一个逻辑“bonded”接口的方法，用于网络负载均衡及网络冗余。

- 网络负载均衡

  bonding 的网络负载均衡是在文件服务器中常用到的，可以把多块网卡当一块来用。解决一个 IP 地址，流量过大，服务器网络压力过大的问题。bonding 是在有限资源的情况下实现网络负载均衡的最好方法。

- 网络冗余

  对于服务器来说，网络设备的稳定也是很重要的，特别是网卡。在生产型的系统中，大多通过硬件设备的冗余来提供服务器的可靠性和安全性，比如电源。bonding 也能为网卡提供冗余支持。把多块网卡绑定到一个 IP 地址，当一块网卡发生物理性损坏时，另一块网卡自动启动，并提供正常服务，即：默认情况下只有一块网卡工作，其他网卡做备份。

8.3.1 参数说明

8.3.1.1 参数列表

1、primay

指定哪个 slave 来成为主设备（primay device），取值为字符串，如 eth0，eth1 等。只要指定的设备可用，它将一直是激活的 slave。只有在主设备（primay device）断线时才会切换设备。这在希望某个 slave 设备优先使用的情形下很有用，比如，某个 slave 设备有更高的吞吐率。

> **注意**
>
> primay 选项只对 active-backup 模式有效。

2、updelay

指定当发现一个链路恢复时，在激活该链路之前的等待时间，以毫秒计算。该选项只对 miimon 链路侦听有效。updelay 应该是 miimon 值的整数倍，如果不是，它将会被向下取整到最近的整数。缺省值为 0。

3、arp_interval

指定 ARP 链路监控频率，单位是毫秒(ms)。如果 ARP 监控工作于以太兼容模式（模式 0 和模式 2）下，需要把 switch(交换机)配置为在所有链路上均匀的分发网络包。如果 switch(交换机)被配置为以 XOR 方式分发网络包，所有来自 ARP 目标的应答将会被同一个链路上的其他设备收到，这将会导致其他设备的失败。ARP 监控不应该和 miimon 同时使用。设定为 0 将禁止 ARP 监控。缺省值为 0。

4、arp_ip_target

指定一组 IP 地址用于 ARP 监控的目标，它只在 arp_interval>0 时有效。这些 IP 地址是 ARP 请求发送的目标，用于判定到目标地址的链路是否工作正常。该设定值为 ddd.ddd.ddd.ddd 格式。多个 IP 地址通过逗号分隔。至少指定一个 IP 地址。最多可以指定 16 个 IP 地址。缺省值是没有 IP 地址。

5、downdelay

指定一个时间，用于在发现链路故障后，等待一段时间然后禁止一个 slave 来，单位是毫秒(ms)。该选项只对 miimon 监控有效。downdelay 值应该是 miimon 值的整数倍， 否则它将会被取整到最接近的整数倍。缺省值为 0。

6、lacp_rate

指定在 802.3ad 模式下，我们希望的链接对端传输 LACPDU 包的速率。可能的选项:

- Slow 或者 0

  请求对端每 30s 传输 LACPDU
- Fast 或者 1

  请求对端每 1s 传输 LACPDU
- 缺省值是 slow

7、max_bonds

为 bonding 驱动指定创建 bonding 设备的数量。比如：如果 bonds 为 3，而且 bonding 驱动还没有加载，那么 bond0，bond1，bond2 将会被创建。缺省值为 1。

8、Miimon

指定 MII 链路监控频率，单位是毫秒(ms)。这将决定驱动检查每个 slave 链路状态频率。

0 表示禁止 MII 链路监控。100 可以作为一个很好的初始参考值。下面的 use_carrier 选项将会影响如果检测链路状态。更多的信息可以参考“高可靠性”章节。缺省值为 0。

9、mode

指定 bonding 的模式。缺省是 6 (balance-alb)Adaptive load balancing（适配器适应性负载均衡）。可选的 mode 包括：0，1，2，3，4，5，6。

8.3.2 bonding 模式

Linux 操作系统下双网卡绑定有七种模式（由 mode 来配置）。

- Mode=0

  即：(balance-rr)（平衡抡循环模式）

  特点：此模式需要 switch(交换机)支持及设定才能发挥效果，其特点是需要传输数据包顺序是依次传输（即：第 1 个包走 eth0，下一个包就走 eth1...一直循环下去，直到最后一个传输完毕），此模式提供负载平衡和容错能力；但是我们知道如果一个连接或者会话的数据包从不同的接口发出的话，中途再经过不同的链路，在客户端很有可能会出现数据包无序到达的问题，而无序到达的数据包需要重新要求被发送，这

样网络的吞吐量就会下降。

- Mode=1

  即：(active-backup)Active-backup policy（主-备份模式）

  特点：只有一个设备处于活动状态，当一个宕掉另一个马上由备份转换为主设备。mac 地址是外部可见得，从外面看来，bond 的 MAC 地址是唯一的，以避免 switch(交换机)发生混乱。此模式只提供了容错能力；由此可见此算法的优点是可以提供高网络连接的可用性，但是它的资源利用率较低，只有一个接口处于工作状态， 在有 N 个网络接口的情况下，资源利用率为 1/N。

- Mode=2

  即：(balance-xor)XOR policy（平衡模式）

  特点：基于指定的传输 HASH 策略传输数据包。缺省的策略是：(源 MAC 地址 XOR 目标 MAC 地址) % slave 数量。其他的传输策略可以通过 xmit_hash_policy 选项指定，此模式提供负载平衡和容错能力。

- Mode=3

  即：broadcast（广播模式）

  特点：在每个 slave 接口上传输每个数据包，此模式提供了容错能力。

- Mode=4

  即：(802.3ad)IEEE 802.3ad Dynamic link aggregation （IEEE 802.3ad 动态链接聚合模式）

  特点：创建一个聚合组，它们共享同样的速率和双工设定。根据 802.3ad 规范将多个 slave 工作在同一个激活的聚合体下。

  外出流量的 slave 选举是基于传输 hash 策略，该策略可以通过 xmit_hash_policy 选项从缺省的 XOR 策略改变到其他策略。需要注意的是，并不是所有的传输策略都是 802.3ad 适应的，尤其考虑到在 802.3ad 标准 43.2.4 章节提及的包乱序问题。不同的实现可能会有不同的适应性。

  必要条件:

  - 条件 1：ethtool 支持获取每个 slave 的速率和双工设定
  - 条件 2：switch(交换机)支持 IEEE 802.3ad Dynamic link aggregation
  - 条件 3：大多数 switch(交换机)需要经过特定配置才能支持 802.3ad 模式

- Mode=5

  即：(balance-tlb)Adaptive transmit load balancing（适配器传输负载均衡模式）

  特点：不需要任何特别的 switch(交换机)支持的通道 bonding。在每个 slave 上根据当前的负载（根据速度计算）分配外出流量。如果正在接受数据的 slave 出故障了，另一个 slave 接管失败的 slave 的 MAC 地址。

  该模式的必要条件：ethtool 支持获取每个 slave 的速率。

- Mode=6

  即：(balance-alb) Adaptive load balancing（适配器适应性负载均衡模式）

  特点：该模式包含了 balance-alb 模式，同时加上针对 IPV4 流量的接收负载均衡(receive load balance,rlb)，而且不需要任何 switch(交换机)的支持。接收负载均衡是通

过 ARP 协商实现的。bonding 驱动截获本机发送的 ARP 应答，并把源硬件地址改写为 bond 中某个 slave 的唯一硬件地址，从而使得不同的对端使用不同的硬件地址进行通信。

8.3.3 配置 bonding

8.3.3.1 操作系统及环境

操作系统为：凝思安全操作系统 V6.0.100（系统中默认安装了 ifenslave）。需启用两块或两块以上网卡设备。

1、查看内核支持：（示例）

root@Linx:~# modprobe -l bond*
kernel/drivers/net/bonding/bonding.ko
root@Linx:~# modinfo bonding
filename: /lib/modules/2.6.32-5-linx-amd64/kernel/drivers/net/
bonding/bonding.ko
author: Thomas Davis, tadavis@lbl.gov and many others
description: Ethernet Channel Bonding Driver, v3.5.0
version: 3.5.0
license: GPL
srcversion: C0EFCD8CB4AC214A8146EC2
depends:
vermagic: 2.6.32-5-linx-amd64 SMP mod_unload modversions
parm: max_bonds:Max number of bonded devices (int)
parm: num_grat_arp:Number of gratuitous ARP packets to send on failover event (int)
parm: num_unsol_na:Number of unsolicited IPv6 Neighbor
Advertisements packets to send on failover event (int)
parm: miimon:Link check interval in milliseconds (int)
parm: updelay:Delay before considering link up, in milliseconds (int)
parm: downdelay:Delay before considering link down, in milliseconds (int)
parm: use_carrier:Use netif_carrier_ok (vs MII ioctls) in miimon; 0 for off, 1 for on
(default) (int)
parm: mode:Mode of operation : 0 for balance-rr, 1 for active-backup, 2 for balance-xor,
3 for broadcast, 4 for 802.3ad, 5 for balance-tlb, 6 for balance-alb (charp)
parm: primary:Primary network device to use (charp)
parm: lacp_rate:LACPDU tx rate to request from 802.3ad partner (slow/fast) (charp)
parm: ad_select:803.ad aggregation selection logic: stable (0, default), bandwidth (1),
count (2) (charp)
parm: xmit_hash_policy:XOR hashing method: 0 for layer 2 (default), 1 for layer 3+4
(charp)

parm: fail_over_mac:For active-backup, do not set all slaves to the same MAC. None (default), active or follow (charp)

2、加载模块

加载模块实现分为手动加载和开机启动自动加载。

- 手动加载

先卸载 bonding 模块。

root@Linx:~# rmmod bonding

示例选模式为 6；允许最大绑定数为 1，加载 binding。

root@Linx:~# modprobe bonding mode=6 max_bonds=1

注意：手动加载 bonding 需重启网络，即：修改完配置文件/etc/network/interfaces 后重启网络。

root@Linx:~# /etc/init.d/networking restart

- 开机启动自动加载

修改配置文件/etc/network/interfaces 如 8.3.3.2 中所示：配置文件中添加了 bonding 的相关配置，开机重启可自动加载 bonding 模块。

8.3.3.2 配置文件

编辑网络接口配置文件，指定 IP。

root@Linx:~# vim /etc/network/interfaces

查看示例文件:

```
root@Linx:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5). auto bond0
iface bond0 inet static
address 172.16.0.73
network 255.255.255.0
netmask 255.255.255.0
gateway 172.16.0.254
slaves eth0 eth1
bond_mode active-backup
bond_miimon 100
bond_downdelay 200
bond_updelay 200
root@Linx:~#
```

查看网络信息:

```
root@Linx:~# ifconfig
```

第 9 章 日志与审计

9.1 日志管理

9.1.1 查看系统日志

以审计管理员登录，运行 less /var/log/kern.log，查看日志文件，可以看到系统核心输出的日志信息。

审计管理员可以查看/var/log/audit/及其下面的文件。

9.1.2 查看用户信息

用户的标识与鉴别、用户帐号的创建、删除、禁止或使能都是可审计的，每个审计事件会记录特定的信息。

除了审计日志，审计管理员还可以查看文件/var/log/auth.log，应能看到用户登录（包括日期，时间，用户身份，登录终端信息以及是否成功等信息）、添加用户、删除用户 等等信息。

提示

口令数据（明码或密码形式）永远不会记录在审计日志中，审计文件 /var/log/auth.log 中也不会有用户口令数据。

9.2 安全审计

审计系统是凝思安全操作系统安全体系的重要组成部分，安全审计能够对文件、目录、系统资源、系统调用进行监控和记录。管理员通过创建完善的监控和审计规则，使违反规则的行为被审计和记录，以便进行安全追踪和定位。

本章主要介绍凝思安全操作系统审计子系统，因此将会涉及到与审计管理员（audadmin）相关的概念，读者可以参考分权管理员章节了解审计管理员的相关概念。

在凝思安全操作系统中，审计系统结构如下图所示:

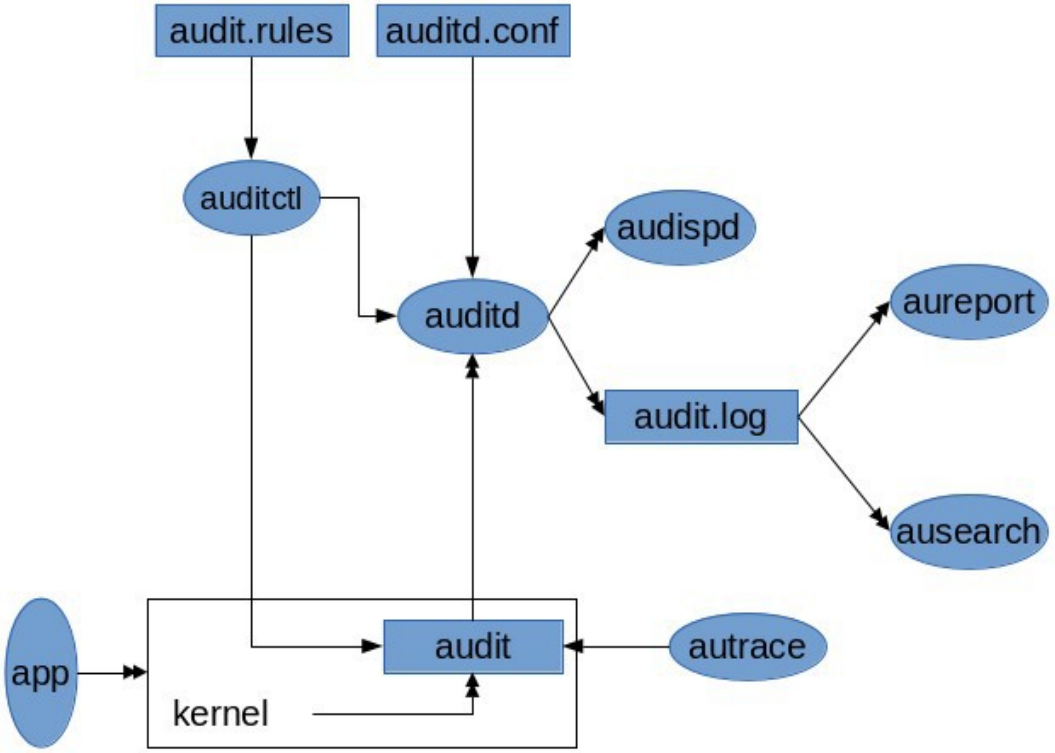


图 9. 1 审计系统结构图

图中双箭头表示审计数据的流向，单箭头表示审计控制的流向。

审计系统组成项目表如下:

表 9. 1 审计系统核心组成

序号	组成项	项目说明
1	auditd	auditd 守护进程负责把内核审计接口传送来的信息写入到磁盘，该守护进程的运行环境配置文件是/etc/default/auditd，运行功能配置文件是/etc/audit/auditd.conf。
2	auditctl	auditctl 程序用来控制审计系统生成何种审计信息。
3	audit rules	/etc/audit/audit.rules 文件内容实际上是一系列 auditctl 命令内容，当操作系统启动后 auditd 启动完成时，该文件的内容通过 auditctl -R 被设置

4	aureport	aureport 程序能够对审计日志进行各种统计报告，方便管理员查看，或者把报告内容转交给其它程序做进一步处理。
5	ausearch	ausearch 程序允许管理员通过关键字或者其它存在于审计日志中的特征字符进行审计日志检索。
6	audispd	audispd 守护进程可以把审计信息进行分发，当 auditd 把审计信息传送给 audispd 的时候，审计信息将不会记入审计日志。
7	autrace	程序可以提供类似于 strace 的功能，用于对一个进程进行追踪并把信息写入审计日志

9.2.1 审计守护进程

在凝思操作系统中该守护进程 auditd 负责把审计信息写入磁盘审计日志。该守护进程有环境配置和功能配置两个配置文件。

9.2.1.1 环境配置文件

环境配置文件的路径为/etc/default/auditd，在凝思安全操作系统中环境变量 AUDIT_LANG 的默认配置为 none，AUDITD_CLEAN_STOP 的默认配置为 yes。

9.2.1.2 功能配置文件

功能配置文件的路径为/etc/audit/auditd.conf，在凝思安全操作系统中该文件的默认配置如下:

log_file = /var/log/audit/audit.log
log_format = RAW
log_group = audadmin
priority_boost = 4
flush = INCREMENTAL
freq = 20
num_logs = 8
disp_qos = lossy
dispatcher = /sbin/audispd
name_format = NONE
##name = mydomain
max_log_file = 300
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
action_mail_acct = audadmin
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
##tcp_listen_port =
tcp_listen_queue = 5
##tcp_client_ports = 1024-65535
tcp_client_max_idle = 0
enable_krb5 = no

功能配置文件参数说明如下表（注意 SINGLE 和 HALT 参数请勿设置）：

表 9. 2 审计功能配置文件参数表

序号	组成项	项目说明
1	log_file	审计日志的完整路径，为了操作系统安全请尽量采用默认配置。
2	Log_format	指日志存放格式，有两个选项 RAW、NOLOG。RAW 表示按照内核传输

		的审计信息记录。NOLOG 表示不记录日志，但该模型并不会影响审计 audispd 分发。
3	Log_group	凝思安全操作系统默认为 audadmin
4	Priority_boost	启动优先级，可取值为 0~4，默认值为 4。
5	flush	可选的值为 NONE、INCREMENTAL、DATA、SYNC。 NONE 表示正常写。 INCREMENTAL 当设置此项目时候，freq 项目必须设置，表示有多少条记录写一次。 DATA 表示磁盘日志保持同步。 SYNC 表示每次写磁盘日志的时候保持数据和元数据同步。
6	frep	配合 flush 项一起使用。
7	Num_logs	当 max_log_file_action 设置为 ROTATE 时表示要保存的审计日志文件数目。必须是 0~99 之间的数，如果设置小于 2 表示不循环日志。
8	Disp_qos	审计调度程序和审计守护进程之间的通信类型，有效值为 lossy 和 lossless。lossy 表示当审计调度程序和审计守护进程之间的缓存区满后，发送给调度程序的信息会被丢弃，但只要 log_format 为 raw，审计信息还是会通过审计守护进程写入磁盘。lossless 表示再发送审计到调度程序前和把审计信息写入到磁盘前，调度程序会等待缓存区有足够空间。
9	dispatcher	审计调度程序，审计信息可以通过该程序分发出去。
10	Name_format	用于插入何种计算机名称信息到审计事件中，可取 NONE、HOSTNAME、FQD、NUMERIC、USER 这几项值。 NONE 表示不插入信息。 HOSTNAME 表示插入系统调用

		gethostname 返回的值。 FQD 表示插入经 DNS 解析的主机名。 NUMERIC 类似 FQD 除了对 IP 的解析。 USER 指 name 项中所填的内容。 默认值为 NONE。
11	name	是配合 name_format 选项的，当 name_format 设置时，本选项用于定义管理员。
12	Max_log_file	表示最大日志文件容量。单位为兆字节，当到达设定容量值时，会执行 max_log_file_action 设置的行为。
13	Max_log_file_action	当到达 max_log_file 设定的容量时要采取的行为。可选值有 IGNORE、SYSLOG、SUSPEND、ROTATE 和 KEEP_LOGS。 IGNORE 表示不采取任何行为。 SYSLOG 表示发送警告到系统日志。 SUSPEND 表示不再向审计日志写入审计信息。 ROTATE 表示循环审计日志，但要保存的老的文件数目由 num_logs 确定，老的审计日志名为 audit.log.N，其中 N 越大所对应的审计信息越老。 KEEP_LOGS 表示会循环审计日志，但会忽略 num_logs 参数，不会删除审计日志。
14	Space_left	当硬盘剩余空间多少 MB 的时候触发 space_left_action 设定的动作。
15	Space_left_action	可选的值为 IGNORE、SYSLOG、EMAIL、SUSPEND、SINGLE、HALT。 IGNORE 表示不进行任何操作。 SYSLOG 表示发送警告到系统日志。 EMAIL 表示发送邮件到 action_mail_acct 设置的邮箱。 SUSPEND 表示停止往硬盘写日志，但审计守护进程保持活动状态。

		SINGLE 表示切换到单用户模式（请勿设置此选项）。 HALT 表示会立刻关机（请勿设置此选项）
16	Action_mail_acct	用于设置维护审计日志的管理员的邮件地址，如果地址没有主机名，则默认采取本地地址，需安装 sendmail。
17	Admin_space_left	这项和 space_left 类似，但是应该设置比 space_left 更小的值，当达到这个值时，意味着系统磁盘即将耗尽，这应当被视为磁盘耗尽前可以采取措施的最后机会。磁盘剩余空间多少 MB 的时候触发 admin_space_left_action 设定的动作。（特别注意，该项设置的值必须小于 space_left 的值，否则会导致审计守护进程无法启动）
18	Admin_space_left_action	可选的值为 IGNORE、SYSLOG、EMAIL、EXEC、SUSPEND、SINGLE、HALT。 IGNORE 表示不进行任何操作。 SYSLOG 表示发送警告到系统日志。 EMAIL 表示发送邮件到 action_mail_acct 设置的邮箱。 EXEC 表示 FIXME。 SUSPEND 表示停止往硬盘写日志，但审计守护进程保持活动状态。 SINGLE 表示进入单用户模式（请勿设置此选项）。 HALT 表示会立刻关机（请勿设置此选项）
19	Disk_full_action	如果审计文件分区已满，则采取此行为。可选值有 IGNORE、SYSLOG、SUSPEND、SINGLE（请勿设置此选项）、HALT（请勿设置此选项）。 以上选项的涵义同上。
20	Disk_error_action	在写入日志或循环日志文件时检测到错误

		要采取的行为。可选值为 IGNORE、SYSLOG、SUSPEND、SINGLE（请勿设置此选项）、HALT（请勿设置此选项）。 以上选项的涵义同上。
21	Tcp_listen_port	接收远程审计日志的端口，取值范围为 1~65535
22	Tcp_listen_queue	接收远程审计日志的队列长度
23	Tcp_client_ports	审计日志客户端端口
24	Tcp_client_max_idle	这个参数代表一个客户端就绪所需要的时间
25	Enable_keb5	kerberos5 认证和加密，默认为 no，需要启用时选 yes。
26	Krb5_principal	本项主要用于服务端的 kerberos5 认证和加密
27	Krb5_key_file	客户端主要的 kerberos5 认证和加密 key 文件路径

9.2.2 审计规则配置程序

在凝思安全操作系统中，审计规则设置程序 auditctl 用于控制审计系统产生何种审计信息。

9.2.2.1 参数详解

审计规则程序参数说明如下表：

表 9. 3 审计规则程序参数表

序号	选项	选项描述
1	-b	设置审计日志缓存大小，默认为 64，当缓存占满时，-f 参数的设置被触发。
2	-e	用于启动或者关闭审计系统：0 表示关闭；1 表示开启；2 表示不允许修改配置，在此模式下任何对审计系统的配置都不被允许，并且会被记录到审计日志。
3	-f	配合其它选项的错误标记：0 表示什么都不做；1 表示打印警告信息到系统日志；2 表示不进行数据同步而立刻关机（请勿设置此选项）。
4	-h	命令的帮助信息。
5	-i	表示从文件读取审计规则时，忽略错误。
6	-l	显示所有规则，每行一条规则。
7	-k	设置过滤关键字，关键字最长可以达到 31 字节，方便过滤审计

		日志。
8	-m	给审计系统发消息。
9	-r	设置每秒传输的审计消息数限制，默认值为 0，表示无限制，当速率超过设定值时，-f 参数的设置被触发。
10	-R	从文件读取所有规则，后跟规则文件名。
11	-s	输出审计系统状态信息。
12	-v	显示版本。
13	-a	在规则列表中追加一项规则。后面跟两个参数，具体参数及用法见下面附表。
14	-A	在规则列表最前面插入一项规则。
15	-d	从规则列表删除规则。
16	-D	删除规则列表的所有规则。
17	-F	建立规则时候用到，后面跟操作、操作符（=、!=、<、>、<=、>=、&、&=）以及对应的取值。详情见下面附表
18	-S	建立规则时候用到，后面跟系统调用的名称或编号。
19	-q	参数后跟挂载点和要挂载目录，中间用逗号隔开。如果先监控一个目录，后在监控目录下挂载另外一个目录，后挂载的目录则不会被监控。如果要使后挂载的目录一旦被挂载就能和挂载点所在目录一样被监控，就需要用本参数进行设置，这样一旦新目录挂载到已经被监控的目录，也会被监控。
20	-w	设置监视路径，注意不要设置根目录，路径也不支持通配符。
21	-W	移除监视路径。
22	-p	对路径进行监视的选项，一共可选 rwxa 四个中的一个或者多个，r 表示监视路径的读行为，w 表示监视路径的写行为，x 表示监视路径的执行行为，a 表示监视路径的属性改动行为。当不设置该选项时默认设置为 rwxa。

审计规则程序参数表-a 参数说明附表如下：

表 9. 4 规则程序参数表-a 参数说明附表

追加的规则种类	task	追加到每个任务列表
	entry	追加到系统调用入口列表
	exit	追加到系统调用退出列表
	user	追加到用户信息过滤列表
	exclude	追加到不想看到的事件的列表
是否写入日志	never	不记录审计日志
	always	写入审计日志
规则种类和是否写入日志之前用逗号隔开		

审计规则程序参数表-F 参数说明附表如下：

表 9. 5 规则程序参数表-F 参数说明附表

a0,a1,a2,a3	系统调用的前四个参数
arch	系统调用的 cpu 架构
auid	用户注册时分配的用户 id
b32	32 位系统调用表的 arch
b64	64 位系统调用表的 arch
devmajor	主设备号
devminor	次设备号
egid	有效组 id
euid	有效用户 id
exit	系统调用的退出值
fsgid	文件系统组 id
gid	组 id
inode	节点号
key	设置过滤关键字，和-k 用法一样
msgtype	用于匹配消息类型，仅可以配合-a 的 exclude
path	监视文件的路径，仅可以配合-a 的 exit 使用
perm	参见-p，仅可以配合 0a 的 exit 使用
pers	操作系统个人序列号
pid	进程 id
ppid	父进程 id
sgid	设置组 id
success	如果退出值大于等于 0，这个值为真；否则为假。当写一条规则时，使用 1 的时候为真，0 的时候为假
suid	设置用户 id
uid	用户 id

9.2.2.2 用法举例

1、要添加对路径/home/test/的监控，具体监控读（r）、执行（x）、属性变更（a），并且设置 key 为 LOG_test 方便对日志进行检索，具体设置和移除命令如下：

设置命令：auditctl -w /home/test/ -p rxa -k LOG_test
移除命令：auditctl -W /home/test/ -p rxa -k LOG_test

2、要添加对 64 位 open 系统调用进行审计，在 open 调用入口进行审计，key 设置为 LOG_open，具体设置和移除命令如下：

设置命令：auditctl -a entry,always -F arch=b64 -S open -k LOG_open
移除命令：auditctl -d entry,always -F arch=b64 -S open -k LOG_open

3、要添加对/etc/crontab 的监控，具体监控写（w）、属性变更（a），并且设置 key 为 CFG_crontab 方便对日志进行检索，具体设置和移除命令如下:

设置命令：auditctl -w /etc/crontab -p wa -k CFG_crontab
移除命令：auditctl -W /etc/crontab -p wa -k CFG_crontab

4、要添加对/etc/passwd 的监控，具体监控写（w）、属性变更（a），并且设置 key 为 CFG_passwd 方便对日志进行检索，具体设置和移除命令如下:

设置命令：auditctl -w /etc/passwd -p wa -k CFG_passwd
移除命令：auditctl -W /etc/passwd -p wa -k CFG_passwd

5、要添加对挂载和解除挂载的审计，在 mount 和 umount2 调用入口进行审计，具体设置和移除命令如下机

设置命令：auditctl -a entry,always -F arch=b64 -S mount -S umount2
移除命令：auditctl -d entry,always -F arch=b64 -S mount -S umount2

6、通过文件批量设置审计监控规则，例如要设置以下 5 条命令:

```
auditctl -w /home/test/ -p rxa -k LOG_test
auditctl -a entry,always -F arch=b64 -S open -k LOG_open
auditctl -w /etc/crontab -p wa -k CFG_crontab
auditctl -w /etc/group -p wa -k CFG_group
auditctl -a entry,always -F arch=b64 -S mount -S umount2
```

可以以审计管理员（audadmin）分别执行上述五条命令，也可以按如下方式一次性设置审计规则。

将以下内容保存至一个文本文件，并命名为 auditrulefile

```
-w /home/test/ -p rxa key=LOG_test
-a entry,always -F arch=b64 -S open -k LOG_open
-w /etc/crontab -p wa -k CFG_crontab
-w /etc/group -p wa -k CFG_group
-a entry,always -F arch=b64 -S mount -S umount2
```

系统中/etc/audit/audit.rules 文件就是一个批量设置审计规则和监视的文件，由于审计服务开启时/etc/audit/audit.rules 会自动通过 auditctl -R 进行设置，从而保持了设置在/etc/audit/audit.rules 文件中的规则在开机时自动生效。

9.2.2.3 注意事项

通过 auditctl 命令直接设置的规则和监控路径在审计守护进程重启后就会失效，如果要设置开机自动生效的规则和监控路径，请通过 audit.rules 规则配置文件进行设置，一旦审计守护进程启动，audit.rules 文件的规则就会被读取设置。

9.2.3 审计规则配置文件

该文件在审计守护进程启动时通过被 auditctl 读取设置。凝思操作系统提供了默认规 则，用户可根据 auditctl 程序的参数自行添加修改规则，但一些危险的参数需要谨慎设置。

9.2.4 审计日志分析工具

在凝思安全操作系统中默认提供 aureport 和 ausearch 两个程序用于分析审计日志。

9.2.4.1 报表生成工具

aureport 程序可以用来从审计日志文件生成汇总报表。

可以直接使用 aureport 生成总的汇总报表，也可以通过查看 help 信息，使用参数导出某一项的汇总报表。

9.2.4.2 日志搜索工具

ausearch 程序可以用来基于一些条件搜索审计日志。这些条件可以是审计的事件 ID，UID，GID，系统调用名称等，也可以是规则中设置的方便查找日志的 key。

9.2.5 审计分发程序

在凝思安全操作系统中使用默认的 audispd 作为分发程序。

9.2.6 审计系统附加功能

在凝思安全操作系统中提供 autrace 附加功能，autrace 程序提供类似于 strace 功能用于跟踪一个进程，并且可以将跟踪信息写入到审计日志中，方便进一步分析。

9.2.7 审计系统设置

在凝思安全操作系统中，应按照以下流程对审计系统进行设置:

1、service auditd stop，使用管理员停止审计守护进程 auditd。

2、配置 auditd 守护进程，修改 auditd 的配置文件/etc/default/auditd 和/etc/audit/auditd.conf。

3、设置要审计的内容，编辑审计规则和观察器配置文件/etc/audit/audit.rules。

4、service auditd start，结束配置启动审计守护进程 auditd。

5、分析审计日志。

第 10 章 凝思安全机制

凝思安全操作系统 V6.0.100 提供了多种强制性安全保护机制，包括强制访问控制（Mandatory Access Control，简称 MAC）、强制行为控制（Mandatory Behavior Control，简称 MBC）、强制能力控制（Mandatory Capability Control，简称 MCC）。

强制访问控制（MAC）主要利用 BIBA 完整性模型，BLP 机密性保护模型，保护敏感数据的访问。详情查看 10.2 章节。

强制行为控制（MBC）主要利用程序的路径特征，限制程序对文件的访问。详情可查看 10.4 章节。

强制能力控制（MCC）主要利用 POISX1003.e 中提出的能力，针对传统 UNIX 模型进行优化，将特权细分成不同的能力。从而可以避免因挟持特权进程而产生的安全威胁。详情参阅 10.5 章节。

10.1 安全模块

凝思安全操作系统的核心实现是通过安全模块来完成的，系统中所有的资源请求都会受到安全模块的监控和限制。

10.1.1 安全模块的参数

对于普通用户不需要了解内核安全模块的实现原理，但为了方便用户的使用内核安全模块提供了几个功能参数供用户使用，通过这些参数可以调整安全安全模块的行为。

表 10. 1 内核安全模块参数表

序号	参数名称	描述	默认值
1	Linx_netlbl_doi	DOI 是网络标签中网络域的标识，凝思安全操作系统的默认值是 12，该网络域标识可以根据客户的要求修改，需要通信的主机 应保持 DOI 一致。	12
2	Linx_type	当安全模块发现出现错误或异常事件时安 全模块的工作方式。linx_type=permissive，打印应该被拒绝的访问的审计信息而不真正拒绝访问，这种工作方式主要是用于调试任务。 linx_type=quiet，拒绝访问但是不打印审 计信息。这种方式是正常的工作方式。 linx_type=enforcing，既打印审计信息又拒绝访问操作。这种方式也是正常工作方式但带有调试信息。	quiet

3	Linx_netlbl_enable	网络标签功能开关。 0 为不使能。 1 为使能	0
4	Linx_mac_inherited	MAC 标签生成方式选择。 linx_mac_inherited=2 时：被生成客体的 MAC 标签是由，生成客体的主体与客体所在目录两者的 MAC 标签的最大值决定的。空域代表最小本如果两者都是空就取空。 linx_mac_inherited=1 时：被生成客体的 MAC 标签和所在目录的标签保持一致。 linx_mac_inherited=0 时：客体不携带 MAC 标签。	1
5	Linx_mac_rules_reload	运行时重新加载 MAC 规则。 0 为不允许 secadmin 重新加载。 1 为允许 secadmin 重新加载。	0

10.1.2 安全模块的加载与卸载

安全模块默认情况下是开机自动加载的。如果用户需要确认系统中是否加载安全模块，可是使用以下命令：

```
lsmod | grep linx_sec
```

如果需要卸载安全模块可以使用以下命令：

```
modprobe -r linx_sec
```

如果需要重新加载安全模块可以使用以下命令：

```
modprobe linx_sec
```

注意

一旦安全模块被卸载，安全模块提供的功能将不生效。

下面将逐一介绍各安全机制。

10.2 强制访问控制（Mandatory Access Control）

为了可以更好的理解本章中的术语及其概念需要在阅读之前熟悉下列概念：

名词	描述
主体（subject）	主体就是引起信息在两个客体间流动的任意活动实体。在凝思安全操作系统中，主体几乎总是代表用户活跃在某一进程中的一个线程。
客体（object）	客体或系统客体是一种实体，信息随主体的导向在客体内部流动。客体包括目录、文件、显示器、键盘、存储器、磁存储器、打印机及其它数据存储术转移设备。基本上，客体就是指数据容器或系统资源。对客体的访问实际上意味着对数据的访问。在凝思安全操作系统中，客体指目录和文件。
完整性（integrity）	作为一个关键概念，完整性是数据可信性的一种程度。若数据的完整性提高，则数据的可信性相应提高。
敏感性（sensitivity）	通常在讨论多级安全模型（MLS）时使用。敏感性程度曾被用来描述数据应该有何等的重要或机密。若敏感性程度增加，则保密的重要性或数据的机密性相应增强。
标签（label）	标签是一种可应用于文件、目录或系统其他客体的安全属性，它也可以被认为是一种机密性印鉴。当一个文件被施以标签时，其标签会描述这一文件的安全参数，并只允许拥有相似安全性设置的文件、用户、资源等访问该文件。标签值的含义及解释取决于相应的策略配置：某些策略会将标签当作对某一客体的完整性和保密性的表述，而其它一些策略则会用标签保存访问规则。
策略（policy）	一套用以规定如何达成目标的规则。策略一般用以描述如何对特定客体进行操作。凝思安全操作系统提供了可以由管理员自由配置规则的机制。

强制访问控制是凝思安全操作系统强制性安全保护最重要的组成部分，它分为本地强制访问控制和网络环境强制访问控制（网络标签）两部分。

系统中文件的访问控制由两部分组成：一个部分是自主访问控制（DAC）、另一部分是强制访问控制（MAC）。

自主访问控制（DAC）是在确认主体身份及所属组的基础上，根据访问者的身份和授权来决定访问模式，对访问进行限定的一种控制策略。所谓自主，是指具有被授予某种访问权力的用户能够自己决定是否将访问控制权限的一部分授予其他用户或从其他用户那里收回他所授予的访问权限。使用这种控制方法，用户或应用可任意在系统中规定谁可以访问它们的资源，这样，用户或用户进程就可有选择地与其他用户共享资源。它是一种对单独用户执行访问控制的过程和措施。

强制访问控制是“强加”给访问主体的，即系统强制主体服从访问控制策略。强制访问控

制用于将系统中的信息分密级和类进行管理，以保证每个用户只能访问到那些被标明允许其访问的信息的一种访问约束机制。在强制访问控制下，所有主体（例如：进程）和客体（例如：文件、段、设备）都被指定敏感标记（如安全级别、访问权限等），系统通过比较主体和客体的敏感标记来决定一个主体是否能够访问某个客体。

凝思安全操作系统强制访问控制机制使用 BLP+Biba 模型作为基础。

BLP 模型是在 1973 年由 D.Bell 和 J.LaPadula 在《Mathematical founda-ons and model》提出并加以完善，它根据军方的安全政策设计，解决的本质题是对具有密级划分信息的访问控制，是第一个比较完整地形式化方法对系统安全进行严格证明的数学模型，被广泛应用于描述计算机系统的安全问题。BLP 一开始作为军方的一个安全模型出台，对于数据间的权利转让而产生变化的访问权限，提供一系列安全检查，避免权利的过度转让产生的模糊泛滥。

BLP 模型是一个形式化模型，使用数学语言对系统的安全性质进行描述，BLP 模型也是一个状态机模型，它反映了多级安全策略的安全特性和状态转换规则。

BLP 模型定义了系统、系统状态、状态间的转换规则，安全概念、制定了一组安全特性，对系统状态、状态转换规则进行约束，如果它的初始状态是安全的，经过一系列规则都是保持安全的，那么可以证明该系统是安全的。

BLP 模型的基本安全策略是“下读上写”，即主体对客体向下读、向上写。主体可以读安全级别比他低或相等的客体，可以写安全级别比他高或相等的客体。“下读上写”的安全策略保证了数据库中的所有数据只能按照安全级别从低到高的流向流动，从而保证了敏感数据不泄露。

整个 BLP 模型的保护原则就是信息流动应受到制约，可获取信息的范围受到保护后，就可以对信息的泄漏起到防御的作用。

Biba 模型或 Biba 完整性模型由 Kenneth J.Biba 于 1977 年开发的，是计算机安全策略，这一策略用来确保数据的完整性。主体和客体被放置到一个有序的完整性集合中。一般来说，模型的开发是为了规避在 BLP 模型只涉及数据保密性的弱点。

强制访问控制一般与自主访问控制结合使用，并且实施一些附加的、更强的访问限制。一个主体只有通过了自主与强制性访问限制检查后，才能访问某个客体。用户可以利用自主访问控制来防范其它用户对自己客体的攻击，由于用户不能直接改变强制访问控制属性，所以强制访问控制提供了一个不可逾越的、更强的安全保护层以防止其它用户偶然或故意地滥用自主访问控制。

10.2.1 功能简介

MAC 机制是基于进程与文件之间的敏感标签的计算来授权访问动作的。而标签的计算可分为标签本身和计算方法（规则）两部分。

在凝思安全操作系统中，规则是可以通过规则配置文件进行配置的。主体往往是进程，进程的 MAC 标签附着在 PCB（进程控制块）中。客体通常是文件、目录，客体的标签是通过 EA 数据存储的。

10.2.1.1 安全标签

凝思安全操作系统 V6.0.100 中，每个主体和客体都有一个标签，每个标签有三个域，主客体标签的格式是一致的，标签中的每个域是 64 位二进制的无符号整数它们分别命名为 f1、f2、f3，故每个域的取值范围为[0,2^64 – 1]。在配置使用标签的过程中，标签的 3 个域不一定都被使用。可以使用其中任意一个或多个域。

10.2.1.2 主体标签

主体通常是进程为代表的动态实体，所以无法事先打标签，所以将主体的标签与 UID 绑定在一起，即当进程的 UID 信息与某个特定 UID 一致时，系统就认为该进程的 MAC 标签就是 UID 所绑定的 MAC 标签，例如：

```
用户名:test
用户 ID:1000
MAC 主体标签:{10,10,10}
```

则当 test 用户运行了一个 bash 程序时，该程序所产生的进程的 MAC 标签与 test 用户的 MAC 标签一致{10,10,10}，也就是说主体进程的标签与进程所属用户的标签有关。

10.2.1.3 客体标签

客体通常是目录、文件，凝思安全操作系统提供了 NAC 客体标签配置工具为客体设置标签。当主体对该客体访问时，系统会读取主客体标签按规则进行比对，如果发现有任意一条规则不满足则拒绝访问进行。客体标签与主体标签格式一致，例如：

```
MAC 客体标签：{10,10,10}
```

10.2.1.4 规则

影响访问的第三个要素是“规则”。凝思安全操作系统 MAC 机制提供的规则，格式如下：

```
<字段> <运算符> <访问控制属性>
```

- <字段>：指示当前规则比对安全标签中的 f1、f2、f3 三个之一。
- <运算符>：主体标签与客体标签运算时使用的运算符。其中包括：>、<、{、}、!、= 六种。

>	用大于运算检查<字段>指示的主客体标签域的值。
<	用小于运算检查<字段>指示的主客体标签域的值。
{	用包含于运算检查<字段>指示的主客体标签域的值（二进制比特位）。
}	用包含运算检查<字段>指示的主客体标签域的值（二进制比特位）。
!	用无关运算检查<字段>指示的主客体标签域的值。
=	用相等运算检查<字段>指示的主客体标签域的值。

> 注意
>
> “{”、“ }”是集合运算符，所以，它们比较的两个量不是单纯的数值量，而是集合量，所以当使用这两个运算符时需要检查主客体标签域中二进制位的分布情况以确定运算结果。

- <访问控制属性>：指示当规则成立后可以赋予主体的访问控制权限。
  rw：可读、可写
  r-：只读
  -w：只写
  -：不可读、不可写

10.2.1.5 MAC 配置基本流程

1、使用 MAC 客体标签配置工具为客体（文件、目录）配置客体标签。
2、修改主体标签配置文件添加新的主体标签信息。
3、修改配置文件后需要手动加载主体标签信息到安全模块。
4、然后修改规则配置文件。（可选）
5、手动加载规则配置文件。（可选）

10.2.2 规则说明

凝思安全操作系统 V6.0.100 的强制访问规则是可配置的，MAC 的规则由“规则配置文件”的内容决定，在安全模块加载时将载入这个规则文件，如果加载模块时加载的规则文件不存在，或内容为空或内容格式不正确则认为使用默认的 BLP+BIBA 规则，如果重新加载规则文件，但文件不存在或内容为空，或内容格式不正确则保留原有规则。如果主客体的关系没有满足任何一条规则，则认为允许读和写。

“规则配置文件”的路径如下：

```
/etc/security/linx/.mac_rules
```

“规则配置”文件的编写规则：

- 每行一条规则。
- 行首以“#”开头表示注释。
- 每条规则分三个部分“域”，“运算符”，“权限”。每部分的书写内容可参照 10.2.1.4 规则中描述。
- 规则文件的大小不能超过 10MB。

“规则配置文件”的范例：

```
f1 ! --
f2 > r-
f2 < -w
f2 = rw
f3 > -w
```

规则的使用参考后面的演示用例。

10.2.3 配置文件

主体标签存放在一个“主体标签配置文件”中，用来配置一个用户 uid 和三个域值的对应关系。由于凝思安全操作系统强制访问控制机制是基于 BLP+Biba 的所以设置了三个标签域以满足模型的需求。

“主体标签配置文件”的路径如下：

```
/etc/security/linx/.mac_subject_label
```

“主体标签配置文件”的标签规则：

- 每行一个主体标签
- 主体标签包括：用户 ID，三个标签域的值共四部分。
- 以“#”开头的行为注释行
- 多个相同 ID 的不同标签值的记录以最后一个为准。

“主体标签配置文件”的范例：

```
 1000       10      10      30
1001      0x12       2      12
```

主体标签的配置支持十进制和十六进制，各域用空格分隔。具体的使用方法见后面的演示用例。

10.2.4 配置工具

凝思安全操作系统提供了两个 MAC 标签配置工具，setfmac 和 getfmac。

setfmac

- 路径：/sbin/setfmac
- 作用：配置 MAC 客体标签
- 格式：

```
setfmac [-r fields] | [-n fields [-v values | -b bits_num]] file
```

-h	帮助信息

-n	选定特定的标签域，多个域之间用“,”隔开。
-v	设置 -n 指定的标签域的值，与 -n 指定的标签域的个数对应，多 值之间用“,”隔开。
-b	按位设置 -n 指定的标签域的值，和 -v 只可选其一，表示将标签域的某些 bits 位设置为朱，同一个域的多个位之间用“,”隔开，多个域之间用“:”隔开。
-r	删除 MAC 标签，可以指定删除某个或某几个域，多个域之间用“,”隔开。删除操作和设置操作是互斥的，不能同时指定。
file	目标文件。

getfmac

- 路径：/sbin/getfmac
- 作用：读取 MAC 客体标签
- 格式：

```
getfmac file
getfmac -h
```

-h	帮助信息
file	目标文件。

10.2.5 功能演示

本节通过讲解 MAC 基本功能演示，MAC 的使用方法。读者可以跟随演示用例在凝思安全操作系统 V6.0.100 上实际操作以便加深理解。

1、准备工作

凝思安全操作系统提供分权管理员机制，系统中有部分特权账户，它们的权限是 root 用户（超级管理员）权限的子集，在不同场景下代替 root 用户完成工作。他们分别是系统管理员（sysadmin）、安全管理员（secadmin）、网络管理员（netadmin）和审计管理员（audadmin）。

2、客体标签配置工具

凝思安全操作系统提供实用工具设置客体标签，setfmac、getfmac 两个命令可以完 成配置工作。

- 示例一：若想保证某文件 F 只能被某用户 U 访问，如何设置?

  为了确保某一文件只能被用户 U 访问，首先需要确定用户 U 的主体标签值，假定用户 U 的 UID 为 1000，MAC 安全标签假设为{10,10,10}。这时需要设置客体标签为{10,10,10}就可以保证只有用户 U 可以访问该文件。

命令序列如下:

```
echo "Hello" > /tmp/tmpfile          #建立一个普通文件
cat /tmp/tmpfile                     #检查文件内容
su – secadmin
#输入 secadmin 用户口令，默认为“R0ck9”
setfmac -n f1,f2,f3 -v 10,10,10 /tmp/tmpfile
#配置文件/tmp/tmpfile 的 MAC 安全标签
exit                                 #返回普通用户
cat /tmp/tmpfile
#再次检查文件内容，此时访问会遭到拒绝，因为 MAC 标签不一致。
su - secadmin                        #切换用户身份
echo "1000 10 10 10" >> /etc/security/linx/.mac_subject_label
#配置 UID1000 的用户三个标签域分别为 10,10,10。
echo s > /proc/linx-trigger
exit #返回普通用户
cat /tmp/tmpfile                     #可以看到“Hello”信息
```

上述命令序列分为两大部分，第一部分是配置客体标签，第二部分是配置主体标签。其中因为 MAC 配置属于安全配置类操作应当使用 secadmin 进行操作。

在本例中使用的 MAC 标签是{10,10,10}，这其中的含义需要根据规则来进行判断。

- 示例二：MAC 规则运算符的含义。

如默认规则列表内容，默认规则中与 f1 域相关的规则使用了集合运算“}”、“{”、“!”。这两个运算符的含义是将主客体标签的 f1 域以二进制形式按位做集合运算。本例以下面两个标签为例说明集合运算。

```
Label1:{10,20,30}
Label2:{5,20,30}
```

集合运算的需求来自于 BLP 模型，在 BLP 模型中有“类别”属性以集合方式表示。所 以凝思安全操作系统 V6.0.100 也相应支持集合运算。将 Lable1 与 Lable2 的 f1 域以二进制形式列出如下:

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

表中列出了主客体 f1 域的全部二进制位，表格第一行是位号，表格第二行是 Label1 的标签值，表格第三行是 Label2 的标签值。

当前这个例子中 Label1 和 Label2 是无关的，即 Label1 既不包含 Label2，也不被 Label2 包含。

将 Label2 作为主体标签，Label1 作为客体标签，有如下演示：

```
echo "hello3" > /tmp/testfile3
cat /tmp/testfile3
su - secadmin                                    #切换至 secadmin 管理员账户
setfmac -n f1,f2,f3 -v 10,20,30 /tmp/testfile3
echo "1000 5 20 30" >> /etc/security/linx/.mac_subject_label
echo s > /proc/linx-trigger
exit
cat /tmp/testfile3                               #用户不可读
echo "Hello3" >> /tem/testfile3                  #用户不可读
```

本例中因为 Label1 和 Label2 在 f1 域比较时是“无关”结果。所以导致主体对客体既不能读也不能写。

我们修改客体标签使其满足可读的要求，如果需要客体可被读在默认情况下需要遵循默认规则列表中的第一条规则，即主体标签应该包含客体标签，又根据标签的二进制表示形式中所示，如果需要使主体标签包含客体标签，客体标签 f1 域可以设置为 4 或 1。演示如下：

```
su - secadmin                                    # 切换至 secadmin 管理员账户
setfmac -n f1,f2,f3 -v 4,20,30 /tmp/testfile3
exit
cat /tmp/testfile3                               # 用户可读
echo "Hello3" >> /tmp/testfile3                  # 用户不可写
```

如果需要可写权限，则需要设置 f1 域中包含客体二进制位的数值，例如 7。

```
su - secadmin                              # 切换至 secadmin 管理员账户
setfmac -n f1,f2,f3 -v 7,20,30 /tmp/testfile3
exit
cat /tmp/testfile3                         # 用户不可读
echo "Hello3" >> /tmp/testfile3            # 用户可写
```

➢ 示例三：按位设置 MAC 标签。

由于 MAC 标签支持 64 位无符号整数，当数值过大时不方便计算有效的二进制位是哪些。所以 MAC 客体标签设置工具提供了按位设置 MAC 标签的功能。以默认规则为例，f1 使用了集合运算（其他域也可使用本方法），所以以 f1 域为例。标签如下：

```
Label1:{289497013546909696，20，30} Label2:
{1010072953926189056，20，30}
```

当遇到上面两个超大的数值时，直接输入难免会有错误，而且它们的含义往往并不是数值本身的含义。应该有一种方法以二进制位的形式操作 f1 域简化输入，如下：

```
echo "hello4" > /tmp/testfile4
cat /tmp/testfile4
su - secadmin                    #切换至 secadmin 管理员账户
setfmac -n f2,f3 -v 20,30 /tmp/testfile4
#设置 f2、f3 域
setfmac -n f1 -b 58,50,47 /tmp/testfile4
#设置 f1 域（按位设置 f1 域）
getfmac /tmp/testfile4                   #查看已设置的文件的 MAC 客体标签
echo "1000 0xE04800000000000 20 30" >> /etc/security/linx/.mac_subject_label
#主体标签配置（以十六进制的形式书写）
echo s > /proc/linx-trigger
exit
cat /tmp/testfile4                       #用户可读
echo "Hello3" >> /tmp/testfile4          #用户不可写
```

因为主体标签包含客体标签所以{可读，不可写}。本例中使用了以下技巧：

- 在设置客体标签时，可以按位设置。

- 在设置主体标签时，可以以十六进制形式设置。

本例中分两步设置客体标签，还可以将两条命令写成一条命令如下：

```
setfmac -n f1,f2,f3 -b 58,50,47:20:30 /tmp/testfile4
#设置 MAC 标签（按位设置 f1、f2、f3 域）
```

➢ 示例四：MAC 规则的修改。

MAC 安全标签的含义是根据安全模型确定的，而安全模型的建立是通过规则配置来完成的。所以 MAC 安全标签中三个域的含义需要结合规则来理解。首先看下面两个标签：

```
label1:{10,10,10}
label2:{10,10,12}
```

两个标签除了 f3 域不同外，f1、f2 是相同的。下面的命令以这两个标签为例。

凝思安全操作系统 MAC 规则配置文件在/etc/security/linx/.mac_rules 文件中配置。可以通过以下命令查看：

```
less /etc/security/linx/.mac_rules
```

在规则文件中会介绍规则的格式，以及规则文件的触发方式，以及出错处理原则。凝思安全操作系统在默认情况使用 Biba+BLP 模型，默认规则如下：

```
f1 } r-
f1 { -w
f1 = rw
f1 ! --
f2 > r-
f2 < -w
f2 = rw
f3 > -w
f3 < r-
f3 = rw
```

由于我们的标签中只有 f3 域不同，所以，我们暂时只分析与 f3 相关的规则条目。f3 相关的规则说明了：

2、当 f3 域主体大于客体时，主体对客体是可写的。（f3 > -w）

3、当 f3 域主体小于客体时，主体对客体是可读的。（f3 < r-）

4、当 f3 域主体等于客体时，主体对客体是对等读写的。（f3 = rw）

注意

规则中的三个字段是相互独立的，第一个字段表示这条规则检查的是哪个标签域。第二个字段表示这条规则以何种运算对主客体进行比较。第三个字段表示当本条规则生效时访问可以被授权的权限。

演示命令序列:

```
#本例假设待测试用户 id 为 1000
echo "world" > /tmp/testfile2
cat /tmp/testfile2                    #确认用户可以查看该文件
su - secadmin
setfmac -n f1,f2,f3 -v 10,10,10 /tmp/testfile2
cat /tmp/testfile2                    #无标签用户无法读写有标签客体。
echo "Hello" >> /tmp/testfile2
#无标签用户不可以对有标签客体进行写操作。
echo "1000 10 10 12" >> /etc/security/linx/.mac_subject_label
echo s > /proc/linx-trigger
exit
cat /tmp/testfile2 #用户不可以读该文件
echo "Hello" >> /tmp/testfile2#用户可写该文件
```

上述命令分别使用无标签的 secadmin 用户和有标签的 UID=1000 的普通用户对有标签的客体/tmp/testfile2 文件进行访问，分别展示了它们之间的访问控制关系。

5、无标签的主体对有标签的客体进行访问时无论读写均拒绝。

6、有标签的主体对有标签的客体进行访问时，根据规则运算的结果和主体请求的访问权限，对访问是否允许进行裁决。

继续下面操作验证规则配置的有效性:

```
su - secadmin                         #切换至 secadmin 管理员账户
echo "f3 > r-" >> /etc/security/linx/.mac_rules
echo "f3 < -w" >> /etc/security/linx/.mac_rules
echo "f3 = --" >> /etc/security/linx/.mac_rules
#配置新的规则
su -root                              #规则加载默认由 root 用户完成
echo r > /proc/linx-trigger           #加载新的规则
exit
cat /tmp/testfile2                    #新规则对无标签用户没有影响
echo "Hello2" >> /tmp/testfile2       #新规则对无标签用户没有影响
exit
cat /tmp/testfile2                    #用户可读
echo "Hello2" >> /tmp/testfile2       #用户不可写
```

上述命令的主要含义是通过修改规则配置文件并重新加载规则，原本{不可读、可写}的文件变成{可读、不可写}。

> 提示
>
> 在修改规则后应使用 root 用户执行规则加载命令，需要使用“ with root” 运行环境。

配置文件在修改后可以在不重新加载模块的情况下，动态更新配置文件的内容，目前支持的配置文件动态加载的有以下几个:

序号	文件	加载命令	备注
1	/etc/security/linx/.mac_subject_label	echo "s">/proc/linx-trigger	MAC 主体标签配置文件
2	/etc/security/linx/.mac_rules	echo "r">/proc/linx-trigger	MAC 规则文件
3	/etc/security/linx/.netlabel/.ambient	echo "a">/proc/linx-trigger	网络环境配置文件

以上配置文件 1、2 在之前的示例中已经介绍过了。3 将会在后面的章节中介绍。

10.2.6 调试方法

详见 10.3.7。

10.3 网络标签

10.3.1 功能简介

网络标签是 MAC 机制在网络环境中的扩展。基本思想是将 MAC 标签扩展到网络数据包中，通过网络数据包中所携带的 MAC 标签与接收进程的 MAC 标签按本地规则进行运算，以确定访问是否被允许。

网络标签的实现是基于在网络数据包中 IP 协议头的 IP option 域填充的 CIPSO 标记。这个 CIPSO[1]标记被定义为“网络标签”。

网络标签是单独系统内主客体标签在网络环境中的一种扩展，这种概念基于一个可信的网络环境。当网络环境中出现了异构的信息系统或当可信网络中的节点需要访问外网站点时，需要配置“网络环境配置文件”，以确保带有网络标签的系统可以和不带网络标签的系统相互通信。

根据标准，CIPSO 中有 4 个域可以配置：DOI、Tag Type、Tag Categories 和 Tag Sensitivity Level。

- DOI

  DOI 是一个 32 位的无符号非 0 整数，初始值为 12。DOI 的值在模块启动时可由模 块参数指定，如果未指定或指定错误则使用默认值。

- Tag Type

  在凝思安全操作系统中，Tag Type 值固定为 1。

- Tag Sensitivity Level

  Tag Sensitivity Level 存储标志，标志三个域中哪个域是空，哪个域是具体数值。因为在使用 Tag Categories 域时有这样的问题:

  - 当从某一位开始后面全是 0 的时候，这些全 0 位将不出现在网络数据包当中，即被舍弃了。
  - f2 域是空值但是 f3 域不是 0，那么 f2 域将会有具体数值（一般是 0）

  上面两种情况都将导致接收方无法分清接收到的标签是数值 0 还是空值。所以使用 Tag Sensitivity Level 来标记后面三个域哪个是空域哪个不是空。

  Tag Sensitivity Level 的有效值为 0~7，即二进制的 000～111，每一位分别对应 F1、F2、F3 这 3 个域，那位为 1，则表示这个域是空域。

- Tag Categories

  网络标签由 mac 的三个域 f1、f2、f3 组成，这三个域的数值存储在 Tag Categories 域中。对于不足 8 个字节的域则按照不足的位都是 0 处理。

10.3.2 配置文件

网络环境配置文件的功能是保证凝思安全操作系统可以与异构的网络操作系统互联互通。在一个可信的网络环境中仍然有可能出现异构的网络操作系统，例如笔记本电脑。对于这些便携式计算机系统，如果需要与可信网络中的节点进行通信时，可以通过将异构系统的 IP 地

1 Commercial IP Security Option.1992 年，有一个 IETF 的 draft 定义了一种 IP option，即 CIPSO，用来承载敏感性信息。

址加入到“网络环境配置文件”中使得安全节点与异构系统通信时不携带网络标签，这样异构操作系统就可以理解安全节点的数据包结构，进而与其正常通信。

凝思安全操作系统的“网络环境配置文件”是由 secadmin 管理，存放路径如下：

```
/etc/security/linx/.netlabel/.ambient
```

配置文件的格式参考如下：

```
192.168.1.100/16 1 2 3
172.16.0.100 4 5 6
172.16.0.79 - 6 -
```

第一条规则是指当发送地址为 192.168.XXX.XXX 时，网络数据包中不携带网络标签。其中“/16”表示匹配 IP 地址的前 16 个二进制位，即前两个点分域。当接收到从 192.168.XXX.XXX 来的数据包时，如果数据包没有带有 MAC 标签则它的标签被强制设置为{1,2,3}。

第二条规则是指当发送地址为 172.16.0.100 时，网络数据包中不携带网络标签。当接收到从 172.16.0.100 来的数据包时，如果数据包没有带有 MAC 标签则它的标签被强制设置为{4,5,6}。

第三条规则是指当发送地址为 172.16.0.79 时，网络数据包中不携带网络标签。当接收到从 172.16.0.79 来的数据包时，如果数据包没有带有 MAC 标签则它的标签被强制设置为{-,6,-}。“-”表示空。

10.3.3 标签使能

网络标签机制默认情况下是不使能的。如果需要使能网络标签需要以 sysadmin 身份登录系统，然后重新加载安全模块并追加 linx_netlbl_enable=1 选项参数。

10.3.4 规则说明

10.3.4.1 在网络环境中的主客体定义

- 主体：网络连接的发起者，网络数据包的发送者。
- 客体：网络连接的接受者，网络数据包的接收者。

10.3.4.2 网络请求接受的条件定义

- 当主体对客体有“写”权限时，客体接受主体的连接请求，或客体接收主体的数据包。

权限来自于主客体标签的运算结果。运算规则遵照 MAC 规则配置文件。

10.3.4.3 标签网络环境必要条件

- 通信主机必须同时使能网络标签功能。
- 没有同时使能网络标签功能的主机之间不能相互通信。

10.3.4.4 收/发数据包的细节

- 接收数据包

- 如果接收进程没有 MAC 标签则忽略对接收数据包的安全检查。
- 如果接收进程有 MAC 标签，并且接收数据包的 CIPSO 中所包含的 MAC 标签与接收进程的 MAC 标签不一致，则追加检查“网络环境配置文件”中是否有与数据包源地址相应的记录。
  - 如果找到相应记录，则使用相应记录的标签作为数据包的标签。
  - 如果没有，则丢弃接收的数据包。
- 发送数据包
  - 发送时检查“网络环境配置文件”，如果找到与待发送数据包目的地址相同的条目，则待发送数据包不包含 CIPSO 标记。

10.3.4.5 TCP/UDP 连接的区别

由于 TCP 是面向连接的协议，而 UDP 是面向无连接的协议，所以在处理这两种协议时采用不同的授权方式。

- TCP 协议

  在 TCP 协议中是“一次连接，一次授权”。TCP 协议的授权规则是在 TCP 连接建立时，进行主客体标签的检查，连接建立之后就不再检查主客体标签了。

- UDP 协议

  在 UDP 协议中是“一次接收，一次授权”。UDP 协议的授权规则是在 UDP 数据包到达接收端时进行安全检查。

10.3.5 功能演示

> **注意**
>
> MAC 标签是附着在进程上的而不是整个主机，为了描述方便，在本节中描述为“主机 A/B 携带标签”。

演示环境配置如下:

表 10. 2 网络标签演示环境配置表

主机 A	主机 B
IP:192.168.122.159	IP:192.168.122.172
MASK:255.255.255.0	MASK:255.255.255.0
两台主机可以是虚拟机，也可以是同在一个交换机上的两台物理机。不可以是一台物理机，一台虚拟机，必须要确保两台测试机在同一网段。	

在进行演示前需要重新搭建网络标签环境，在两台测试主机中输入以下命令:

```
su - sysadmin                                    #切换至 sysadmin 系统管理员账户
modprobe -r linx_sec                             #卸载当前安全模块
modprobe linx_sec linx_netlbl_enable=1
#重新加载安全模块并使能网络标签
exit
```

分别在两台机器上执行下列命令:

步骤	主机 A	主机 B	备注
1	nc -l -p 9999		主机 A 建立服务器监听
2		nc 192.168.122.159 9999	主机 B 连接主机 A 端口号 9999
3	Hello		主机 B 会接收到 Hello
4		This is HostB	主机 A 会接收到 This is HostB

两台主机可以互相通信。

网络标签功能演示:

本例中将主机 A 的用户带有一个 MAC 标签，而主机 B 中的用户并不配置 MAC 标签，查看他们之间的访问关系。标签如下:

```
Label1：{10,10,10}
```

在主机 A 中配置，如下命令:

```
su - secadmin      #切换至 secadmin 管理员账户
echo "1000 10 10 10" >>
/etc/security/linx/.mac_subject_label
#设置 MAC 标签
echo s > /proc/linx-trigge#加载新的主体标签列表
exit
```

在两测试主机中输入如下命令:

步骤	主机 A	主机 B	备注
1	nc -l -p 9999		主机 A 建立带有网络标签的服务器
2		nc 192.168.122.159 9999	主机 B 不带有网络标签，连接主机 A 带有网络标签服务

			器
3	Hello		主机 B 不会接收到 Hello
4		This is HostB	主机 A 不会接收到 This is HostB

无标签的主机 B，连接到有标签的主机 A，访问被拒绝。
在主机 B 中配置如下命令:

```
su - secadmin      #切换至 secadmin 管理员账户
echo "1000 10 10 10" >>
/etc/security/linx/.mac_subject_label
#设置 MAC 标签
echo s > /proc/linx-trigge#加载新的主体标签列表
exit
```

在两测试主机中输入如下命令:

步骤	主机 A	主机 B	备注
1	nc -l -p 9999		主机 A 建立带有网络标签的服务器
2		nc 192.168.122.159 9999	主机 B 带有网络标签，连接主机 A 带有网络标签服务器
3	Hello		主机 B 会接收到 Hello
4		This is HostB	主机 A 会接收到 This is HostB

主机 A 与主机 B 携带了相同的标签，所以主机 B 对主机 A 有“写”权限。可以连接。
在主机 B 中配置如下命令:

```
su - secadmin      #切换至 secadmin 管理员账户
echo "1000 10 10 1" >> /etc/security/linx/.mac_subject_label
#设置 MAC 标签
echo s > /proc/linx-trigge#加载新的主体标签列表
exit
```

在两测试主机中输入如下命令:

步骤	主机 A	主机 B	备注
1		nc -l -p 9999	主机 B 建立带有网络标签的

			服务器
2	nc 192.168.122.159 9999		主机 A 带有网络标签，连接主机 B 带有网络标签服务器
3	Hello		主机 B 会接收到 Hello
4		This is HostB	主机 A 会接收到 This is HostB

主机 A 与主机 B 携带了不同的标签，{1000,10,10}和{1000,10,1}，根据默认规则“f3 > -w”，所以主机 B 对主机 A 有“写”权限。可以连接。

更多网络标签的功能演示可以参考 10.3.6 案例展示，里面介绍了 MAC 应用案例，目前正运用在某大型国有企业网络中。

10.3.6 案例展示

网络标签是 MAC 在网络环境中的扩展，在许多基于 Web 的应用场景中有广泛的应用空间。大多基于 Web 场景的企业都会有基于角色的访问控制权限。在 Web 服务器中需要设定登录界面，不同级别的人通过验证口令等方式查看到不同的信息页面这种方式非常普遍。但是，系统中往往会有漏洞诸如 SQL 注入等，可以使入侵者绕过鉴权机制，直接访问到后台服务器或敏感数据。

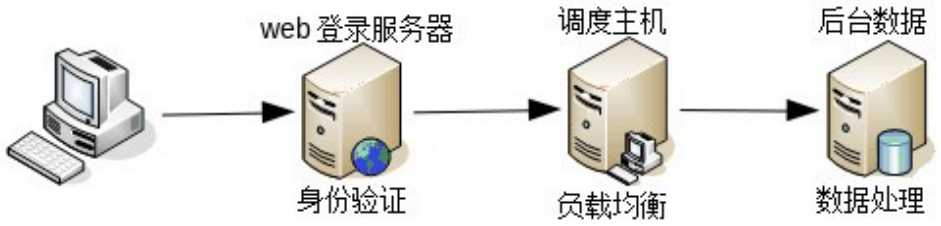


图 10. 1 Web 登录示意图

通过使用网络标签机制可以强制性保证远程用户不可以对没授权的服务器进行访问。演示如下：

表 10.4 主机相关信息

主机 A	主机 B
IP：192.168.122.159	IP:192.168.122.172
MASK:255.255.255.0	MASK:255.255.255.0
客户端	服务器
用户 testuser1：MAC 标签{1,10,10} 用户 testuser2：MAC 标签{2,12,10} 用户 testuser1：MAC 标签{3,5,10} 用户 testuser1：MAC 标签{4,10,4} 用户 testuser1：MAC 标签{5,10,15} 用户 testuser1：MAC 标签{128,10,10}	MAC 标签{31,10,10}

在客户端创建多个用户模拟局域网络中不同的使用者。一般情况下这种使用场景也可能

存在，即多用户共享同一机器资源。

在主机 B 中执行下面操作设置服务器标签：

```
su - secadmin     #切换至 secadmin 管理员账户
echo "1000 31 10 10" >>
/etc/security/linx/.mac_subject_label
#设置 MAC 标签
echo s > /proc/linx-trigge#加载新的主体标签列表
exit
```

在主机 A 中执行下面操作设置客户端标签：

```
su - secadmin              #切换至 secadmin 管理员账户
useradd -m -U testuser1
useradd -m -U testuser2
useradd -m -U testuser3
useradd -m -U testuser4
useradd -m -U testuser5
useradd -m -U testuser6
passwd testuser1           #输入默认密码 rocky
passwd testuser2           #输入默认密码 rocky
passwd testuser3           #输入默认密码 rocky
passwd testuser4           #输入默认密码 rocky
passwd testuser5           #输入默认密码 rocky
passwd testuser6           #输入默认密码 rocky
```

```
#通过 id 命令检查每一个新建用户的 id，例子中的 id 号是从 1001 开始至 1006
echo "1001 1 10 10" >> /etc/security/linx/.mac_subject_label
#设置 1001 用户 MAC 标签
echo "1002 2 12 10" >> /etc/security/linx/.mac_subject_label
#设置 1002 用户 MAC 标签
echo "1003 3 5 10" >> /etc/security/linx/.mac_subject_label
#设置 1003 用户 MAC 标签
echo "1004 4 10 4" >> /etc/security/linx/.mac_subject_label
#设置 1004 用户 MAC 标签
echo "1005 5 10 15" >> /etc/security/linx/.mac_subject_label
#设置 1005 用户 MAC 标签
echo "1006 128 10 10" >>
/etc/security/linx/.mac_subject_label
#设置 1006 用户 MAC 标签
echo s > /proc/linx-trigger      #加载新的主体标签列表
```

输入以下命令:

步骤	主机 A	主机 B	备注
1		以 UID = 1000 用户在终端中运行下面命令: Python -m SimpleHTTPServer	主机 B 启动 Web 服务器，端口 8000
2	以 testuser1 在图形界面下登录，打开终端，输入以下命令 Firefox 192.168.122.172:8000		可以看到内容
3	以 testuser2 在图形界面下登录，打开终端，输入以下命令 Firefox 192.168.122.172:8000		不可以看到内容
4	以 testuser3 在图形界面下登录，打开终端，输入以下命令 Firefox 192.168.122.172:8000		可以看到内容
5	以 testuser4 在图形界面下登录，打开终端，输入以下命令 Firefox 192.168.122.172:8000		不可以看到内容
6	以 testuser5 在图形界面下登录，打开终端，输入以下命令 Firefox 192.168.122.172:8000		可以看到内容
7	以 testuser6 在图形界面下登录，打		不可以看到内容

	开终端，输入以下命令： Firefox 192.168.122.172:8000		

“可以看到内容”的例子不做解释。主要解释“不可以看到内容”的例子:

- testuser2 看不到 Web 内容，是因为它的标签{2,12,10}和服务器的标签{31,10,10}的运算结果导致的，f3 域相等不需要考虑，f1 域是集合运算：客体{31}包含主体{2}的所有有效二进制位（置 1 的二进制位），所以 f1 域拿到了“写”权限，但 f2 域是数值运算客体{10}小于主体 12 没有拿到“写”权限，所以最终的结果是不可写，当主体不可写客体时连接被拒绝。
- testuser4 看不到 Web 内容，是因为它的标签{4,10,4}和服务器的标签{31,10,10}的运算结果导致的，f2 域相等不需要考虑，f1 域是集合运算：客体{31}包含主体{4}的所有有效二进制位（置 1 的二进制位），所以 f1 域拿到了“写”权限，但 f3 域是数值运算客体{10}大于主体{4}没有拿到“写”权限，所以最终的结果是不可写，当主体不可写客体时连接被拒绝。
- testuser6 看不到 Web 内容，是因为它的标签{128,10,10}和服务器的标签{31,10,10}的运算结果导致的，f2、f3 域是相等的不需要考虑，f1 域进行的是集合运算，而 128 和 31 的二 进制形式是无关的（既不包含也不被包含），所以没有写权限。

将每个客户端都分配相应的标签，就可以有效防止客户端通过软件漏洞访问本不应该访问的数据。

10.3.7 调试方法

本节介绍几种针对 MAC 安全机制的调试方法，帮助实施人员或用户对现场情况进行诊断。调试方法可以针对当前系统的状态信息的确认和收集，回传数据后，开发人员可以根据现场情况复现。

- 如何确定当前系统是否开启 MAC 安全机制?

  检查安全机制是否开启，可以使用命令:

  ```
  lsmod | grep linx_sec
  ```

  如果命令有返回值则说明安全机制已经开启。

  如果命令没有返回值则说明安全机制没有开启。
- 如何查看当前系统生效的主体标签列表?

  打开终端，输入如下命令:

  ```
  cat /proc/linx-mac-sub
  ```

  如果返回“linx_subject_label file content:”，表示列表为空。

  如果返回具体列表数据，则表示列表不空。
- 如何查看当前系统生效的 MAC 规则列表?

  打开终端，输入如下命令:

```
cat /proc/linx-mac-rules
```

如果返回“linx_mac_rules file content:”，表示列表为空。

如果返回具体列表数据，则表示列表不空。

- 如何查看当前安全模块的运行参数?

  打开终端，输入如下命令:

```
cat /proc/linx-module-parameter
```

  查看返回结果，其中包括四个参数:

```
Linx_type=quit          #安全模式警告提示方式
inx_netlbl_enable=0     #网络标签机制开启位
linx_mac_inherited=1    #自动生成客体标签方式
linx_netlbl_doi=12          #网络标签 DOI 的值
```

- 如何确定当前用户是否带有标签?

  打开一个终端，输入如下命令:

```
cat /proc/$$/linx/mac_label
```

  如果返回“---”，则表示当前用户没有标签。

  如果返回具体数值，则表示当前用户有标签。

- 如何获得一个正在运行的程序的标签?

  打开一个终端，使用 top 或 ps 命令找到特定程序的进程号（PID），然后输入命令:

```
cat /proc/<pid>/linx/mac_label
```

  如果返回“---”，则表示当前用户没有标签。

  如果返回具体数值，则表示当前用户有标签。

- 如何确认当前系统开启了网络标签功能?

  通过检查安全模块的运行参数，可以确定网络标签功能是否开启。当模块运行参数 linx_netlbl_enable=1 时，网络标签使能。

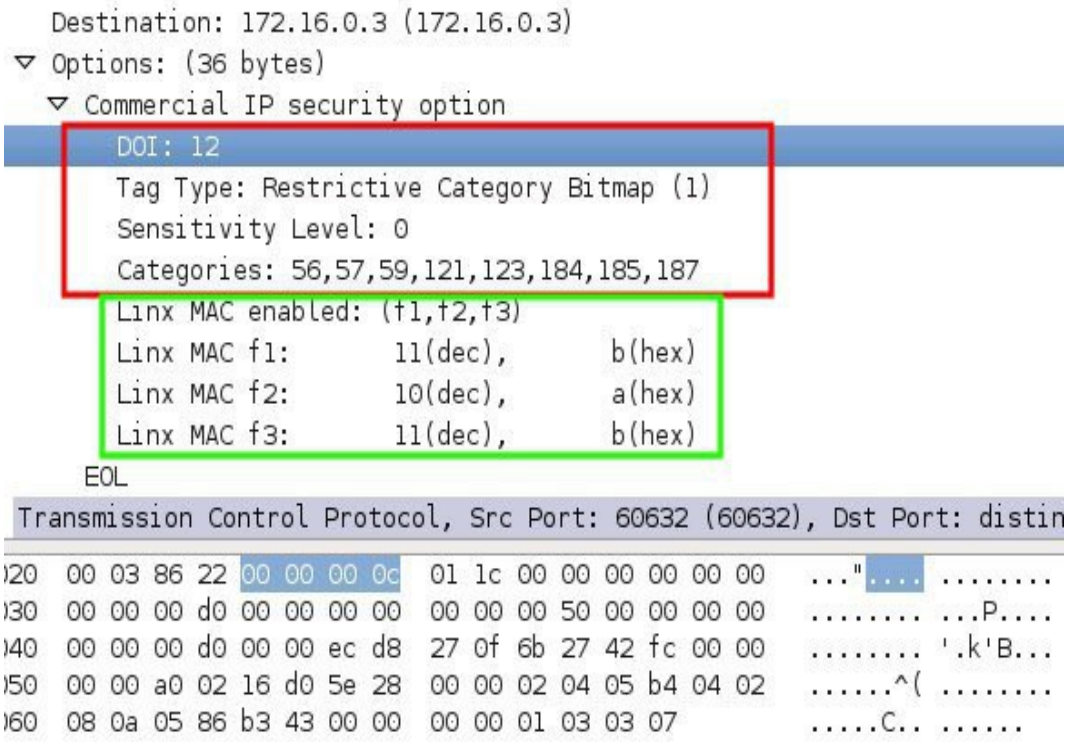
- 如何确认当前系统接收到的数据包带有网络标签信息?

  准确的而有效的方法是使用抓包工具，抓取接收数据包并查看 IP 报头，以确认是否有 CIPSO 标签。具体方法如下:

  打开一个终端，输入以下命令:

```
xhost +
su - netadmin
wireshark
```

数据包抓到后如下图:



红色方框是原始的 CIPSO 显示风格。

绿色方框是凝思 MAC 显示风格。

10.4 强制行为控制（Mandatory Behavior Control）

10.4.1 功能简介

强制行为控制（Mandatory Behavior Control，简称 MBC）是凝思安全机制的一部分，它的前身是 MEC（Mandatory Execution Control）。

MEC（Mandatory Execution Control）强制运行控制机制对系统的所有程序以及进程进行了权限范围的限定，并且在系统运行过程中进行实时监控，一旦有进程在执行时被发现超出了其运行权限范围，强制运行控制机制便强制中止该进程继续运行，保护系统的安全，并记录所有关于该事件的安全相关信息。

当前网络环境下的安全性主要表现在保护服务进程，防止入侵者通过对外提供服务的进程闯入计算机。这样，仅仅增强访问控制、安全审计等机制是不充分的，我们必须提供适当的机制保护和监视系统中运行的主体，保证该进程主体以恰当的方式对外提供服务。MBC 制运行于系统核心，能够充分保证其强制权的有效性。它可监控的对象不仅仅是特定的进程，对于受控对象家族的子进程、孤儿进程、会话组长进程，强制运行控制机制都同样有效。有了强制运行控制机制的实现，能够有效地防范系统漏洞被利用，阻止外界的入侵攻击。

MBC 是对进程执行系统调用 execve 来进行控制，控制结果为允许或者拒绝。MBC 在 MEC 的基础上将控制的行为从执行扩展到读、写和执行，可控制的客体包括普通文件、目录、符号连接、硬链接、fifo、namepipe 和 unix domain socket。

10.4.2 规则说明

在 MBC 机制中主客体访问控制规则记录在每一个主客体对象的 MBC 标签中。

MBC 标签分为主动规则与被动规则两部分。

主体的 MBC 标签继承自被执行的客体 MBC 标签的主动规则部分。

客体的 MBC 标签通过 MBC 客体标签配置工具设置。

MBC 规则格式如下：

```
<规则类别>  <访问控制属性>  <路径>
```

- 规则类别：主/被动规则。a 表示主动，p 表示被动。
- 访问控制属性：同 DAC 访问控制属性。

  ---：表示不可读，不可写，不可执行。

  rwx：表示可读，可写，可执行。
- 路径：文件绝对路径。注意以下两点：

  1、表示一个完整的路径名。必须是绝对路径名，不支持相对路径的表示，也不支持含“.”和“..”等的表示。

  2、表示某个目录下所有客体的通配表达。这种情况的规定在结尾加上“/*”字符，例如想表示/usr/bin 目录下所有的客体(不包括目录本身)，写法是/usr/bin/*。如果想表示/usr/bin 这个目录，那可以写成/usr/bin 或/usr/bin/。

MBC 机制控制进程对目录和文件客体的访问规则如下:

1、每个文件上的规则支持正、反逻辑和主动、被动控制，正和反逻辑是指允许和拒绝逻辑，主动和被动逻辑是指操作和被操作的逻辑。即:

- 这个文件名作为客体被主体访问时的被动正和被动反的逻辑（ALLOWED_BY、DENIED_BY）。
- 这个文件名作为主体要去访问别的客体时的主动正和主动反的逻辑（ALLOW、DENY）。

2、如果主体携带的规则和客体携带的规则发生冲突，那么以两个规则的拒绝的并集为准。

3、规则中分 active 行和 passivity 行，如果哪一种没有配置，则这一种默认对全部路径名都认为是可以读、写和执行的。

4、规则中支持目录通配的匹配。即可以配置允许（或被允许）拒绝（或被拒绝）对某一个目录下所有客体的访问（或被访问），而不用列举目录下所有的文件。

- 如果一个文件名同时匹配两个或以上的规则，那么以最精确匹配为主，即最长匹配。
- 如果有两个以上同样的匹配则选取这些重名中最后的一个配置。

这里目录的通配匹配只支持在结尾的通配，不支持在中间或开头等。

每一个进程有一个 MBC 安全属性。当进程对文件进行读、写或执行操作时，运行逻辑会检查 MBC 安全属性，如果返回结果是拒绝则拒绝操作。

当进程对目录进行读、写或执行操作时，规则如下:

- 当进程的 MBC 属性不允许对目录的读操作时，目录的内容（文件列表）不能被读出；反之，当进程的 MBC、MAC 和 DAC 属性都允许对目录的读操作时，目录的内容（文件列表）可以被读出。
- 当进程的 MBC 属性不允许对目录的写操作时，该进程不能在目录下创建文件、删除文 件、或转移（mv）文件到该目录下；反之，当进程的 MBC、MAC 和 DAC 属性都允许对目录的写操作时，上述操作可以被正确执行。
- 当进程的 MBC 属性不允许对目录的执行操作时，该进程不能将该目录作为自己的当前目录，也不能以该目录为中间目录去访问文件、目录等客体；反之，当进程的 MBC 和 DAC 属性都允许对目录的执行操作时，上述操作可以被正确执行。

当父进程通过系统调用 fork 创建子进程时，子进程的 MBC 安全属性与父进程一致，即使将来子进程不属于这个父进程了，其 MBC 安全属性也不变。

当进程成功执行了系统调用 exec 之后本如果进程所执行的新文件拥有自己的安全属性本则新执行后的进程的 MBC 安全属性与新文件的 MBC 安全属性一致；如果新文件没有安全属性，则新执行后的进程与执行前的进程的安全属性一致。

当某个文件的安全属性发生变化时，已执行的进程的 MBC 安全属性不随之变化。规则举例:

```
a --- /bin/ls
p --- /bin/bash
```

第一行是一条主动规则，当主体拥有本规则时它不可以对/bin/ls 文件，实施读、写、执行操作。

第二行这是一条被动规则，主体不可能拥有这条规则，当客体拥有本条规则时客体不能被/bin/bash 程序实施读、写、执行操作。

10.4.3 配置工具

凝思安全操作系统提供了两个 MBC 标签配置工具，setfmbc 和 getfmbc。

setfmbc

- 路径：/sbin/setfmbc
- 作用：配置 MBC 客体标签
- 格式：

```
setfmbc [-h] [-r pathname] [ -f filename | mbc_rules..... pathname ]
```

h	不跟踪符号连接，无此选项时忽略符号连接本身，直接将标签设置在目标文件身上。
r pathname	删除 pathname 文件上的 MBC 标签。
f filename	读取 filename，将内容当作 MBC 规则处理。
mbc_rules pathname	指明规则列表和要设置的路径名，表示重置 pathname 上的 MBC 标签为 mbc_rules，mbc_rules 字符串要使用引号括起来，规则间的不同部分以空格隔开，如果有多条规则用逗号“,”隔开

getfmbc

- 路径：/sbin/getfmbc
- 作用：读取 MBC 客体标签
- 格式：

```
getfmbc [-h] pathname
```

-h	不跟踪符号连接，无此选项时忽略符号连接本身，直接将标签设置在目标文件身上。

pathname	目标文件，读取此客体的 MBC 标签

10.4.4 功能演示

- 演示一：禁止用户使用文件编辑器修改系统敏感文件如/etc/passwd。

```
setfmbc "p --- /usr/bin/vim.basic" /etc/passwd
```

命令说明:

- setfmbc：命令名称
- "p --- /usr/bin/vim"：规则
- /etc/passwd：规则载体
- 命令含义：在 /etc/passwd 文件上标记 MBC 客体标签，标签中包含有一个被动规则，规则含义是客体对象不能被/usr/bin/vim 程序读、写、执行。

- 演示二：禁止用户在字符界面下输入关机命令。

```
setfmbc "a --- /sbin/shutdown" /bin/bash
```

命令含义：在/bin/bash 文件上标记 MBC 客体标签，标签中包含有一个主动规则，规则含义是客体对象不能主动调用/sbin/shutdown 程序执行关机操作。

10.4.5 调试方法

通过 proc 文件系统，可以对 MBC 标签进行调试。

示例一：确认一个进程是否带有 MBC 标签、并查看其内容。

假设待查询的进程 PID 号为 2119

```
cat /proc/2119/linx/mbc_label
```

命令含义：查看一个进程号为 2119 的进程的 MBC 标签。

10.5 强制能力控制（Mandatory Capability Control）

在传统朴 UNIX 安全模型中，root 用户的地位举足轻重。在 UNIX 系统中它是一个权利至高无上的用户，它可以修改别人的文件，可以关闭计算机，可以将其它用户从系统中完全剔除。对于这些操作在传统 UNIX 系统中被叫做“特权操作”。只有 root 用户才可以执行这些特权操作。这种特权集中式的系统，往往面对很多潜在的威胁，而这一潜在威胁中最为突出的就是 setuid-root 机制。当一个以 setuid-root 运行的程序被黑客劫持，黑客可以马上拿到 root 特权，进而对系统中的敏感文件进行修改劫持服务器。

凝思安全操作系统，针对这个特权集中式的系统，进行权限划分，分权管理，无 root 运行配置。通过 Linux 提供的“能力”机制进行，程序执行时所能执行操作的限制。Linux 内核提供了“能力”机制，替代传统 UNIX 模型中“特权”/“非特权”的简单划分，而是通过具体的能力要求，来判定某一个动作是否可以执行。通过这样的限制，可以确保系统中不存在“完全不被限制的”进行。

Linux 内核对系统中的能力划分较细，请参阅如下表格：

表 10. 3 用户或进程特权列表

序号	名称	含义
1	CAP_AUDIT_CONTROL	使能/关闭内核审计系统； 更改审计过滤器规则； 检索审计状态/审计过滤器规则。
2	CAP_AUDIT_WRITE	发送审计信息到内核日志。
3	CAP_CHOWN	修改客体的用户 ID 和组 ID。
4	CAP_DAC_OVERRIDE	跳过读、写、执行权限检查。不包含 CAP_LINUX_IMMUTABLE 覆盖的 DAC 访问。
5	CAP_DAC_READ_SEARCH	跳过文件读、目录读、执行权限检查，如果定义了[_POSIX_ACL]，也包括 ACL 访问的限制。 不包含 CAP_LINUX_IMMUTABLE 覆盖的 DAC 访问
6	CAP_FOWNER	跳过文件拥有者的权限检查。 不超越 MAC 和 DAC 限制。
7	CAP_FSETID	设置文件 setuid/setgid 标记位。
8	CAP_IPC_LOCK	锁定内存 (mlock(2)、mlockall(2)、mmap(2)、shmctl(2))
9	CAP_IPC_OWNER	跳过 SystemV 对象的安全检查。
10	CAP_KILL	跳过发送信号的安全检查。
11	CAP_LEASE	允许进程对与本身 FSUID 不同 UID 的文件设置租借。
12	CAP_LINUX_IMMUTABLE	可以通过 chattr 为文件配置“i”属性。当文件配

		置“i”属性后不可以被删除，不可以被重命名，不可以被写。
13	CAP_MAC_ADMIN	重写强制访问控制，本系统中无效。
14	CAP_MAC_OVERRIDE	允许改变强制访问控制规则，本系统中无效。
15	CAP_MKNOD	可以通过 mknod 创建设备节点。
16	CAP_NET_ADMIN	成各种网络相关操作（设置特权套接字选项、使能多播、网络接口配置、修改路由表）。
17	CAP_NET+BIND_SERVICE	绑定一个套接字到一个特权端口（小于 1024）。
18	CAP_NET_BROADCAST	使套接字广播，监听多播
19	CAP_NET_RAW	允许使用原始套接字
20	CAP_SETGID	允许维护进程的 GIDs 和备选 GIDs 列表
21	CAP_SETFCAP	允许设置文件的能力
22	CAP_SETPCAP	允许设置进程的能力
23	CAP_SETUID	允许维护进程的 GIDs
24	CAP_SYS_ADMIN	系统管理员能力，可以完成大部分的系统管理操作，如挂载文件系统、卸载文件系统、配额管理、交换分区使能、设置主机名、设置域名。 可以完成创建和删除 IPC 对象。 完成与 trusted、security 扩展属性相关的操作。 允许使用 lookup_dcookie 系统调用。 允许使用 ioprio_set 系统调用 当使用 socket 发送身份凭据时可以伪造 UID。 超越最大文件数的限制。 可以使用 CLONE_NEWNS 标记。 可以完成带有 KEYCTL_CHOWN 和 KEYCTL_SETPERM 的 keyctl 操作。 可以完成 madvisede 的 MADV_HWPOISON 操作。
25	CAP_SYS_BOOT	可以使用 reboot 和 kexec_reload 系统调用。
26	CAP_SYS_CHROOT	可以使用 chroot 系统调用。
27	CAP_SYS_MODULE	插入和删除模块
28	CAP_SYS_NICE	设置进程 nice 值。 设置实时调度策略。 设置 CPU 亲和性。 允许使用 migrate_pages 系统调用。 允许使用

		move_pages 系统调用。 允许使用带有 MPOL_MF_MOVE_ALL 标记的 mbind 和 move_pages 系统调用。
29	CAP_SYS_PACCT	允许使用 acct 系统调用。
30	CAP_SYS_PTRACE	允许使用 ptrace 跟踪进程。 允许对任意进程使用 get_robust_list 系统调用。
31	CAP_SYS_RAWIO	完成 IO 端口操作。 可以访问/proc/kcore。
32	CAP_SYS_RESOURCE	使用 ext2 文件系统的保留空间。 允许通过 ioctl 控制 ext3 文件系统。 重写磁盘配额限制。 增加资源限制。 重写 RLIMIT_NPROC 资源限制。 增加 SystemV 消息队列字节数上限，/proc/sys/kernel/msgmnb。 增加一个管道的字节上限，/proc/sys/fs/pipe-max- size。
33	CAP_SYS_TIME	设定系统时间。
34	CAP_SYS_TTY_CONFIG	允许使用 vhangup 系统调用。

通过能力的细化，可以使完成特定工作的进程只拥有特定能力，而不是拥有全部能力。以防止某能力过大的进程出了问题后，殃及这个系统安全。

从能力的角度来说，root 用户就是一个拥有全部能力的超级用户。分权管理就是将全部的能力分成若干组。按功能性分组，由不同的用户管理系统中不同的功能组件，进而不需要使用 root 这样的超级用户来完成一个网络 IP 地址的配置工作。

当整个系统运行时的重要功能被分配到若干个系统管理员身上以后，系统就可以在无 root 环境中运行了，即系统的基本运行时配置可以由分权管理员来完成，并不需要借助 root 用户。这样当入侵者尝试使用 root 身份登录系统时会遭到拒绝，进而阻止入侵的发生提高了系统安全性。

10.5.1 功能简介

10.5.1.1 线程的能力

凝思安全操作系统中每一个线程，都可以配置能力。每一个线程被设计有三组能力集合，每组能力集合中都包括上述的 34 个能力。

三组能力集合分别为：Permitted，Inheritable，Effective。

- Permitted

  这个集合是 Effective 的限制集，即 Permitted 集合中有的能力，Effective 集合中可以有，Permitted 集合中没有的能力 Effective 集合不能有。它也是 Inheritable 集合的限

制集，当一个进程需要往 Inheritable 集合中添加一个 Permitted 集合中没有的能力时，会失败，除非在它的 Effective 集合中有 CAP_SETPCAP 能力。

如果一个进程将 Permitted 集合中的能力抛弃了，则它永远不可能再拿到这个能力，除非它执行 execve()系统调用去执行一个 set-user-ID-root 程序或一个带有相应能力的文件。

- Inheritable

  这是一个通过来 execve 系统调用保持的能力集合，这种机制的作用是可以将进程的能 力，通过 execve 系统调用继承到新进程的 Permitted 集合中。

- Effective

  内核就是通过这个集合中的能力对线程进行安全检查的。

10.5.1.2 文件的能力

在凝思安全操作系统中，每一个客体文件也都包含有能力相关的属性，他们与进程的能力略有不同，文件的能力与进程的能力协同工作，决定在执行 execve 后这个进程的能力状态，请参阅下面的说明:

三组能力集合分别为：Permitted，Inheritable，Effective。

- Permitted（又名 forced）

  这个集合中的能力会自动赋予执行它的线程。

- Inheritable（又名 allowed）

  这个集合中的位与线程的 Inheritable 集合的位进行“与操作”，以此决定在执行 execve 后，线程的 Permitted 集合中哪些能力可以被置位。

- Effective

  这不是一个集合。这仅仅是一个使能位。

  如果这个位被使能，则在执行 execve 时，新的线程的 Permitted，会直接填充入新线程的 Effective 集合。

  如果这个位没有被使能，则在执行 execve 后，新线程的 Effective 集合是空的。

10.5.2 规则说明

凝思安全操作系统，在执行一个程序时，程序的能力计算遵循下表中所示的公式:

```
P’(permitted)=(P(inheritable)&F(inheritable))|
(F(permitted)&cap_bset)

P’(effective)=F(effective)?P’(permitted):0

P’(inheritable)=P(inheritable)
```

公式说明:

- P：代表执行 execve 之前线程的能力集合
- P’：代表执行 execve 之后线程的能力集合
- F：代表一个文件的能力集合

- cap_best：它表示一个限制，当 cap_best 中没有某一个能力时，这个能力将不可被添加到 Inheritable 集合中。

10.5.3 配置工具

凝思安全操作系统提供了四个用于配置能力的工具。分别为:

- /sbin/getcap 读取文件能力工具
- /sbin/capsh 能力管理工具
- /sbin/getpcaps 读取进程能力工具
- /sbin/setcap 设置文件能力工具

工具简介:

/sbin/getcap

- 作用：显示查询文件的能力。
- 用法:

```
getcap [-v] [-r] [-h] <filename> [<filename> ...]
```

-r	递归查找目录里的文件。
-v	显示所有查找项的能力，甚至它没有能力配置。
-h	显示帮助信息。
filename	待查询能力的文件名

/sbin/capsh

- 作用：这是一个 Bash 的封装脚本，用来维护 best 和进程的继承集。
- 用法:

```
capsh [args....]
```

--print	打印当前用户进程的能力信息。
--decode	将一个十六进制数解码为能力名称。
--drop	从 best 中移除能力
--caps	使用能力字符串设置能力
--inh	设置 inh 能力集合
--secbits	设置新的 secbits 值

--keep	设置保持能力的位
--uid	设置 UID
--chroot	设置 chroot 后的根目录
--killi	给某个进程发信号
--forkfor	新建子进程时睡眠 n 秒
==	重新执行带有-的 capsh
--	使 capsh 执行一个/bin/bash，如果不带这个选项本程序将会自动结束。

/sbin/setcap

- 作用：设置文件的能力。
- 用法:

```
setcap [-q] [-v] (-r|-|<caps>) <filename> [ ... (-r|-|<capsN>) <filenameN> ]
```

-v	返回能力配置结构或状态。
-q	显示较少的信息。
-	如果使用这个选项，能力字符串通过标准输入读取。
-r	删除能力配置。
filename	待设置能力的文件名。应该是一个常规文件，不应该是一个符号链接。

/sbin/getpcaps

- 作用：读取进程的能力。
- 用法:

```
getcaps <pid> [<pid> ...]
```

pid	进程号。

10.5.4 功能演示

> **警告**
> 能力配置是对系统影响较大的操作，如果需要实验请在虚拟机中进行。

示例一：改变系统时间

```
$ date
2013年  12月  23日  星期一  16:32:37 CST
$ date -s ’2018-02-18 20:20’
date: 无法设置日期: 不允许的操作
2018年  02月  18日  星期日  20:20:00 CST
$cp /bin/date .
$su
#setcap "cap_sys_time=ep" ./date
#exit
$./date -s "2018-02-18 20:20"
2018年  02月  18日  星期日  20:20:00 CST
```

10.5.5 调试方法

当需要对系统中进程的能力信息进行调试，可以用以下方法。

```
$cat /proc/<pid>/status
```

命令返回结果包含如下内容:

```
...
CapInh: 0000000000000000
CapPrm: ffffffffffffffff
CapEff: ffffffffffffeff
CapBnd: ffffffffffffffff
...
```

进程的每一个能力对应一个二进制位，但由于能力数量较多，这种方法不能直观的显示哪些能力被使能。可以通过以下命令进行转换:

```
$/sbin/capsh –decode=ffffffffffffeff
0xffffffffffffeff=cap_chown,cap_dac_override,cap_dac_read_sear
ch, cap_fowner,cap_fsetid,cap_kill,cap_setgid,cap_setuid,
cap_linux_immutable,cap_net_bind_service,cap_net_broadcast,
cap_net_admin,cap_net_raw,cap_ipc_lock,cap_ipc_owner,cap_sys_mo
dule,
cap_sys_rawio,cap_sys_chroot,cap_sys_ptrace,cap_sys_pacct,
cap_sys_admin,cap_sys_boot,cap_sys_nice,cap_sys_resource,cap_sy
s_time, cap_sys_tty_config,cap_mknod,cap_lease,cap_audit_write,
cap_audit_control,cap_setfcap,cap_mac_override,cap_mac_admin,34
,35,36,
37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,
58,59,
60,61,62,63
```

返回结果中数值部分表示系统尚未启用的能力位。

系统还提供一个命令可以查看进程的能力信息，命令如下:

```
$/sbin/getpcaps 11

Capabilities for ‘11’ : =ep cap_setpcap-e
```

命令返回的字符串是通过函数 cap_from_text(3)取得的。

10.6 分权管理员

传统的 Unix/Linux 只有一个管理员，根（root）。根用户运行的进程具备全部能力。权力过大当然是一个安全隐患。如果引入一些只具备部分能力的用户，我们叫它分权管理员，那就消除了这个安全隐患。分权管理员具有一部分能力，有了能力他们就可以执行原本由根用户执行的部分操作。但是，除了对能力的判断，在众多的应用程序中还存在着对执行进程的用户身份的判断（通过 getuid()等函数调用），如果进程的用户标识不是根用户标识，这些应用程序就打印错误提示然后退出。这种检查使得这些应用程序不能由非根用户执行，凝思强制能力控制机制实现了让应用软件被根用户执行和被某些分权管理员执行效果相同。

凝思安全操作系统设置 4 个分权管理员：系统管理员（sysadmin）、网络管理员（netadmin）、安全管理员（secadmin）和审计管理员（audadmin），以实现等保四级中的最小特权原则。系统管理员主要完成系统设备的管理，网络管理员主要完成网络的管理，安全管理员主要完成系统用户的管理，审计管理员用于管理审计信息。使用这种“四权分立”的原则完成系统的管理任务，各个管理员都不能控制整个系统，他们之间相互牵制，相互制约，能够防止管理员因疏忽而削弱整个系统的安全性。

用户可以分别安装和卸载 4 个管理员软件包，来添加和删除这 4 个管理员。这 4 个软件包分别为：linx-sysadmin、linx-netadmin、linx-secadmin 和 linx-audadmin。

> 提示
>
> 系统管理员（sysadmin）、安全管理员（secadmin）、网络管理员（netadmin）和审计管理员（audadmin）的初始口令均为 R0ck9，请用户务必进行修改。

10.6.1 管理员职责

四个管理员的职责划分如下:

1、系统管理员

- 关闭或重新启动系统，相关命令：shutdown、halt、init、reboot
- 检查文件系统，相关命令:

fsck、e2fsck、fsck.ext2、fsck.ext3、fsck.ext4、fsck.ext4dev、fsck.cramfs、fsck.minix、fsck.nfs

- 创建文件系统，相关命令:

mkfs、mkfs.ext2、mkfs.ext3、mkfs.ext4、mke2fs、mkfs.bfs、mkfs.cramfs、mkfs.minix、mkfs.ext4dev

- 加载内核模块，相关命令：insmod、modprobe
- 卸载内核模块，相关命令：rmmod
- 设置系统时间，相关命令：date
- 获取网络时间，相关命令：ntpdate
- 操作网络时间服务，相关命令：ntpd
- 设置硬件时钟，相关命令：hwclock

- 创建、删除或修改硬盘分区，相关命令：fdisk、cfdisk、sfdisk、parted
- 在运行时修改内核参数，相关命令：sysctl
- 挂载和卸载文件系统，相关命令：mount、umount

2、网络管理员
- 检查网络，相关命令：ping、ping6
- 命名网络设备，相关命令：nameif
- 配置网卡，相关命令：ifconfig、ifup、ifdown
- 配置路由，相关命令：route
- arp 管理，相关命令：arp
- ip 防火墙管理，相关命令：iptables*

3、安全管理员
- 添加用户，相关命令：useradd
- 删除用户，相关命令：rmuser
- 批量添加用户，相关命令：newusers
- 批量更新口令，相关命令：chpasswd
- 带锁复制口令文件，相关命令：cppw
- 验证组文件的完整性，相关命令：grpck
- 验证口令文件的完整性，相关命令：pwck
- 批量更改组口令，相关命令：chgpasswd
- 添加组，相关命令：groupadd
- 删除组，相关命令：groupdel
- 设定组口令，相关命令：gpasswd
- 修改用户口令，相关命令：passwd
- 修改用户 shell，相关命令：cheh
- 修改用户 finger 信息，相关命令：chfn
- 修改用户属性，相关命令：usermod
- 修改组属性，相关命令：groupmod
- 进行口令到期设置，相关命令：expiry

4、审计管理员
- 查看最近登录用户的 login 信息，相关命令：aulastlog
- 查看当前的系统调用名称对应的序号，相关命令：ausyscall
- 查询审计守护进程的日志，相关命令：ausearch
- 生成审计守护进程的日志的总结报告，相关命令：aureport
- 控制内核审计的行为，相关命令：auditctl
- 分发审计信息，相关命令：audispd

- 启停审计服务，相关命令：auditd

10.6.2 功能演示

- 添加用户：添加用户的操作是由安全管理员（secadmin）完成的。

```
su - secadmin                #切换到 secadmin 身份
useradd -m -U testuser1
#添加一个名为 testuser1 的用户，并建立用户的$HOME 目录及同名组
passwd testuser1  #修改用户口令
```

- 设置网络接口的 IP 地址：网络相关的操作是由网络管理员(netadmin)完成的。

```
su - netadmin                #切换到 netadmin 身份
ifconfig eth0 192.168.1.1 netmask 255.255.255 up
#设置网络接口的 IP 地址为 192.168.1.1，子网掩码为 255.255.255.0
```

- 查看审计日志信息：审计相关的操作是由审计管理员（audadmin）完成的。

```
su - audadmin                #切换到 netadmin 身份
ausearch -ts today -i
#查看今天系统的审计日志信息，并转换日志信息中的数据以便可以直接阅读
```

- 添加内核模块：系统相关的操作是由系统管理员（sysadmin）完成的。

```
su - sysadmin                #切换到 sysadmin 身份
modprobe                  #添加内核虚拟化模块
```

10.7 无 root 系统运行

凝思安全操作系统提供了无 root 运行时环境，所谓无 root 运行时环境是指在系统运行态时，系统不支持 setuid()系统调用执行，并且禁止 root 用户登录系统。但在系统的启动阶段仍可能存在具有 root 身份的后台服务。

在无 root 运行态，如果需要对系统功能进行配置可以使用上面提到的分权管理员来完成相应的工作。

提示

在无 root 系统中，任何程序不可以调用 setuid() 系统调用。所以调用 setuid() 的程序都将返回出错提示。

在无 root 系统中，禁止以 root 身份登录，所有以 root 身份登录的会话都将被终结。

第 11 章 开发

凝思安全操作系统 V6.0.100 支持多个开环境与开发工具。

11.1 开发环境

凝思安全操作系统 V6.0.100 支持 Java、C、C++、Python、Perl、Shell、Ruby、Php、 Tcl/tk、Lisp 等开发环境。

11.1.1 java 开发环境

1、查看 javac 版本

```
$ javac -version
openjdk version "11.0.21" 2023-10-17
OpenJDK Runtime Environment (build 11.0.21+9-post-Debian-1deb10u1)
OpenJDK 64-Bit Server VM (build 11.0.21+9-post-Debian-1deb10u1, mixed mode)
```

2、编译 java 程序

Hello World.java 源程序如下:

```
public class HelloWorld{
   public static void main(String[] args) {
   System.out.println("Hello World!");
   }
}
```

编译:

```
$ javac HelloWorld.java
```

3、运行 java 程序

```
$ java HelloWorld
```

输出结果:

```
Hello World!
```

11.1.2 C 开发环境

1、查看 gcc 版本

```
Linx:~# gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/aarch64-linux-gnu/8/lto-
wrapper
Target: aarch64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='L
inx 8.3.0-6-linx2' --with-bugurl=file:///usr/share/doc/gc
c-8/README.Bugs --enable-languages=c,ada,c++,go,d,fortran
,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only
 --program-suffix=-8 --program-prefix=aarch64-linux-gnu-
--enable-shared --enable-linker-build-id --libexecdir=/us
r/lib --without-included-gettext --enable-threads=posix -
-libdir=/usr/lib --enable-nls --enable-bootstrap --enable
-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-
time=yes --with-default-libstdcxx-abi=new --enable-gnu-un
ique-object --disable-libquadmath --disable-libquadmath-s
upport --enable-plugin --with-system-zlib --disable-libph
obos --enable-multiarch --enable-fix-cortex-a53-843419 --
disable-werror --enable-checking=release --build=aarch64-
linux-gnu --host=aarch64-linux-gnu --target=aarch64-linux
-gnu
Thread model: posix
gcc version 8.3.0 (Linx 8.3.0-6-linx2)
Linx:~#
```

2、编译 gcc helloworld.c 程序

Helloworld.c 源程序如下:

```
#include <stdio.h>
int main(int argc, char *argv[])
{
   printf( “hello world!” );
   return 0;
}
```

3、运行 helloworld

```
$ ./helloworld
```

11.1.3 C++开发环境

1、查看 g++版本

```
Linx:~# g++ -v
Using built-in specs.
COLLECT_GCC=g++
COLLECT_LTO_WRAPPER=/usr/lib/gcc/aarch64-linux-gnu/8/lto-
wrapper
Target: aarch64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='L
inx 8.3.0-6-linx2' --with-bugurl=file:///usr/share/doc/gc
c-8/README.Bugs --enable-languages=c,ada,c++,go,d,fortran
,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only
 --program-suffix=-8 --program-prefix=aarch64-linux-gnu-
--enable-shared --enable-linker-build-id --libexecdir=/us
r/lib --without-included-gettext --enable-threads=posix -
-libdir=/usr/lib --enable-nls --enable-bootstrap --enable
-clocale=gnu --enable-libstdcxx-debug --enable-libstdcxx-
time=yes --with-default-libstdcxx-abi=new --enable-gnu-un
ique-object --disable-libquadmath --disable-libquadmath-s
upport --enable-plugin --with-system-zlib --disable-libph
obos --enable-multiarch --enable-fix-cortex-a53-843419 --
disable-werror --enable-checking=release --build=aarch64-
linux-gnu --host=aarch64-linux-gnu --target=aarch64-linux
-gnu
Thread model: posix
gcc version 8.3.0 (Linx 8.3.0-6-linx2)
Linx:~#
```

2、编译 C++ helloworld.cpp 程序

helloworld.cpp 源程序如下:

```
#include <iostream>
using namespace std;
int main()
{
   cout << "Hello World!" << endl;
   cout << "Welcome to C++ Programming" << endl;
}
```

编译:

```
$ g++ -g -Wall helloword.c -o helloworld
```

3、运行 helloworld

```
$ ./helloworld
Hello World!
Welcome to C++ Programming
```

11.1.4 Python 开发环境

1、查看 Python 版本

```
$ python -V
Python 2.7.16
```

2、编译运行 helloworld.py 程序

helloworld.py 源程序如下:

```
#!/usr/bin/python
# Hello world python program
print "Hello World!";
```

编译运行:

```
$ python helloworld.py
Hello World!
```

或者

```
$ chmod u+x helloworld.py
$ ./helloworld.py
Hello World!
```

11.1.5 Perl 开发环境

1、查看 Perl 版本

```
rocky@linx:~$ perl -v
This is perl 5, version 28, subversion 1 (v5.28.1) built for
aarch64-linux-gnu-thread-multi
(with 65 registered patches, see perl -V for more detail)

Copyright 1987-2018, Larry Wall

Perl may be copied only under the terms of either the Artistic
License or the
GNU General Public License, which may be found in the Perl 5
source kit.

Complete documentation for Perl, including FAQ lists, should be
found on
this system using "man perl" or "perldoc perl".  If you have
access to the
Internet, point your browser at http://www.perl.org/, the Perl
Home Page.
```

2、编译运行 helloworld.pl 程序

helloworld.pl 源程序如下:

```
#!/usr/bin/perl
# Hello world perl program
print "Hello World!";
```

编译运行:

```
$ perl helloworld.pl
Hello World!
```

或者

```
$ chmod u+x helloworld.pl
$ ./helloworld.pl
Hello World!
```

11.1.6 Shell 开发环境

1、查看 shell 版本

```
rocky@linx:~$ bash -version
GNU bash，版本 5.0.3(1)-release (aarch64-unknown-linux-gnu)
Copyright (C) 2019 Free Software Foundation, Inc.
许可证 GPLv3+: GNU GPL 许可证第三版或者更新版本
<http://gnu.org/licenses/gpl.html>

本软件是自由软件，您可以自由地更改和重新发布。
在法律许可的情况下特此明示，本软件不提供任何担保。
```

2、编译运行 helloworld.sh 程序

helloworld.sh 源程序如下:

```
#!/bin/bash
echo "Hello, World!"
```

编译运行:

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

11.1.7 Php 开发环境

1、查看 php 版本

查看是否安装 php5 包，没有则安装 php5 包:

```
$ sudo apt-get install php

$ php -i phpinfo
phpinfo()
PHP Version => 7.3.31-1~deb10u5

System => Linux Linx 4.19.0-11-linx-security-arm64 #1 SMP Linx
4.19.90-1linx13 (2023-12-04) aarch64
Build Date => Sep  4 2023 21:49:25



Server API => Command Line Interface
Virtual Directory Support => disabled
Configuration File (php.ini) Path => /etc/php/7.3/cli
Loaded Configuration File => /etc/php/7.3/cli/php.ini
Scan this dir for additional .ini files => /etc/php/7.3/cli/conf.d
Additional .ini files parsed => /etc/php/7.3/cli/conf.d/10-opcache.ini,
```

11.1.8 Tcl/tk 开发环境

1、查看 Tcl/tk 版本

```
$ tclsh8.6
% put $tcl_version
8.6
```

2、编译运行 tk_hello.tk 程序

tk_hello.tk 源程序如下:

```
#!/usr/bin/wish
#Make a label "Hello World"
label .hello -text "Hello World"
pack .hello
```

编译运行:

```
chmod +x tk_hello.tk
./tk_hello.tk
```

```
$ clisp helloworld
"Hello World"
```

11.2 开发工具

凝思安全操作系统 V6.0.100 支持 eclipse、qtcreate 等开发工具。

11.2.1 eclipse

下载 eclipse-inst-jre-linux-aarch64.tar.gz，下载地址为 https://www.eclipse.org/downloads。

解压获得 eclipse 目录：

```
tar -xf eclipse-inst-jre-linux-aarch64.tar.gz
```

进入此目录，运行 eclipse

```
$ ./eclipse
```

eclipse 软件界面如图 11.1 所示。

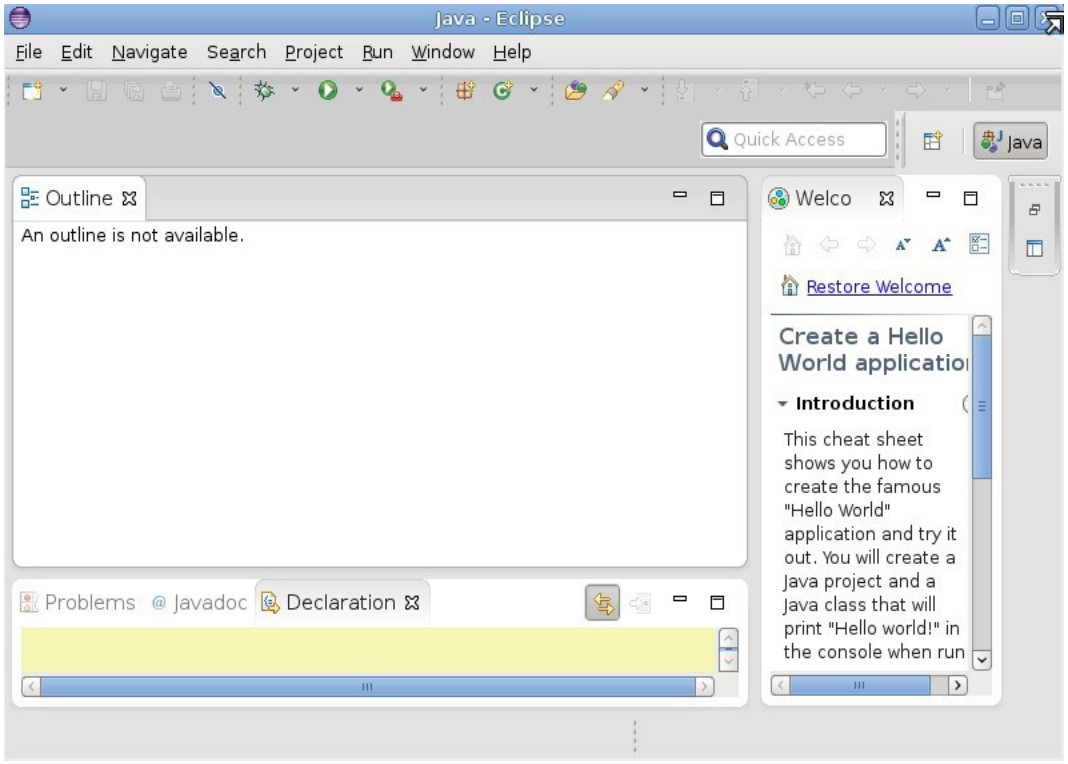


图 11. 1 eclipse 软件界面

11.2.4 qtcreater

安装编译完成的 qt5.2.1-linx_5.2.1-1_arm64.deb，安装 qt 及 qtcreater 后，在此 qt 安装目录下运行 ./bin/qtcreater。

```
$ ./bin/qtcreater
```

qtcreater 软件界面如图 11.2 所示。

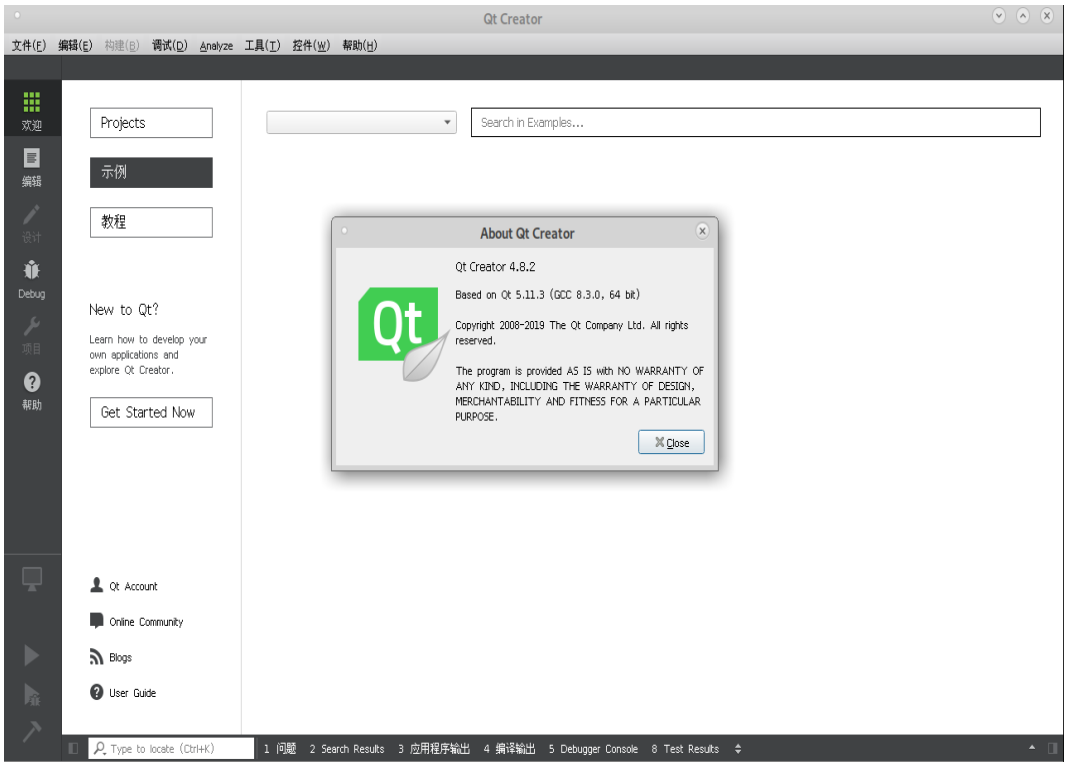


图 11. 2 qtcreater 软件界面

第 12 章 常见问题解答

本章用于解答凝思安全操作系统 V6.0.100 使用过程中可能出现的一些问题，并建议了相应的解决方法。

当您无法解决该问题或者遇到文档中未提及的问题时，请详细记录该问题并联系北京凝思软件股份有限公司技术支持部。

1、无法登录

用户输入用户名和口令无法登录到系统。

解决方法:

1）确认用户名和口令是否正确。

2）口令可能已过期，帐户已锁定，请联系管理员。

2、查找信息

系统日志可以客观的反应出系统的运行状态，当出现异常情况时，应检查系统日志信息。凝思安全操作系统会记录大量的详细信息。在您遇到系统问题时，有几个地方可以查看。

以下是最常用到的日志文件及其通常所包含的内容:

- /var/log/syslog
  全局系统日志。
- /var/log/dmesg
  内核启动日志。
- /var/log/debug
  内核日志。
- /var/log/messages
  运行时来自内核和系统日志守护程序的信息。
- /var/log/auth.log
  包含系统授权信息，包括用户登录和使用的权限机制等。
- /var/log/wtmp
  包含当前机器会话的用户登录记录的二进制文件。可使用 last 命令查看它。
- /var/log/Xorg.*.log
  来自 X Windows 系统的各种启动和运行时日志。在调试失败的 X 启动时，该日志很有用。
- /var/log/apache
  包含 Apache 服务器及客户机日志信息的目录。
- /var/log/samba/
  包含 Samba 服务器及客户机日志信息的目录。
- /var/log/mail*
  来自邮件系统的信息。
- /var/log/audit/
  包含审计日志的目录。

3、网络问题

系统的许多问题可能都与网络相关。您可以使用以下步骤来确定所遇到的网络问题的原因。

- 如果使用的是以太网连接，请首先检查硬件。请确保网线已正确地插入计算机。以太网连接器旁边的控制灯（如果有的话）应全部亮起。

  如果连接失败，请检查网线在别的机器上是否正常。如果正常，则可能是网卡引起了该问题。如果网络设置中包含集线器和交换机，也需要对它们进行检查。

- 如果使用的是无线连接，请检查网络服务器是否可与其它机器建立此无线连接。如果无法建立，请与网络管理人员联系。

- 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：

1）请检查您的序列号是否过期或无效。可执行命令 cat /proc/cmdline 查看序列号。

序列号为一串 16 个十六进制的字符，保存在/etc/default/grub 文件中，例如

```
GRUB_CMDLINE_LINUX="linx_serial=01234567890abcde security=linx"
```

修改为正确的序列号后，执行 update-grub 命令将其更新到/boot/grub/grub.cfg 文件中，然后重启机器。

2）可使用 ping localhost（请将 localhost 替换为服务器的主机名）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 ping 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，同时显示讯息目标主机不可访问，则表明您的系统或期望的服务器未正确配置或已宕机。可从其它机器运行 ping 命令来检 查您的系统是否可被访问。如果能够从其他机器成功访问您的机器，则故 障原因是服务器未在运行或未正确配置。
- 如果 ping 命令失败，同时显示未知主机，则表示名称服务未正确配置或使用的主机名不正确。请使用 ping -c <count> <ipaddress> （<count>为尝试连接次数，<ipaddress> 为主机 IP 地址）尝试连接到这一没有名称服务的主机。如果成功，则请检查主机名的拼写是否正确以及网络中的名称服务是否配置正确。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。

3）请使用 host localhost 来检查您尝试连接的服务器的主机名是否能够正确地转换为 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。

4）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。

5）如果名称服务和网络硬件已正确配置并正在运行，但是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（<fully_qualified_domain_name>为完全合法的域名）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机器传递到其目的地所经过的所有网关。它列出了每个网关的响应时间以及该网关是否可访问。请将 traceroute 和 ping 结合使用以确定故障原因并通知网络管理人员。

网络相关的检查工具较多，系统中还有其它常用的网络相关的命令如下:

- Traceroute

  查看包在网络中的路由情况可或者那个跳点（路由器）出了问题。

- netstat

  查看网络状态信息。

- tcpdump

  高级用户用来分析网络数据包格式。以检查网络通信协议是否出现问题。

详细的使用方法可以查阅系统中相关命令的 man 手册。

4、运行级别切换

完成系统启动进程后，init 启动所有在默认运行级别配置为启动的服务。默认运行级别由/etc/inittab 中的 id 给出。凝思安全操作系统 V6.0.100 默认使用 id=2。

凝思安全操作系统 V6.0.100 使用下列的运行级别:

- 0　　关闭系统
- 1　　单用户模式（single-user mode）
- 2~5 多用户模式（multi-user modes）[2]
- 6　　重启系统

您可以使用 runlevel 命令查看当前的运行级别，使用 init 命令来转换运行级别。

几种运行级别之间的切换方式有三种:

1）执行 init [0-6]命令

以系统管理员身份登录系统，执行 init [0-6]，则系统将立即切换到设定的运行级别。例如当系统处于 2 级时以系统管理员身份执行 init 1，系统将立即进入单用户模式。

2）修改/etc/inittab 文件/etc/inittab 文件中“id:2:initdefault:”字段中间的数字即为运行级别，例如将“id:2:initdefault:”中的 2 改为 1，系统重启后将从多用户模式切换到单用户模式。

3）通过 grub 菜单修改系统运行级别

系统启动进入 grub 界面时，选择任意一项，按 e 键，在二级菜单中选择 kernel 所在的行，按 e 键，在此行的最后输入如“_5”,按 Esc 键，再按 b 键，系统启动后将进入 5 级多用户模式。

注意

如果想通过第三种方式进入运行级别 1，则需要在 kernel 一行的最后输入“ _single” 而不是“ _1” 。

2　在没有定制过的系统中，运行级别 2、3、4、5 是没有区别的。

> **警告**
> 不要把运行级别 initdefault 设为 0 或 6，否则系统将不能正常启动。

5、系统启动

在系统启动时，会启动若干服务程序，这些服务程序提供整个操作系统最为核心的基本功能。例如：用户登录界面，后台服务程序，日志服务程序，IPC 通信服务等。要确保系统工能正常稳定，这些服务必须正常工作。

- 诊断哪些服务在开机时正常启动，哪些服务没有正常启动?

执行以下命令列出当前系统所有服务列表。

```
service –status-all
```

服务名左侧用-、+、?表示服务当前状态。-是未自启动，+自启动，？没有配置。

当某一个功能不可用时，首先检查相应的服务是否启动。如果没有启动，可以执行以下命令手动启动相应服务:

```
service <服务名> start
```

- 诊断开机启动过程中的异常问题

当用户发现开机启动异常，需要查看开机过程中更多的信息，以便与技术支持人员联系。可以 root 用户身份执行以下操作:

➢ 修改内核参数:

编辑 /etc/default/grub 文件，删除其中的 quite。

➢ 更新 grub 配置文件

执行命令:

```
update-grub
```

➢ 重启系统执行命令:

```
reboot
```

➢ 检查启动过程中的信息

> **警告**
> /etc/default/grub 对系统开机影响较大，请谨慎修改，必要时请联系技术支持人员。

6、Core File

当有程序异常退出时，可以通过 core 文件对崩溃的程序进行调试从而找出问题所在。

使用 core 文件的前提是需要开启 core 文件生成开关。步骤如下：

- 在运行异常代码前执行命令：

  ulimits -c 10000

- 运行一个有异常的程序：

  ./a.out

- 设置系统会生成一个 core 文件在当前目录中，core 文件的名称可能是 core 或是一串数字。
- 使用调试工具进行调试：

  gdb ./a.out core

- 进入调试界面，输入命令查看程序的调用栈：

  bt

- 这时就会发现是哪个函数调用导致程序异常，然后定位这个函数。

7、内核崩溃转储

kernel 是在系统崩溃、死锁或者死机的时候用来转储内存运行参数的一个工具和服务，打个比方，如果系统一旦崩溃那么正常的内核就没有办法工作了，在这个时候将由 kdump 产生一个用于 capture 当前运行信息的内核，该内核会将此时的内存中的所有运行状态和数据信息收集到一个 dump core 文件中以便于工程师分析崩溃原因，一旦内存信息收集完成，系统将自动重启。

配置步骤：

- 安装 Kdump 软件包：

  #apt-get install kexec-tools kdump-tools

- 安装内核调试信息软件包：

  #apt-get install linux-image-2.6.32-5-amd64-dbg

- 配置 Kdump 服务

  编辑/etc/default/kdump-tools 文件，修改 USE_KDUMP 和 debug kernel 项。

  USE_KDUMP=1
  DEBUG_KERNEL=/usr/lib/debug/boot/vmlinux-2.6.32-5-amd64

- 修改 GRUB（引导加载程序）配置编辑/etc/default/grub 文件，修改对应行内容：

GRUB_CMDLINE_LINUX_DEFAULT="crashkernel=64M quiet"

更新 GRUB

update-grub

- 重启系统

  reboot

> **警告**
>
> 当遇到系统内核崩溃或严重错误时请立即通知凝思软件技术支持部门，由技术人员帮助解决，切勿自行修改系统配置文件从而导致不可逆转的错误。

8、命令丢失

系统中提供大量的基础命令，供用户完成日常系统使用之用。系统中还可能部署很多其他的应用软件或程序。往往使用者会出现这种现象，当输入某一个命令时，系统提示找不到该命令或命令不存在。这可能会有两种可能，一是命令安装了但没有找到，二是命令没有被安装。

首先，需要确定命令是否被安装。

```
Find / -name <命令>;          #在系统中查找命令
Whereis <命令>;               #在系统中查找命令，用户手册
```

9、root 密码恢复

当某些设备由于长时间没人接管，而导致 root 口令丢失时，可以采取某种方法重置 root 口令。由于这种方法需要对所操作的设备直接控制权（即，需要在设备启动早期阶段介入设备拿到设备控制权，这需要操作人员对设备有直接操作权而非远程操作。)

在设备启动阶段会看到如图 11.1 所示界面。

图 11. 1 GRUB 启动界面

在当前界面中输入字符 e，进入命令配置界面如图 11.2 所示。

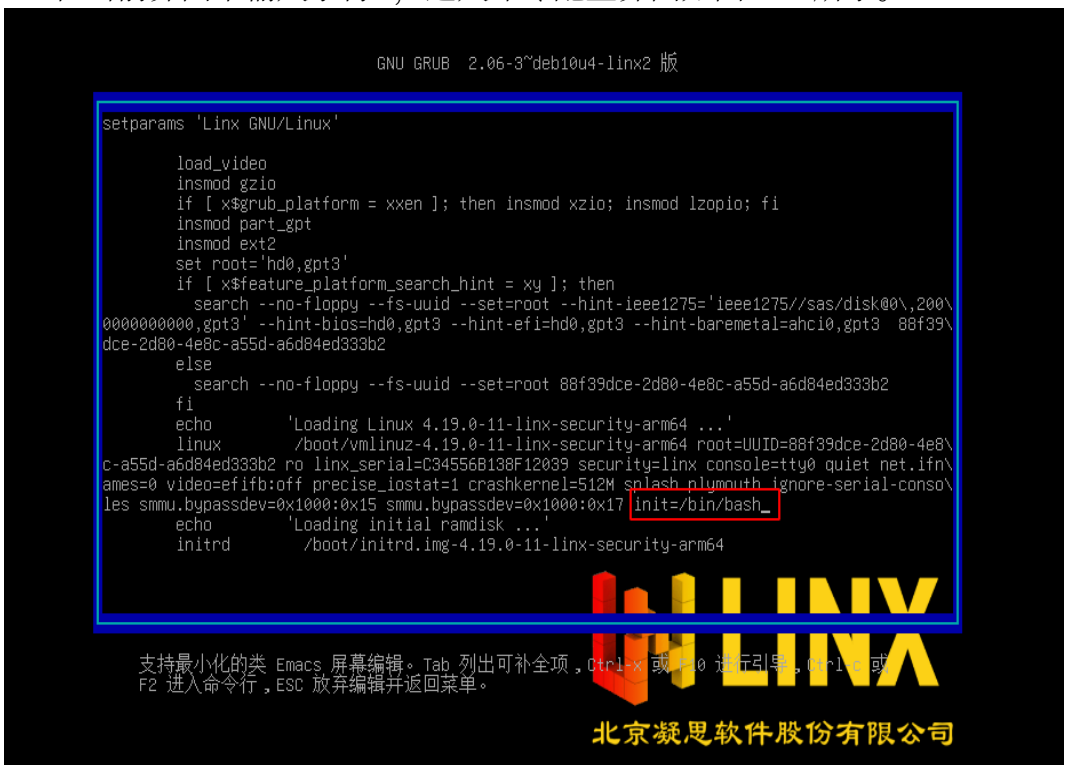


图 11. 2 GRUB 命令编辑界面

图中红框白字是需要添加的内容，写好后按 Ctrl + x 键启动系统。

系统会直接启动一个终端程序，为您使用。通过这个终端程序你可以使用 passwd 命令修改 root 用户的口令。

10、诊断和调试工具

部分应用程序在运行时会出现异常情况，这时管理员或开发者可能需要跟踪这些程序的运行轨迹，以便发现其中的问题。

在系统运行的程序可能有两种：一种是二进制程序，其执行代码是二进制的是人类不可读的；另一种是脚本程序是它是人类可读的。对于这两种程序的调试方法可能略有不同。

首先，介绍二进制程序。

当有一个二进制程序需要调试时可以用 strace 命令。

```
strace ./a.out
```

这时系统会运行 ./a.out 程序，同时将它调用的系统调用打印出来。使用者可以通过 strace 命令的输出信息了解到./a.out 程序的运行轨迹。

在系统中有很多非二进制程序，如 shell 脚本程序，当遇到这个的程序出现异常时可以通过加命令行选项的方式进行调试。

```
bash -x ./install.sh
```

当使用上述命令时，系统会运行上述命令并开启调式信息，系统会将./install.sh 每一行的运行结果打印出来，以便用户观察系统出现的问题。

附录 A 文件和目录

本章从目录的角度出发，总览整个系统，有关各个文件的详情，请参考手册页。完整的目录树分为若干部分，如果将各部分挂载到不同的分区，将更易于进行备份和系统管理。目录树中，主要的几个部分是/、/usr、/var 和 /home，各部分用途不同。具体描述如下：

/ 目录是每台机器特有的（它一般保存在本地磁盘上，但也可保存在 RAM 磁盘或网络驱动器上），其中包含启动系统和引导装入文件系统所需的文件。因此，对单用户模式来说，只使用一个分区挂载 / 目录就足够了。另外，该目录内还包含一些工具，用于修复损坏的系统和从备份中恢复丢失的文件。

/usr 目录内包含所有的命令库，手册页和其他没有变动过的文件（这些文件是普通操作期间所需要的）。/usr 目录内没有专门针对哪台机器的文件，也没有普通用户操作期间应该修改的文件。这样一来，便可以通过网络实现文件共享，从而有效地节约成本，因为这样可节省磁盘空间（要知道/usr 文件系统起码也有几百个 MB），使管理工作更容易（在更新应用程序时，只有 master/usr 需要改动，而不是逐一在每台机器上改动）。即使文件系统是在本地磁盘上，也可以采用只读方式装入它，减小系统崩溃时文件系统受损的可能性。

/var 目录中包含有变动的文件，比如假脱机目录（用于邮件、新闻、打印机等）、日志文件、格式化的手册页和临时文件。

/home 目录中包含用户的家目录，一般是系统保存用户数据的地方。把家目录和其它目录区分开是为了方便备份；其它部分通常是不需要备份的，或者说至少不需要经常备份（因为它们几乎没有变动）。一个大型的/home 目录可能必须分为若干个小型的目录，这就需要在/home 下面加一个额外的命名级别，比如/home/students 和/home/staff。

上述各个目录可以分别挂载到不同的分区。如果是一个小型的单用户系统，而且用户想一切简单明了的话，可以只挂载一个 / 目录。

不同用途的目录所需空间大小不同，可根据需要挂载到不同的分区。如果要使两个目录在同一个分区，例如/var 和/usr，可以将 var 挂载到 /usr/var 目录，然后在 / 中建立一个链接到/usr/var 的符号链接/var 。

A.1 /

一般说来，/ 应该较小，因为其中包含非常关键的文件和一个小型的，非频繁变动的文件系统。受损的 / 目录一般意味着系统不能启动，除非借助于特殊的启动设备（比如说软盘），所以一般不要轻易更改它。

/ 目录一般不包含任何文件，系统的标准启动镜像除外，这个镜像通常称为/vmlinuz。其他的所有文件都保存在 / 目录的子目录下：

1、/bin

启动期间，可供普通用户使用的命令（也可能在启动之后）。

2、/sbin

和/bin 一样，但不是供普通用户使用的，虽然在必要或经过允许的情况下，普通用户也可使用它们。

3、/etc

某台机器专用的配置文件。

4、/root

用户 root 的家目录。

5、/lib、/lib32

/ 目录上的程序所需的共享库。

6、/lib/modules

可装载的内核模块，特别是从灾难中恢复时，启动系统所需的那些模块（比如，网络和文件系统驱动程序）。

7、/dev

设备文件。

8、/tmp

临时文件。

9、/boot

启动装载程序所用文件，比如 LILO。内核镜像通常保存在这里，而不是 / 目录中。如果有多个内核镜像，这个目录就可能增长得很快，所以最好把它单独保存在一个文件系统内。这样做的另一个原因是确保内核镜像在 IDE 磁盘的前 1024 个磁道内。

10、/mnt

系统管理员临时装入的装入点。程序不会自行装入/mnt。/mnt 也可以分为若干个子目录（比如/mnt/dosa 可能是使用 MS-DOS 文件系统的软驱，而/mnt/extra 则可能和 ext2 文件系统如出一辙）。

11、/proc、/usr、/var 和 /home

其它目录的装入点。

A.2 /etc

/etc 目录中包含许多文件。下面将讨论其中的一部分。另外的文件，则应该由你决定它们属于哪个程序，并参考该程序的手册页。许多网络配置文件也被包含在/etc 内。

1、/etc/rc 或/etc/rc.d 或 etc/rc?.d

启动时或运行级别发生变化时运行的脚本或脚本的目录。

2、/etc/passwd

用户数据库，其中有一些字段指定用户名，用户真名，根目录，加密密码以及该用户的其他信息。

3、/etc/fstab

列出启动时由 mount -a 命令（在/etc/rc 或等同的启动文件内）自动装入的文件系统。Linux 系统中，这个文件还包含一些信息，这些信息和 swapon -a 自动采用的交换区有关。

4、/etc/group

类似于/etc/passwd，但它描述的不是用户，而是组。更多详情，请参考 group 手册页。

5、/etc/inittab

init 配置文件。

6、/etc/issue

登录提示出现之前的 getty 输出。通常包含对系统的简短说明或欢迎消息。其内容由系统管理员决定。

7、/etc/motd

日期消息，是在成功登录之后自动输出的。其内容由系统管理员决定。通常用来提示每个用户，比如既定的系统关闭警告等。

8、/etc/mtab

列出当前已装入的文件系统。最初是由启动脚本设置，由 mount 命令自动更新的。用于需要已装入文件系统列表时（比如说在运行 df 命令时）。

9、/etc/shadow

在已安装影子密码软件的系统上的影子密码文件。影子密码把已加密的密码从/etc/passwd 移入/etc/shadow；后者只有 root 才能读取。这样可进一步保证密码的安全性。

10、/etc/login.defs

login 命令的配置文件。

11、/etc/profile、/etc/csh.login 和/etc/csh.cshrc

登录或启动时，由 Bourne 或 C 外壳执行的文件。这些文件允许系统管理员为所有的用户设置全局默认设置。各外壳的详情，请参考手册页。

12、/etc/securetty

标识安全终端，也就是允许 root 通过哪些终端登录。一般说来，只列出了虚拟控

制台，如此一来，恶意用户不可能通过 modem 或网络攻击系统，从而获得超级用户特权（至少说很难）。

13、/etc/shells

列出受托（信得过的）外壳。chsh 命令允许用户把他们自己的登录外壳改成这个文件内列出的受托外壳。为计算机提供 FTP 服务的。ftpd 服务器进程，将复查用户的外壳是否在/etc/shells 内，如果在，将允许用户登录，如果不在，就不会让用户登录。

A.3 /dev

/dev 目录下包含所有设备的特定设备文件。设备文件的命名有特殊的约定；对这些约定的描述包括在 Device 列表中。

A.4 /usr

/usr 通常较大，因为所有的程序都是保存在这个文件系统中的。/usr 内的文件通常是系统文件；本地安装的程序和其他东西都保存在/usr/local 下面。这样一来，就能够通过安装该目录的新版本来升级系统，而不需要再次安装所有的程序。下面列出了部分/usr 子目录。

1、/usr/bin

几乎包含所有的用户命令。有些命令在/bin 或/usr/local/bin 内。

2、/usr/sbin

root 文件系统上不需要的系统管理命令，例如，大多数服务器程序。

3、/usr/share/man，/usr/share/info 和 /usr/share/doc

分别包含手册页，GNU 信息文档和名目繁多的其他文档文件。

4、/usr/include

C 编程语言的头文件。实际上，为了保持数据的一致，这个文件应该保存在/usr/lib 下面，但过去一直都采用这个名称。

5、/usr/lib、/usr/lib32

程序和子系统所用的未变动过的数据文件，其中包括一些和站点有关的配置文件。lib 这个名称源于库（library）；最初用来编写子例程的库都保存在/usr/lib 和/usr/lib32 内。

A.5 /var

/var 内包含系统正常运行时所改动的数据。它是各个系统专有的，也就是说，不能通过网络和其他计算机共享。

1、/var/lib

系统正常运行期间发生变化的文件。

2、/var/lock

锁文件。许多程序都习惯在/var/lock 内建立一个锁文件，借以表明它们正在使用某个特定的设备或文件。其他程序将注意到这个锁文件，并不再尝试使用这个特定的设备或文件。

3、/var/log

日志文件，它源于各个程序，特别是 login（/var/log/wtmp，记录所有的系统登 录和注销活动）和 syslog（/var/log/messages，通常保存所有的内核和系统程序消息）。/var/log 内的文件通常增长较快，需要定期清空。

4、/var/run

系统信息文件，其中包含系统相关信息，在系统下一次启动之前，都是有效的。例如，/var/run/utmp 内包含和当前登录用户有关的信息。

5、/var/spool

用于邮件，新闻，打印机队列和其他队列作业的目录。对每个不同的假脱机来说，在/var/spool 下面都有其自己的子目录，比如用户信箱就在/var/spool/mail 内。

6、/var/tmp

临时文件，通常存放较大或需要保存的时间比/tmp 长的文件。

A.6 /proc

/proc 内包含一个伪文件系统，用于提供和系统相关的信息（最初是进程相关信息，并由此得名）。下面将对有些比较重要的文件和目录进行解释。/proc 文件系统的更多详情，请参考 proc 手册页。

1、/proc/1

目录，其中有 1 号进程的相关信息。每个进程在/proc 下面都有一个子目录，这个子目录名就是该进程的编号。

2、/proc/cpuinfo

其中保存关于中央处理器的信息，比如型号，制造商，模型和性能等。

3、/proc/devices

其中列出了已经配置到当前正在运行的内核之中的设备驱动程序。

4、/proc/dma

展示当前正在使用的 DMA 通道。

5、/proc/filesystems

已配置到内核中的文件系统。

6、/proc/interrupts

展示哪些中断号正在使用中，以及各中断号使用了多少次。

7、/proc/ioports

展示此时哪些 I/O 端口正在使用中。

8、/proc/kcore

系统物理内存的镜像。其大小完全和你的物理内存一样，但事实上占不了多少内存；它是在程序访问它时，即时生成的（记住，除非你把它复制到别的地方，否则，/proc 根本就不占用任何磁盘空间）。

9、/proc/loadavg

系统的“装载平衡”；无意义的三个识别符，表示此时系统应该做多少操作。

10、/proc/meminfo

包含和内存使用相关的信息，其中既包括物理内存，又有交换空间。

11、/proc/modules

表明此时正在装载哪些内核模块。

12、/proc/net

目录，其中包含和网络协议相关的状态信息。

13、/proc/self

指向一个程序进程目录的符号链接，这个程序此刻正在查看/proc。如果有两个程序都在查看/proc，它们就会得到两个不同的符号链接。这主要是为了方便程序更容易得到自己的进程目录。

14、/proc/stat

关于系统的各种统计数据，比如自系统启动以来出现的页故障次数统计。

15、/proc/uptime

表明系统已启用多久。

16、/proc/version

内核版本号。

注意，上面的文件越来越发展成为易于理解的文本化文件，但有时，它们采用的格式却是难以理解的。所以，目前有许多命令将上面的文件转换为更便于理解的格式。比如，有个自由软件读取/proc/meminfo，并将指定的字节转换为千字节（同时，还增加了少许信息）。