Linux操作系统高级管理

联想集团有限公司
高性能服务器事业部

lenovo 联想

第一部分　安装Linux操作系统作为服务器软件
第二部分　单主机系统的管理
第三部分　Internet网络服务
第四部分　内部网(Intranet)网络服务
第五部分　Linux操作系统的高级网络功能

lenovo联想

第一部分　安装Linux操作系统作为服务器软件
第二部分　单主机系统的管理
第三部分　Internet网络服务
第四部分　内部网(Intranet)网络服务
第五部分　Linux操作系统的高级网络功能

lenovo 联想

第一部分 安装Linux操作系统作为服务器软件

- Linux发行版本与Windows NT的技术异同
- 按服务器配置安装Linux
- GNOME和KDE桌面环境
- 安装软件

第一部分 安装Linux操作系统作为服务器软件

- **Linux发行版本与Windows NT的技术异同**
- 按服务器配置安装Linux
- GNOME和KDE桌面环境
- 安装软件

Linux发行版本与Windows NT的技术异同

- Linux操作系统和Linux发行版本
- GNU公共许可证
- “自由”软件的优势
- Linux和NT的主要差异

Linux操作系统和Linux发行版本

- Linux操作系统以**Linux内核**作为操作系统的核心部分。内核是完成那些最基本的操作的程序，它负责其他程序的启动与终止、内存申请处理、硬盘访问、网络连接管理等方面的工作。
- Linux发行版本是由**开发工具软件、编辑器软件、GUI图形用户界面、网络工具软件**等组成的一个完整的软件包。如**RedHat**、**Caldera**、**SuSe**等商业发行版本及**Slackware**和**Debian**等非商业发行版本。各种Linux发行版本都使用Linux内核，最基础的操作都是一样的，它们之间彼此产生差异的主要原因是各自附带不同的“增值”工具软件。

GNU公共许可证

- GNU公共许可证(GNU Public License,GPL)是GNU计划中产生的最重要的事物。这个许可证明确表示：按照这个许可证发行的软件是自由的，任何人都不能剥夺这种自由。获得某个软件再把它转卖给其他人是合法的，就是加价获利也没什么不可以；但在转卖过程中，卖方必须把完整的源代码及对它的任何增补都完整地转移给买方。因为这份经过转卖的软件依然遵守着GPL许可证制度，所以它还可以自由发行，允许再次转卖给其他人获利。这个许可证制度中最重要的部分就是其免责条款，即程序开发人员对他们编写的软件在事实使用中引起的损失将不承担任何责任。

“自由”软件的优势

- 第一，由于源代码公开，在编程同行的注视之下，代码本身中的错误将比较容易被查出并迅速纠正；
- 第二，在GPL制度下，程序开发人员发表代码的时候能够不再考虑法律诉讼方面的问题。

Linux和NT的主要差异

- 单用户、多用户、网络用户情况的比较

  Linux允许多个用户登录到服务器上运行程序，而非S/C结构

- GUI图形界面与操作系统内核的彼此相对独立

  在Linux中用户图形界面是用户级的应用程序，与内核无关

- Windows中的“网络邻居”概念

  Linux使用网络文件系统(NFS)共享远程磁盘空间

- Windows中的注册表文件与文本文件的比较

  Linux中各种配置文件通常被保存为一系列文本文件

- 域的概念

  Linux系统网络安全模型的基础是网络信息服务（NIS）

第一部分 安装Linux操作系统作为服务器软件

- Linux发行版本与Windows NT的技术异同
- **按服务器配置安装Linux**
- GNOME和KDE桌面环境
- 安装软件

按服务器配置安装Linux

- 安装之前
- 安装RedHat Linux操作系统

安装之前

- **硬件设备**
  查看硬件兼容性清单（HCL） http://www.redhat.com/hardware
  普遍原则：避免使用最新的硬件设备和软件配置
- **服务器主机的规划**
  稳定性、可用性、运行性能
  规则：严格禁止使用服务器进行娱乐；确保良好的操作环境；设置后备电源；尽量避免启动X-Window；禁掉所有不需要的功能；没有充足理由不要重新编译内核
  信条：如果到目前为止很好，那么继续下去会更好。
- **双引导系统**
  Linux可以和Windows共存。
- **安装方式**
  通过本地CD-ROM光盘进行安装；通过网络进行安装
- **安装后可能面临的问题**
  硬件不支持；硬件设备组合彼此冲突；CD-ROM光盘介质有问题；软件本身的“bug”……
- **寻求帮助**
  技术支持电话；提供软件安装服务的商业公司；互联网站点

安装RedHat Linux操作系统

- **制作引导盘**

  开始安装引导可以使用引导软盘或CD-ROM光盘，如果要安装的机器没有光驱或不支持CD-ROM引导，就需要制作一张引导软盘。

  - 对于Unix，可以使用dd命令将安装CD-ROM光盘中的下列文件写入软盘

    \image\boot.img

  - 对于Windows，可以使用安装CD-ROM光盘中的下列程序

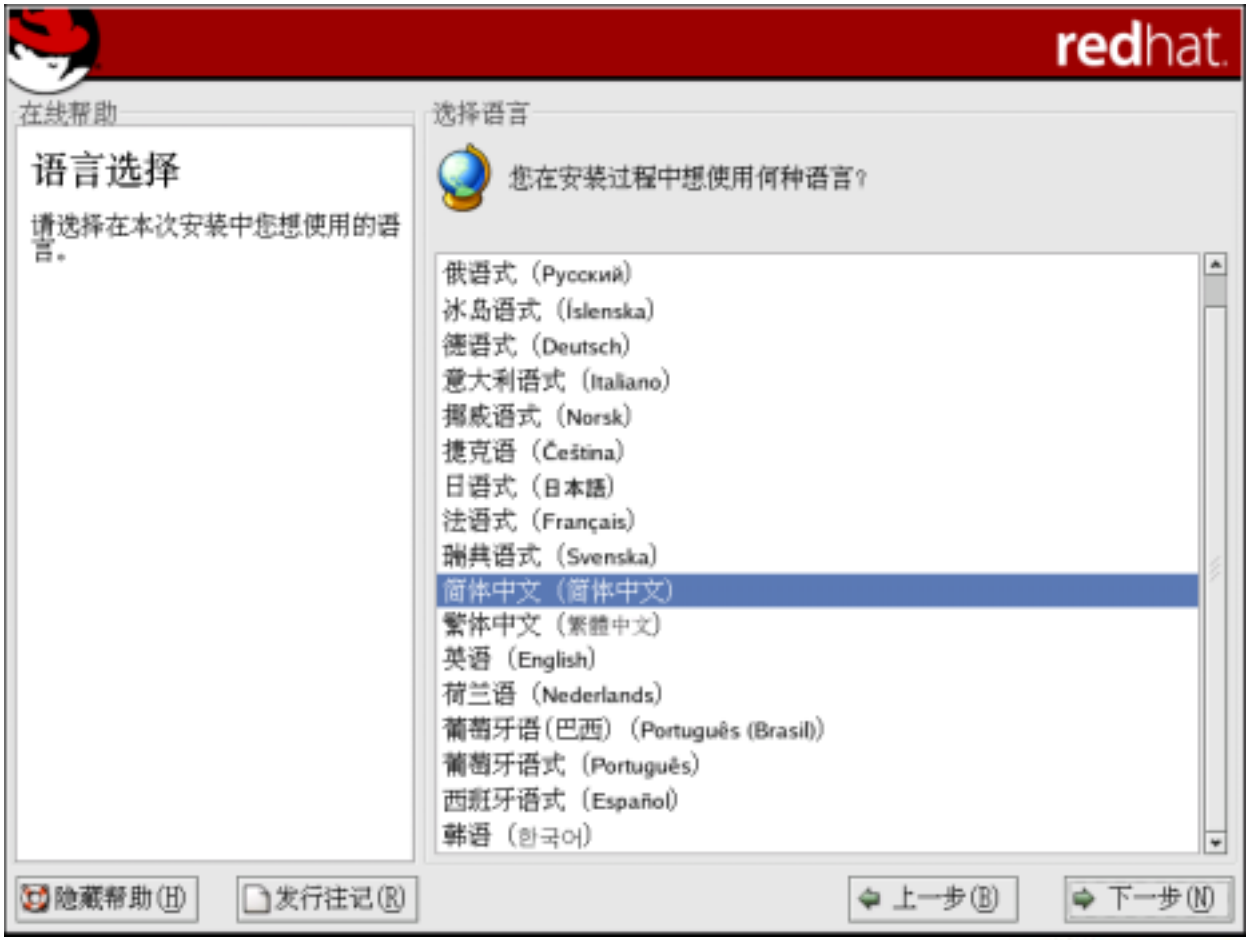
    \dosutils\rawrite.exe

    运行这个可执行文件，程序会自动提示输入源文件名和目的软盘，源文件即为**\image\boot.img**。

安装RedHat Linux操作系统（续）

- 开始安装

1、语言选择

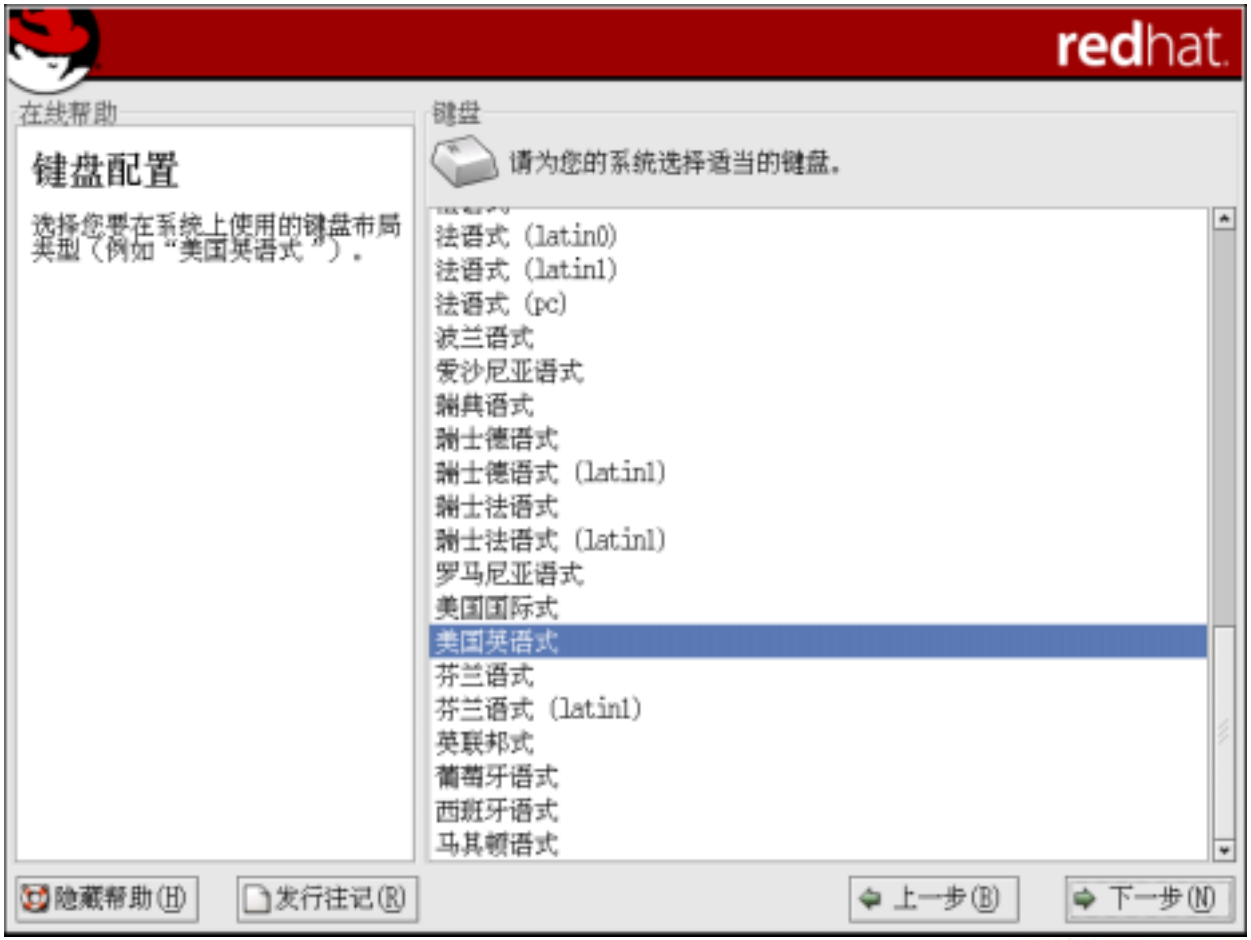


lenovo联想

安装RedHat Linux操作系统（续）

- 开始安装

2、键盘配置



安装RedHat Linux操作系统（续）

- 开始安装

3、鼠标配置

安装RedHat Linux操作系统（续）

- 开始安装

4、选择安装还是升级

安装RedHat Linux操作系统（续）

- 开始安装

5、安装类型

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

  6、为Linux建立分区

  - **/usr** 这里将保存所有程序文件（类似于C:\Program Files子目录）。
  - **/home** 这里有每一位用户的登录子目录。这样做可以防止用户消耗掉硬盘上的全部空间，为其他关键组件（比如各种系统记录文件）留出余地。
  - **/var** 各种系统记录文件的最终保存位置。由于系统记录文件可能会受到来自本系统以外的用户的影响，因此把它们单独保存到另外的分区上是十分重要的，这样就可以防止别人通过生成大量登录数据项填满整个硬盘而施行的“拒绝服务”（Denial of Service，DoS）攻击。
  - **/tmp** 这里将用来保存各种临时文件。因为这个子目录的设计目的就是要让任何一个用户都可以对它进行写操作，所以我们必须保证不会因为某些冒失用户的滥用而让这个子目录扩张填满整个硬盘；我们采用为它单独开辟一个分区的方法来保证这一点。
  - **Swap** 这并不是一个用户能够访问的文件系统，它是保存虚拟内存(virtual memory)文件的地方。

安装RedHat Linux操作系统（续）

- 开始安装

6、为Linux建立分区（续）

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

7、自动分区

lenovo 联想

安装RedHat Linux操作系统（续）

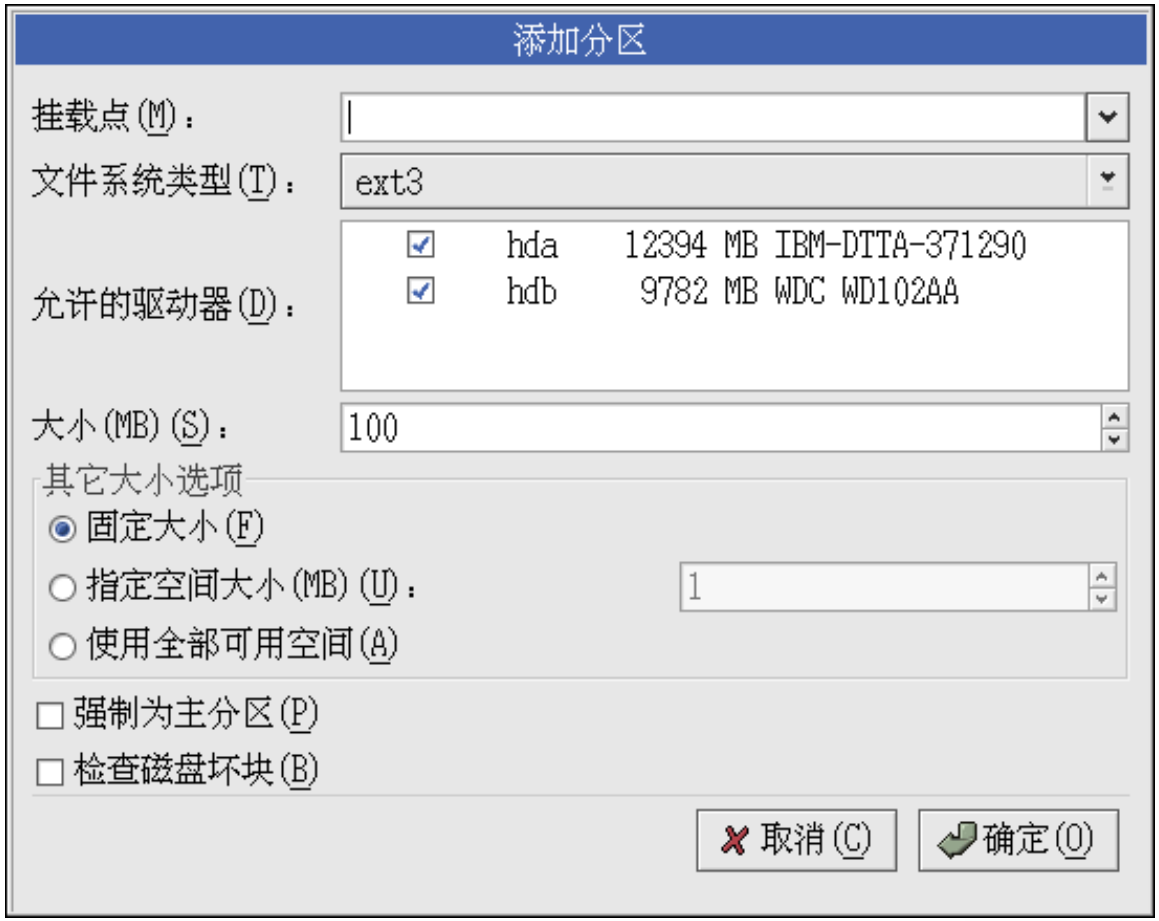
- 开始安装

8、手工分区

安装RedHat Linux操作系统（续）

- 开始安装

9、添加分区



安装RedHat Linux操作系统（续）

- 开始安装

10、引导装载程序配置

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

11、高级引导装载程序配置

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

12、网络配置

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

12、网络配置（续）

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

13、防火墙配置

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

14、附加语言支持

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

15、时区配置

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

16、设置Root口令

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

17、验证配置

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

18、软件包组的选择

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

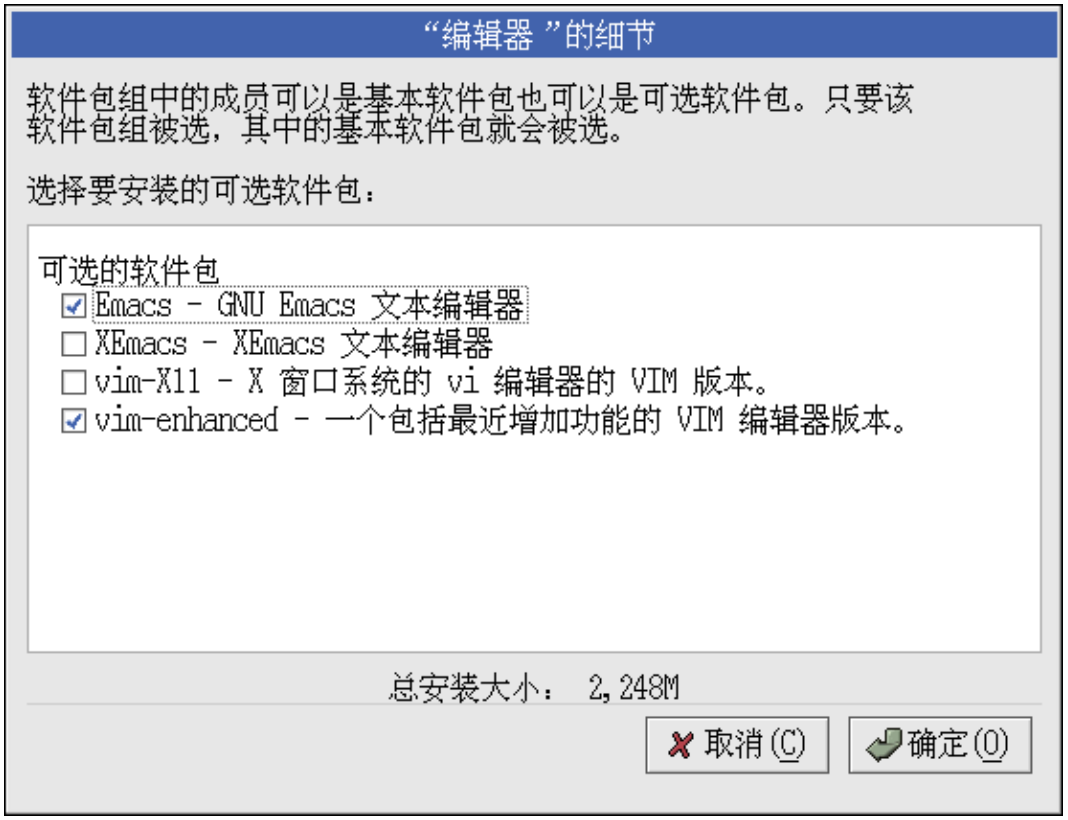
18、软件包组的选择（续）

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

18、软件包组的选择（续）



安装RedHat Linux操作系统（续）

- 开始安装

18、软件包组的选择（续）

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

18、软件包组的选择（续）

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

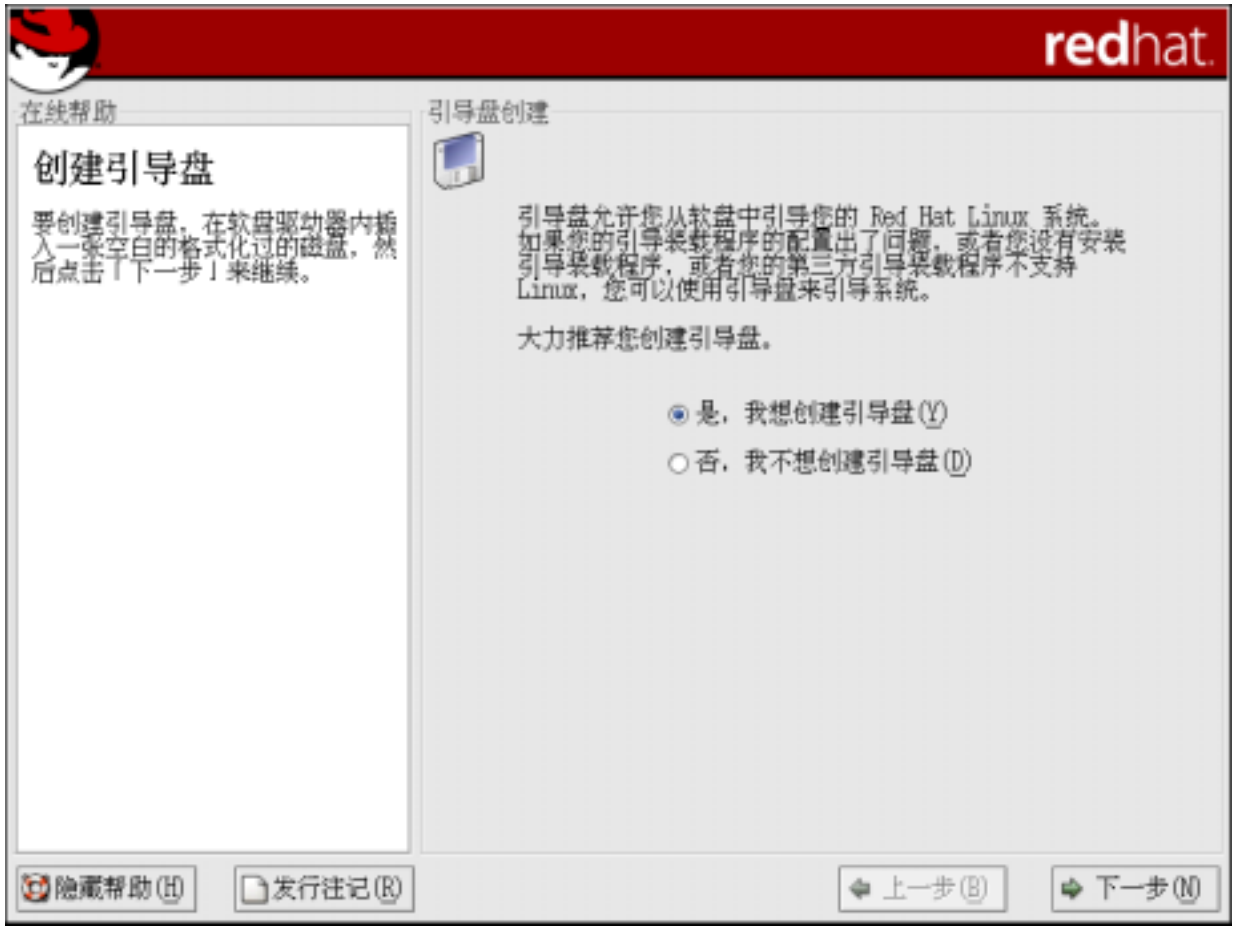
19、安装软件包

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

20、创建引导盘



lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

21、显卡配置

lenovo联想

安装RedHat Linux操作系统（续）

- 开始安装

22、X配置–显示器

lenovo 联想

安装RedHat Linux操作系统（续）

- 开始安装

23、X配置–定制

安装RedHat Linux操作系统（续）

- 开始安装

24、完成安装

大功告成！安装过程结束了。在系统重新启动的时候，**千万记得把引导系统的CD-ROM光盘和软盘都取出来**。

安装RedHat小结

在按服务器配置安装Linux中我们讨论了以下几个过程：建立一个服务器、选择正确的硬件设备、建立正确的操作环境、安装RedHat Linux 。

在我们真正开始安装RedHat Linux操作系统之前的各种讨论题目都适用于用户建立的任何一台服务器----不管使用哪一种操作系统。

安装RedHat本身的步骤也比较简明。只要有人见识过老版本的安装过程，就一定会注意到现在的过程变得多么的简单、需要选择的选项又有多么的少。

Linux操作系统的精彩之处在于：即使这些选项都不再出现在安装过程中了，用户还是可以在完成安装并实际启动系统之后再来改变这些配置，把它们细调到自己满意的程度。

第一部分 安装Linux操作系统作为服务器软件

- Linux发行版本与Windows NT的技术异同
- 按服务器配置安装Linux
- **GNOME和KDE桌面环境**
- 安装软件

GNOME和KDE桌面环境

- X-Window的历史
- KDE桌面环境
- GNOME桌面环境

X-Window的历史

- 以Unix操作系统为参照的操作系统，比如Linux的设计人员在开始设计用户操作环境的时候，认为他们提供给用户的操作界面应该是百分之百地不依赖于核心操作系统的。作为其结果，Linux的核心（即它的内核）是完全与它的用户操作界面分开的。
- 由于规模如此之大的一族程序完全与核心操作系统无关，因此这也就极大地增加了系统的稳定性。在Windows或者在Mac OS环境下，如果GUI崩溃了，用户就必须重新引导计算机。而在Linux操作系统下，用户可以终止GUI后再重新启动它，完全不会影响到系统提供的其他服务。
- 在二十世纪80年代中期，出现了一个与操作系统无关的图形化用户环境的基本标准，叫做X-Window。“X”只是一个简单的定义，说明应用程序与图形硬件设备之间进行通信的方法。另外X-Window还精心设计了一组功能函数，程序员们可以调用它们来对窗口进行基本的操作。

X-Window的历史（续）

- **X-Window的不足之处**
  - **不友好的编程环境、不友好的用户界面**

商业化的UNIX供货商试图使用**CDE**（通用桌面环境）来弥补这个漏洞，这样他们的用户就可以得到一个稳定的程序外观和操作感觉；另外还为X-Window开发了一个改进的函数库，叫做**Motif**。对Linux来说，这两方面的发展都出现了问题，因为它们与开放源代码的理想是背道而驰的。

- **进入GNOME和KDE**

到了90年代后期，经过日积月累，出现了两种针对X-Window难题的解决方案：它们就是**GNOME**和**KDE**。KDE提供了一个新的窗口管理器程序和必要的函数库，大大简化了为它编写应用程序的问题。GNOME则准备了一个通用框架供其他窗口管理器程序和应用程序工作在其环境中。两种方案各有千秋，但由于都工作在X-Window顶层，因此它们并非完全不兼容。

KDE桌面环境

- KDE是一个桌面环境（是**K Desktop Environment**的简称），它与我们曾经描述过的典型窗口管理器程序稍有不同。除了描述了界面的外观以外， KDE还提供了一整套函数库，应用程序可以使用它实现这个窗口管理器程序提供的某些特殊功能。这包括诸如拖放操作的支持、标准化的打印支持等等。
- 这种窗口管理技术的不足之处在于一旦设计了某个应用程序运行在KDE中，它就必须有KDE才能正常工作。这是对早期的窗口管理器程序的一个巨大改变，当时的应用程序都是与窗口管理器程序完全无关的。从程序员的角度看，KDE提供了一个函数库，使用它比直接对X接口编程要容易得多。KDE还提供了一个标准化的面向对象的框架结构，允许使用一组工具程序生成另外一个X-Window本身不能提供的东西。

KDE桌面环境（续）

- **许可证问题**

  在以前，还存在着一些由**Qt函数库**开发人员设置的许可证限制方面的问题，而Qt库就是KDE的基石。它禁止在未付费情况下对KDE进行商业方面的使用。GNOME项目就是因为这个限制而开始的。最近，KDE的许可证经过了修改。修改后的许可证–大家都称之为**QPL**，现在更加开放并允许商业化的使用了；但是它还是与随Linux发行版本发行的大多数软件包使用的**GPL**或者**伯克利**（Berkeley）风格的许可证不一样。访问KDE的互连网站点可以找到更多关于这个许可证的资料。 **http://www.kde.org**

- **启动X-Window和KDE**

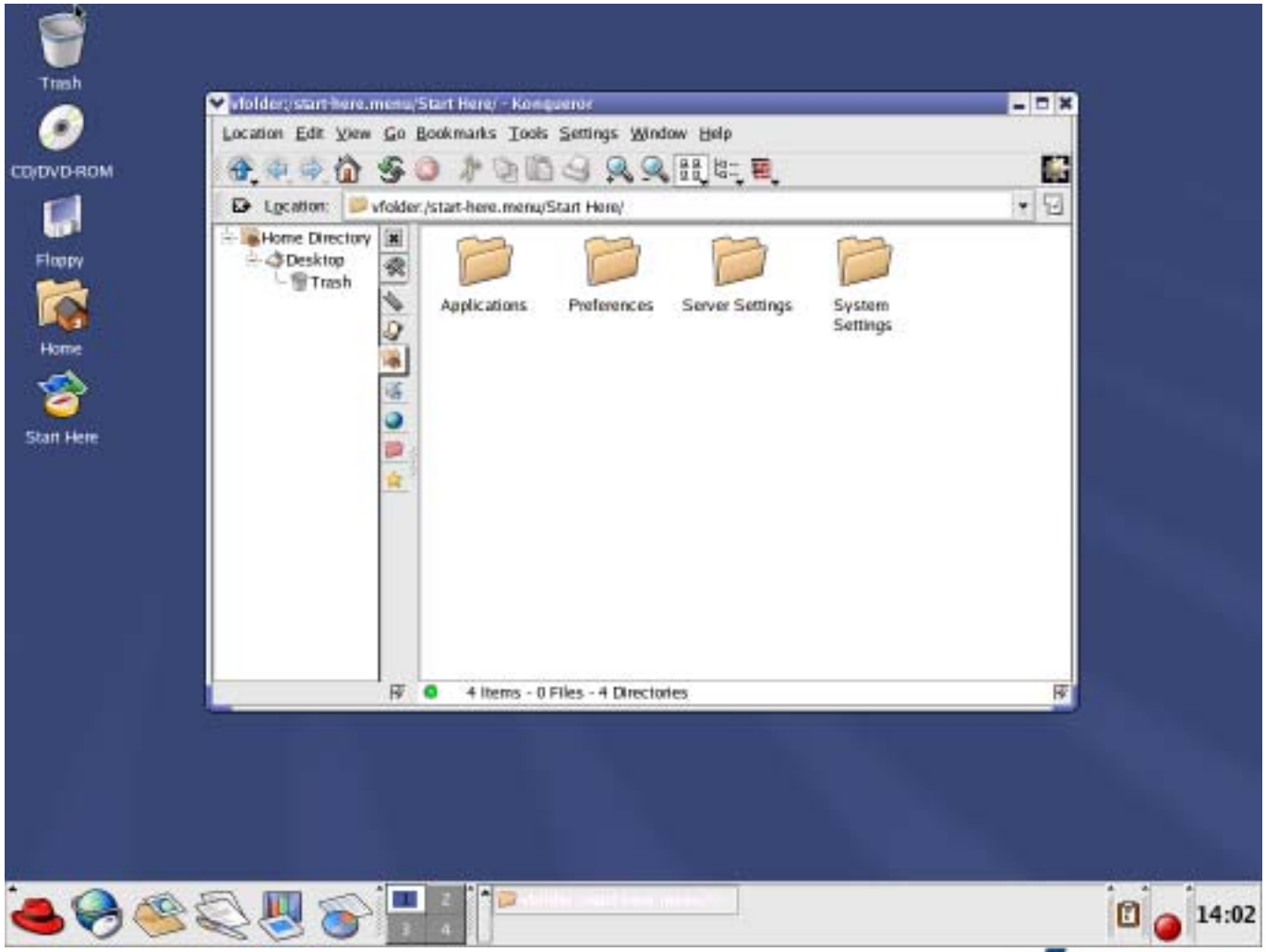
  登录并执行 startx就可以进入X-Window环境：

  ```
  [root@lenovo/root]# startx
  ```

  如果进入的是GNOME桌面环境，我们可以使用**switchdesk**命令更改为使用KDE桌面环境

KDE桌面环境（续）

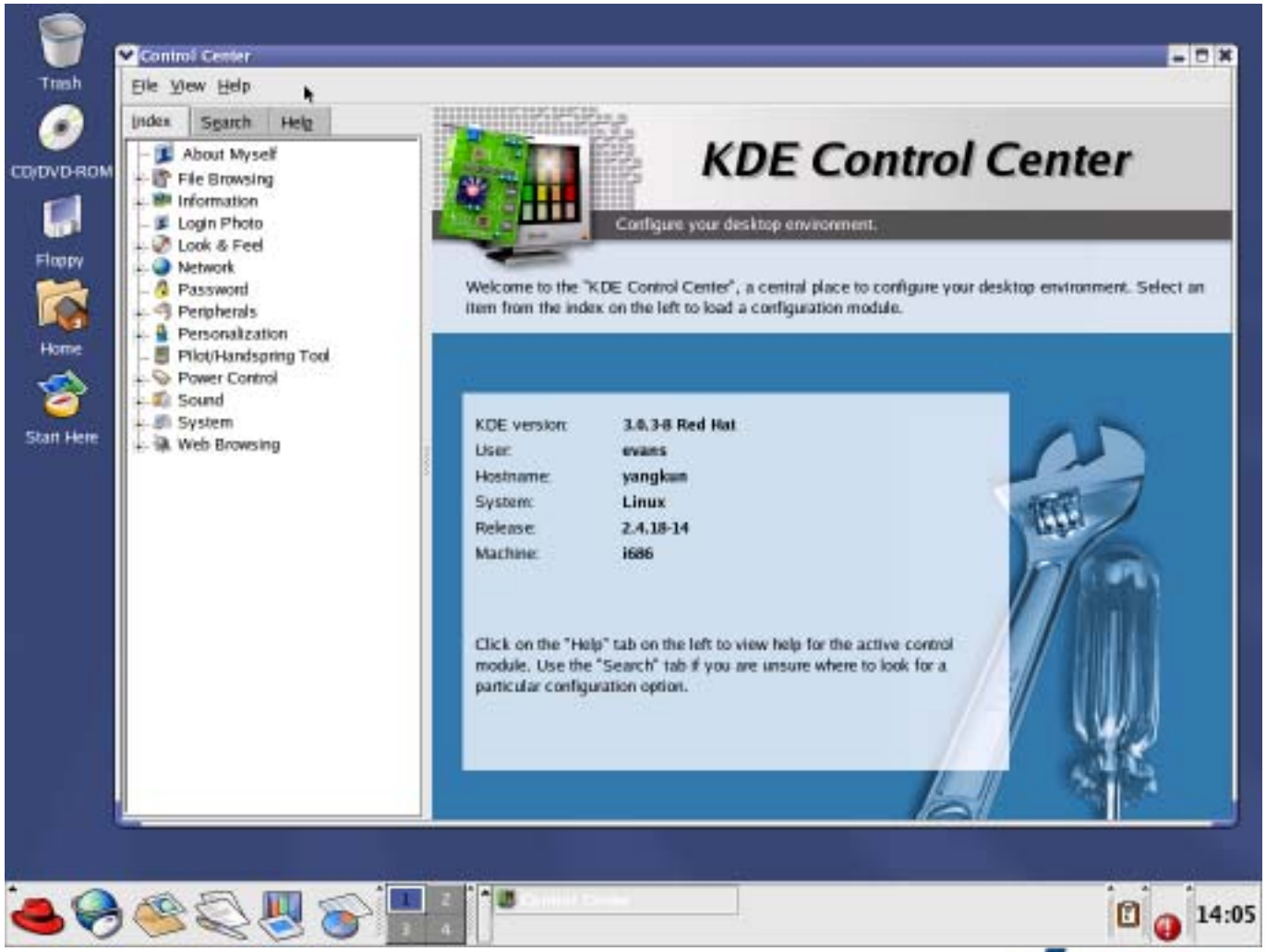
- KDE的“File Manager”（文件管理器）



lenovo 联想

KDE桌面环境（续）

- KDE的“Control Center”（控制中心）



KDE桌面环境（续）

- KDE的“Control Center”（控制中心）（续）

KDE的控制中心（**Control Center**）非常像Windows中的“**Control Panel**”（控制面板），它可以用来完成桌面的配置工作。Control Center（控制中心）提供了非常多的工具让读者按照自己的想法配置KDE。这包括支持多种桌面风格、颜色、背景、屏幕保护器、特定应用程序以及某些类型的硬件设备等等。

可以通过Control Center进行配置的几种常见任务：

1、使用多个桌面；

2、改变背景颜色；

3、改变桌面颜色；

4、改变屏幕保护器等等。

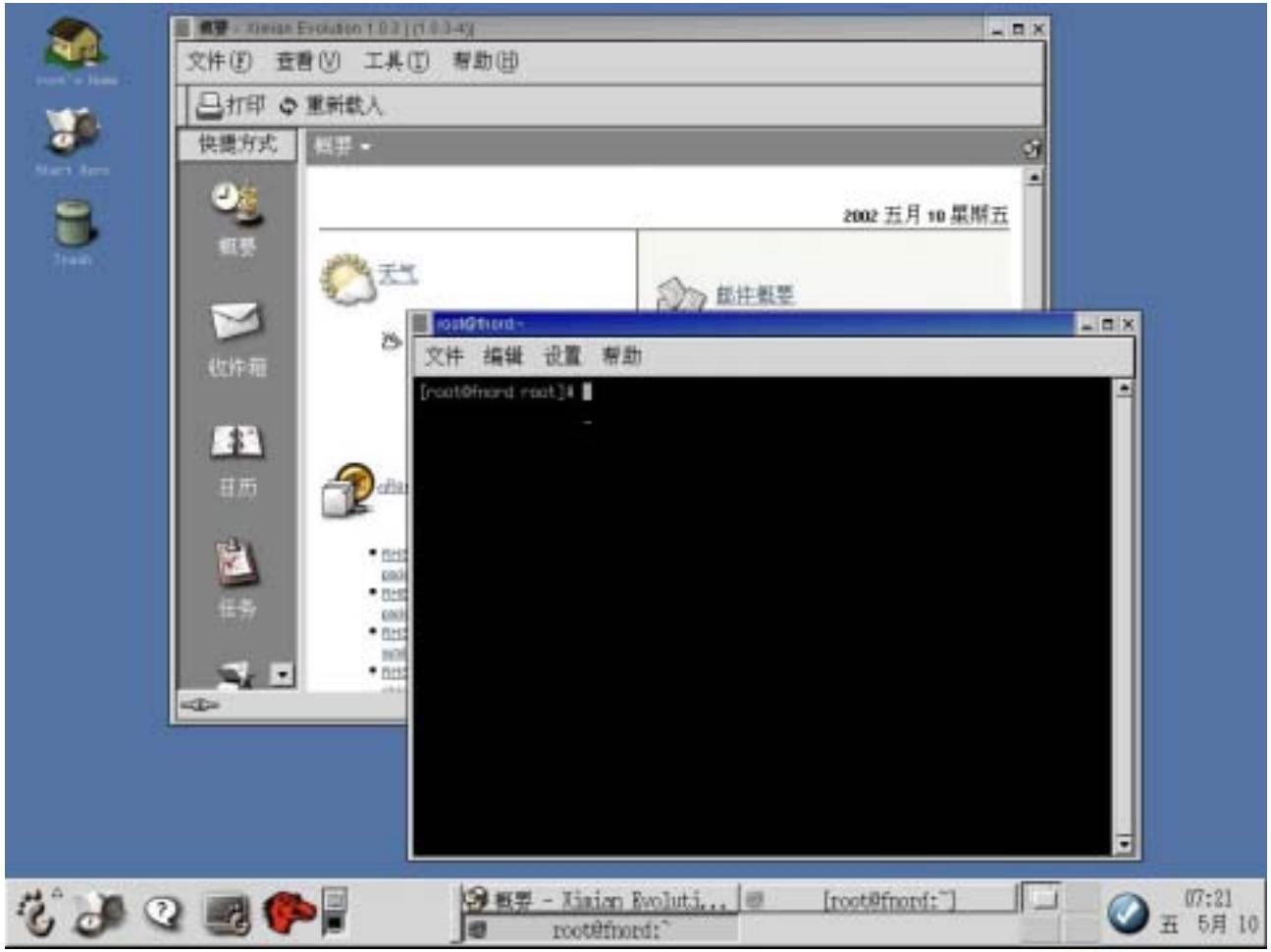
GNOME桌面环境

- **GNOME**（**G**NU **N**etwork **O**bject **M**odel **E**nvironment，GNU网络对象模型环境）。与KDE一样，都提供了方便开发和使用的完整桌面环境及应用程序框架。使GNOME与众不同的是它实现这些目的的手段。与KDE不同，GNOME并不是一个窗口管理器程序。GNOME提供了开发函数库和操作任务管理–这些都是作为普通用户的我们所看不到的基本特色。在这个基础上是一个窗口管理器程序，它负责管理桌面的一般外观。缺省的窗口管理器程序是**Enlightenment**。

GNOME桌面环境（续）

- 启动X-Window和GNOME

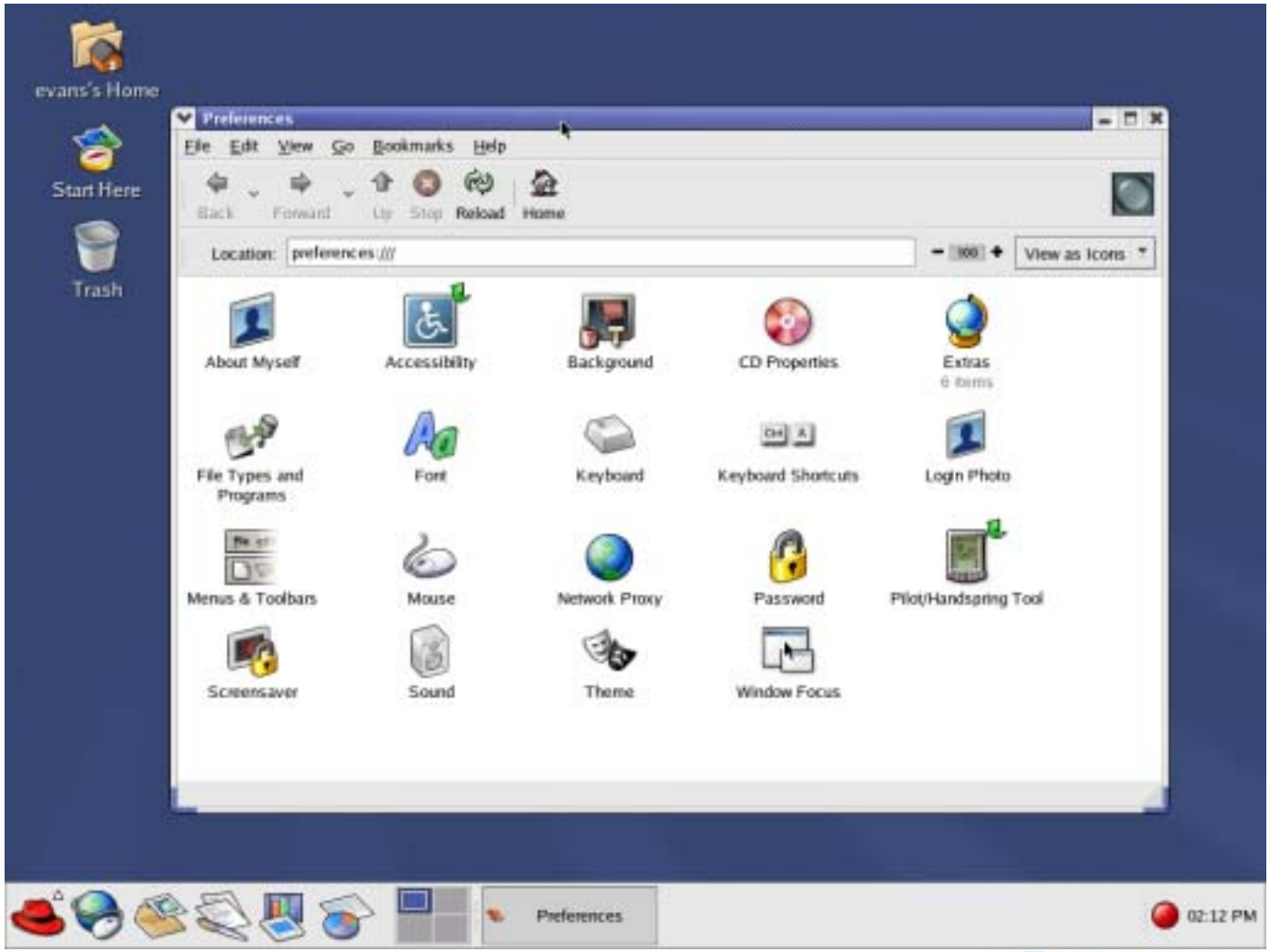
在RedHat默认的情况下，直接执行startx就可以进入GNOME的X环境。



lenovo 联想

GNOME桌面环境（续）

- GNOME配置工具（GNOME Control Center）



lenovo联想

GNOME桌面环境（续）

- GNOME配置工具（GNOME Configuration Tool）

GNOME配置工具允许用户对GNOME的外观和动作行为进行控制，这和Windows的“**Control Panel**”（控制面板）的做法很相似。要想启动GNOME的配置工具，可以单击控制条上看起来像是工具箱的图标；或是单击屏幕左下角的脚印图标，选择“ **Settings**”，并单击下一级菜单中的“**GNOME Configuration Tool**”。

可以使用GNOME配置工具进行配置的几种常见任务：

1、改变背景；

2、设置屏幕保护器；

3、改变桌面主题及风格；

4、更换窗口管理器等等

桌面环境小结

- X-Window环境“不”是操作系统的核心部分。
- 窗口管理器运行在X-Window顶层，我们可以选择最适合自己情况的窗口管理器。
- KDE环境是窗口管理器和（用于GUI应用程序开发的）应用程序框架的结合体。
- KDE环境的控制面板叫做**Control Center**（控制中心），可以单击窗口左下角的K图标后找到。
- GNOME为窗口管理器和函数库定义了一个应用程序框架。因此读者可以使用多种窗口管理器，比如**Enlightenment**和**Window Maker**等等。
- GNOME和KDE代表了当把Linux和UNIX作为一个整体来看时，在图形用户界面的质量方面的巨大进步。对这两种图形化操作环境的继续开发，使我们有足够的信心相信：会有更多的人从双重引导的系统转变到单引导的Linux操作系统上来。

第一部分 安装Linux操作系统作为服务器软件

- Linux发行版本与Windows NT的技术异同
- 按服务器配置安装Linux
- GNOME和KDE桌面环境
- 安装软件

安装软件

- 管理员作为系统管理的中心，避免不了需要安装提供各种服务所必须的软件，因此学习新软件包的安装机制和学习如何编译以源代码形式传播的软件包就是十分重要的了。
- 一般的情况下，大多数应用程序都有非常相似的安装操作模式。密切注意信息的主要来源，再加上一些常识，就可以使大多数的安装过程顺利进行。
- 最常用的两种软件安装方法：
  - **RedHat Package Manager (RPM)**
  - 自行编译源代码

RedHat Package Manager (RPM)

- **RedHat Package Manager（RPM）**软件包管理工具软件的基本功能是安装和清除文件，它使用起来很简便，许多Linux的发行版本都已经开始使用这个工具软件发行他们的软件。
- **RPM文件**：是能够让某个特定程序运行的全部文件的一个集合，它还包括对程序的说明、版本信息以及实现安装过程本身必须的脚本程序等。
- **RPM工具**：对安装在某个指定主机上的全部RPM软件包进行全面的管理的程序。管理包括已经安装了哪些软件包、它们的版本号码以及文件的存放位置等方面的记录。这些资料全部保存在主机上的一个简单的数据库文件中。
- 一般来说，RPM格式的软件要比需要编译的软件更容易安装和维护。因为使用RPM的时候用户也就接受了这个RPM中提供的缺省参数。大多数情况下，这些缺省参数使用起来都不会有什么问题。但是如果用户需要更具体地关心某项服务的实现过程，那么自行编译源代码将会使用户对软件包中都有哪些组件以及它们是如何协调工作的有更进一步的了解。

RPM（续）

- 安装新的软件包

  安装新软件包使用**-i**参数，例如：

```
[root@lenovo/root]# rpm –i bc-1.05a-4.i386.rpm
```

  升级已经存在的软件包使用**-U**参数，例如：

```
[root@lenovo/root]# rpm –U bc-1.05a-4.i386.rpm
```

RPM（续）

- RPM的一些附加命令行参数

命令行参数	说　明
---force	强行安装。典型的使用情况是：准备安装某个奇怪或者不寻常的配置，但是RPM软件的安全措施禁止继续操作。--force参数告诉RPM不做任何安全检查，只管安装就行。使用这个参数时一定要谨慎
-h	使用符号“#”指示安装进度，与-v参数一起使用时显示效果更好
--percent	显示已完成的百分比指示安装进度。如果从另外一个程序（比如某个Perl脚本程序）中来运行RPM，并且想了解安装的进度时，这个参数就很方便
-nodeps	如果RPM提示丢失了依赖文件，但是读者打算无论如何也要完成安装，那么在命令行上使用这个参数将使RPM不进行依赖关系检查
-test	这个参数并不进行真正的安装；它只是用来检查安装能否成功地完成。如果发现了问题，会把问题打印到显示器屏幕上
-v	告诉RPM报告每一步操作的情况

RPM（续）

- 查询软件包

有时候，了解系统中都已经安装了哪些软件包以及它们的用途是很有用的， RPM的查询参数就可以做到这一点。下面是一些例子：

```
[root@lenovo/root]# rpm -qa    //列出已经安装的全部软件包

[root@lenovo/root]# rpm -qf filename //找出某个特定文件属于哪个软件包

[root@lenovo/root]# rpm -qi packagename //了解某个已安装软件包的功
                                        //能
[root@lenovo/root]# rpm -qlp packagename //了解某个软件包中有哪些
                                         //文件
```

RPM（续）

- 卸载（清除）软件包

  要卸载软件包使用**-e**参数，格式为：

```
[root@lenovo/root]# rpm –e packagename
```

自行编译源代码

- 源代码开放软件的重要优点之一是用户手里已经有了源代码。如果程序员不再开发了，用户还可以继续弄到。如果发现问题了，用户就可以修补它。换句话说，是用户控制着形势，用不着再依靠某个用户无法控制的职业程序员了。但是有了源代码就意味着用户还必须能够对它进行编译，否则用户只不过是攥着一大把没什么用处的文本文件而已。

自行编译源代码（续）

- 获得并解压缩新的软件包

以源代码形式出现的软件通常都是一个“ **tarball**”文件，也就是说先是被归档为单独的一个大文件，再进行压缩。用来完成这个任务的是**tar**命令和**gzip**命令。tar命令负责把许多文件归档为单独的一个大文件，而gzip命令负责进行压缩。

典型情况下，人们通常都会选用一个独立完整的子目录对tarball文件进行制作和保存工作。这样在需要从其中取出什么东西的时候，系统管理员就可以把各个软件包的tarball文件保存到一个安全的位置。这还可以让系统管理员掌握除了基本的系统以外，机器里还安装有哪些软件包。**/usr/local/src**就是一个不错的子目录位置。

```
$ tar -zvxf tarballname  //解压缩的命令
```

命令中的z参数调用gzip命令进行解压缩操作，v参数让tar命令在解归档操作的同时显示被处理文件的文件名。

自行编译源代码（续）

- 查找软件包中的有关文档

  通常会包含**README**和**INSTALL**两个文件

  - **README文件**：包括了软件包的说明、附加文档的索引、软件包作者的联系方法等
  - **INSTALL文件**：是编译和安装这个软件包的操作指导

  有些软件包可能会有一些额外的文档，可以从它的文件名上进行猜测，如**SSH2.QUICKSTART**，**README.1ST**，**README.NOW**等等。

  额外资料的另外一个存放位置可能会是名为**“doc”**或者**“documentation”**之类的子目录

自行编译源代码（续）

- 配置新软件包

大多数软件包都会带有一个名为“**configure**”的执行自动配置操作的脚本程序，它通过大量可以设置的参数为软件包激活或禁止某些功能。查看软件包都有哪些配置选项，只需执行下面的命令：

```
$ ./configure –help
```

configure将建立**Makefile**，Makefile是编译阶段的基石。

- 编译新软件包

编译软件包是一个很简单的操作，执行**make**命令。

编译过程中的出错在大多数情况下并不是程序本身出现了问题，通常会是因为文件访问权限不正确或是文件未找到。如果确信不是这方面的问题，可以通过软件包中列出的E-mail地址向开发人员寻求帮助。

自行编译源代码（续）

- 安装新软件包

  安装软件包，只需在编译过程完全结束后，执行下面的命令：

  ```
  $ make install
  ```

  在这个过程中如果出现了错误信息，通常是因为访问权限的问题。检查出现问题时最后一个安装的文件，然后检查那里所有的访问权限要求，把需要的文件加上去。可能用到的命令：**chmod**、**chown**和**chgrp**。

- 安装完后的清理工作

  将安装过程中建立的所有临时文件都删除掉。

  因为拥有源代码tarball文件，所以全部删除编译工作的子目录是安全的。

安装软件小结

- 使用RPM进行安装的命令是 `# rpm -i packagename.rpm`
- 使用RPM进行升级的命令是 `# rpm -U packagename.rpm`
- 使用RPM删除已安装软件的命令是 `# rpm -e packagename.rpm`
- 大多数源代码都是“**tarball**”格式的文件，可以使用**tar**命令进行解压缩。
- 解压缩完成后，查阅软件包附带的文档是非常重要的。
- 配置软件包可以使用**./configure**命令。
- 编译软件包可以使用**make**命令。
- 安装编译好的软件可以使用**make install**命令。

第一部分　安装Linux操作系统作为服务器软件

第二部分　单主机系统的管理

第三部分　Internet网络服务

第四部分　内部网(Intranet)网络服务

第五部分　Linux操作系统的高级网络功能

lenovo 联想

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- 核心级系统服务
- 编译Linux内核
- 提高单个服务器的安全性

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- 核心级系统服务
- 编译Linux内核
- 提高单个服务器的安全性

用户的管理

- 基本概念
- 用户数据库
- 用户管理工具
- 使用命令行进行用户管理
- 使用redhat-config-users进行用户管理
- SetUID和SetGID程序
- 文件的所有权
- 小结

基本概念

- 在Linux操作系统中，任何东西都有一个所有者。也就是说，Linux系统中没有用户是无法存在的！最少它必须有一个根用户。
- 用户的账户通常包含以下内容：
  - **用户登录子目录**（home directory）

    保存用户专有的配置文件及日常工作文件
  - **口令**

    用户帐户必须要有口令才能登录系统；薄弱的口令会降低系统安全性
  - **shell**

    shell是用户在系统中的操作环境
  - **启动上机脚本程序**

    “命令脚本文件”和“点文件”
  - **电子邮件**

    每个用户都有一个基于他或她的登录名的电子邮箱

用户数据库

- Linux操作系统采用了UNIX传统的方法，把全部的用户信息保存为普通的文本文件。这样就允许你不必借助于其他的工具，只使用文本编辑器就可以对用户信息进行修改，十分简便。

- **/etc/passwd**

  /etc/passwd文件保存着用户的登录名、加过密的口令数据项、用户ID（UID）、缺省的用户分组ID（GID）、姓名（有时也叫做GECOS）、用户登录子目录以及登录后使用的shell。这个文件的每一行保存一个用户的资料，而用户资料的每一个数据项采用分号分隔开。如下所示：

  sshah：boQavhhaCKaXg：100：102：Steven Shah：/home/sshah：/bin/tcsh

用户数据库（续）

- **/etc/shadow**

家用电脑的速度开始让黑客们能够比较任意地对口令文件实现字典攻击，这就导致了把加过密的口令从**/etc/passwd**文件分离出去的做法。/etc/passwd依然保持对全部用户都可读，但是保存在/etc/shadow文件中的口令则只对那些具有根用户优先权的程序如登录程序等可读。/etc/shadow文件中每一行的格式包含着如下所示的几个部分：

- 登录名
- 加过密的口令
- 从1970年1月1日起计算，该口令修改后已经过去了多少天
- 需要再过多少天才能修改这个口令
- 需要再过多少天这个口令必须被修改
- 需要在这个口令失效之前多少天对用户发出提示警告
- 口令失效多少天之后禁用这个账户
- 从1970年1月1日起计算，该口令已经被禁用了多少天
- 保留域

用户数据库（续）

- **/etc/group**

每个用户至少会属于一个用户分组，也就是他缺省的用户分组。在需要的情况下，用户还可以分配到其他的分组中去。/etc/passwd文件中包含着每个用户缺省的分组ID(GID)。在/etc/group文件中，这个GID被映射到该用户分组的名称以及同一分组中的其他成员去。/etc/group文件中每一行的格式如下所示：

- 用户分组名
- 加过密的用户分组口令
- 用户分组ID号（GID）
- 以逗号分隔的成员用户清单

每一个数据域还是以冒号隔开的，其中的数据项看起来应该如下所示：

project：baHrE1KPNjrPE：102：sshah,hdc

用户管理工具

- 使用命令行进行用户管理

  useradd、userdel、usermod、groupadd、groupdel、groupmod

- 使用**redhat-config-users**进行用户管理

  在X-Window环境中使用交互式的图形界面

使用命令行进行用户管理

- **useradd**命令

  使用方法:

```
useradd [-c comment] [-d homedir] [-e expire date] [-f inactive time]
[-g initial group][-G group[,...]] [-m [-k skeleton dir]] [-M]
[-s shell] [-u uid [-o]] [-n] [-r] login
```

举例来说，如果打算建立这样一个用户：他的姓名是H.D.Core ，同时属于admin和support用户分组（缺省的用户分组是admin）、喜欢使用Turbo C Shell并希望使用登录名“hdc”，请使用下面这样的命令：

```
# useradd -c "H.D.core" -g admin -G sopport –s /bin/tcsh hdc
```

使用命令行进行用户管理（续）

- **useradd**命令（续）

参数及说明：

参　数	说　明
-c comment	允许你在GECOS域中设置用户的姓名。与其他命令行参数一样，如果它的设置值中有空格，就必须在文本两端加上引号——比如说，如果想把用户的姓名设置为Steven Shah，就必须把参数定义为-c “Steven Shah”
-d homedir	缺省的情况下，用户的登录子目录被定义为/home/login。这样如果我的登录名是sshah，我的登录子目录就是/home/shah。在建立一个新用户的时候，该用户的登录子目录是和用户账户一起建立的。因此，如果你想把缺省值修改为另外一个位置，就可以定义新的位置——比如说，-d /home/sysadm/sshah
-e expire-date	在经过了一段时期之后，账户有可能失效。缺省的情况下，账户永远都不会失效。如果你想指定一个日期，一定要按照MM/DD/YY的格式进行（本系统中，对2000年请使用00做为设置值）——使用-e 04/01/00设置为在2000年4月1日失效
-f inactive-time	定义口令失效后，新账户还能够使用的天数。0（零）值表示要立刻禁用这个账户。-1值表示即使在口令失效之后也不禁用这个帐户——比如说：-f 3允许每个用户在口令失效还可以再使用3天。这个参数的缺省值是-1
-g initial group	使用这个参数可以在口令文件中定义用户缺省的分组。你可以使用那个分组的一个编号或者名称；但如果你使用的是每个用户分组的名称，那么这个用户分组必须已经在/etc/group进行了定义——比如-g project

使用命令行进行用户管理（续）

- **useradd**命令（续）

参数及说明（续）：

-G group [, . . .]	允许你把新用户设置到其他的分组中去。如果你使用了 -G参数，就必须至少指定一个额外的用户分组。另外，你还可以使用逗号分隔定义多个用户分组（举例来说，如果你打算把用户加到 project和admin用户分组中去，就可以使用 -G projetc, admin）
-m [-k skel-dir]	缺省情况下，系统将自动地建立用户的登录子目录。这个参数明确地建立用户的登录子目录名称。建立这个子目录的部分工作是把缺省的配置文件拷贝到这个子目录中去。这些文件缺省情况下是从 /etc/skel子目录中拷贝过去的。使用第 2个参数-k skel dir可以改变缺省设置（你必须使用了 -m参数才能使用-k参数）。举例来说，如果想指定/etc/adminskel子目录，我们需要使用-m -k /etc/adminskel
-M	如果已经使用了-m参数，就不能再使用 -M参数，反之亦然。这个参数告诉系统不要建立用户的登录子目录
-n	Red Hat Linux把建立一个与新用户同名的新用户分组作为用户添加过程的一部分。使用这个参数可以禁止这种行为
-s shell	用户的登录shell是一个用户登录进入一个系统之后运行的第一个程序。它通常是一个命令行操作环境，除非你是从 X-Windows登录屏幕登录的。缺省情况下，它将是Bourne Shell (/bin/bash)。有些人喜欢使用其他的shell如Turbo C Shell (/bin/tcsh)。这个参数可以让你随意选择新用户登录之后运行的 shell（shell的清单保存在/etc/shells文件中）
-u uid	缺省的情况下，程序将会自动地找出下一个可用的 UID并使用它。如果出于某种原因你需要强制让某个新用户的 UID是一个特殊的数值，则可以使用这个参数。请记住对全部用户来说，他们各自的 UID必须是唯一的
Login	最后，唯一“不是”可选项的参数！你必须指定新用户的登录名

使用命令行进行用户管理（续）

- **userdel命令**

  使用方法:

  ```
  # userdel [ -r ]  username
  ```

  在执行这个命令的时候，如果只指定了用户的登录名–比如说userdel sshah，那么在/etc/passwd文件和/etc/shadow文件中的有关数据项以及/etc/group文件中的关联数据项都将被自动删除。如果使用了可选参数，比如说userdel -r sshah，那么在其登录子目录中归这个用户所有的全部文件也将被删除。

使用命令行进行用户管理（续）

- **usermod命令**

使用方法：

```
usermod [-c comment] [-d homedir] [-m] [-e expire date]
[-f inactive time] [-g initial group]
[-G group[,...]] [-l login] [-s shell]
[-u uid] login
```

-l参数允许你改变用户的登录名，它和**-u**参数在使用中必须引起足够的重视。在修改用户的登录名或者UID的时候，必须确认该用户当时没有登录上机或运行任何进程。如果在用户已经登录上机或正在运行进程的时候修改这些信息会引起不可预见的结果。

下面是一个使用usermod命令对用户hdc进行修改的例子，我们打算把她的姓名域由“ H.D.H”修改为“H.D.Core”。

```
# usermod -c " H.D.Core " hdc
```

使用命令行进行用户管理（续）

- **groupadd命令**

使用方法：

```
# groupadd [-g gid] [-r] [-f] group
```

参数及说明：

参　数	说　明
-g *gid*	把新用户分组的GID指定为gid。缺省的情况下，这个值被自动选定为找到的第一个可用值
-r	缺省情况下，Red Hat会自动搜索第一个大于499的GID值。-r参数告诉groupadd命令正在添加的用户分组是一个系统分组，需要使用第一个小于499的可用数值
-f	在添加新用户分组的时候，如果准备添加的用户分组已经存在，Red Hat Linux会自动退出执行，并且没有错误信息。使用这个参数，在退出执行之前，这个命令不会修改用户分组的设置值。在进行命令脚本程序编程的时候，如果你打算在用户分组已经存在时让命令脚本程序继续执行下去，这个参数就非常有用了
group	这个参数是必需的。它定义了你打算添加的用户分组名称为 group

使用命令行进行用户管理（续）

- **groupdel**命令

  使用方法:

  ```
  # groupdel groupname
  ```

- **groupmod**命令

  使用方法：

  ```
  # groupmod –g gid –n group-name group
  ```

  其中的**-g**参数允许改变用户分组的GID值，**-n**参数允许你给用户分组起一个新名字，当然还需要把现有用户分组的名称作为最后一个参数。

使用redhat-config-users进行用户管理

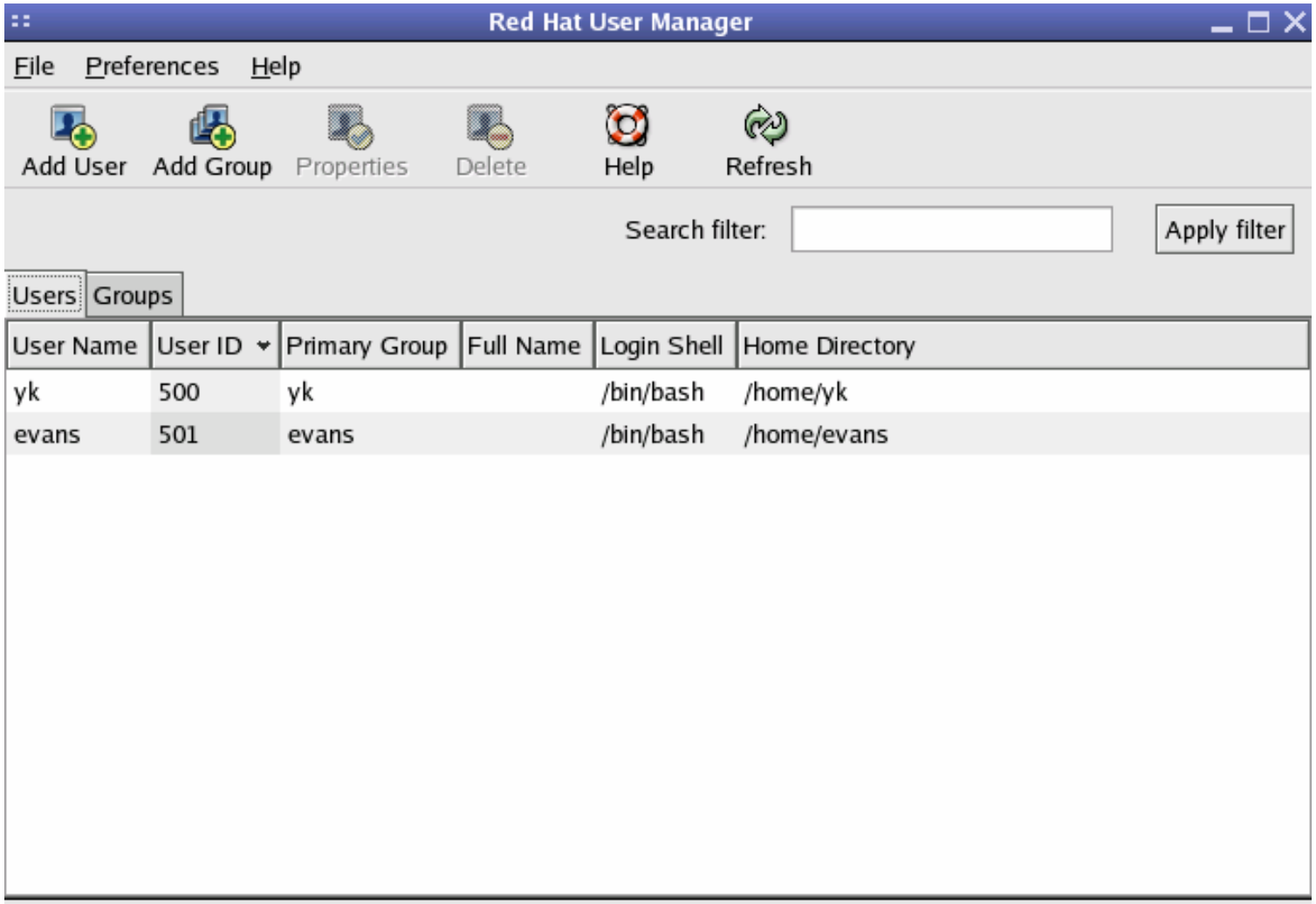
- redhat-config-users工具是一个很有用的配置工具，它是由redhat公司提供的redhat-config-系列配置工具中专门用于管理用户帐号的。
- 启动redhat-config-users之前，必须以根用户身份登录进入系统并运行X-Window环境。然后从某个终端窗口中输入

  # redhat-config-users

  命令启动redhat-config-users。

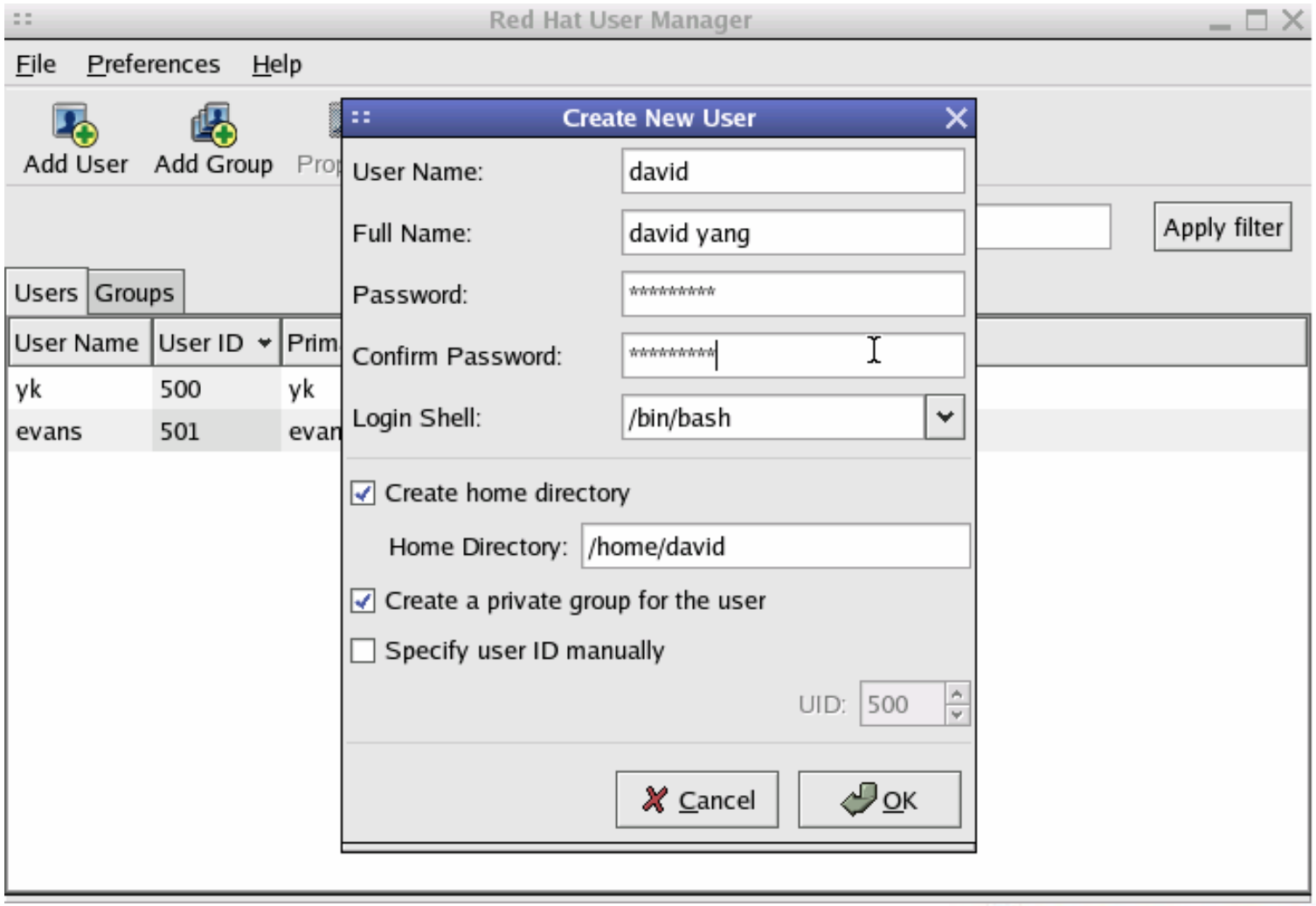
使用redhat-config-users进行用户管理

- 启动redhat-config-users后的窗口:



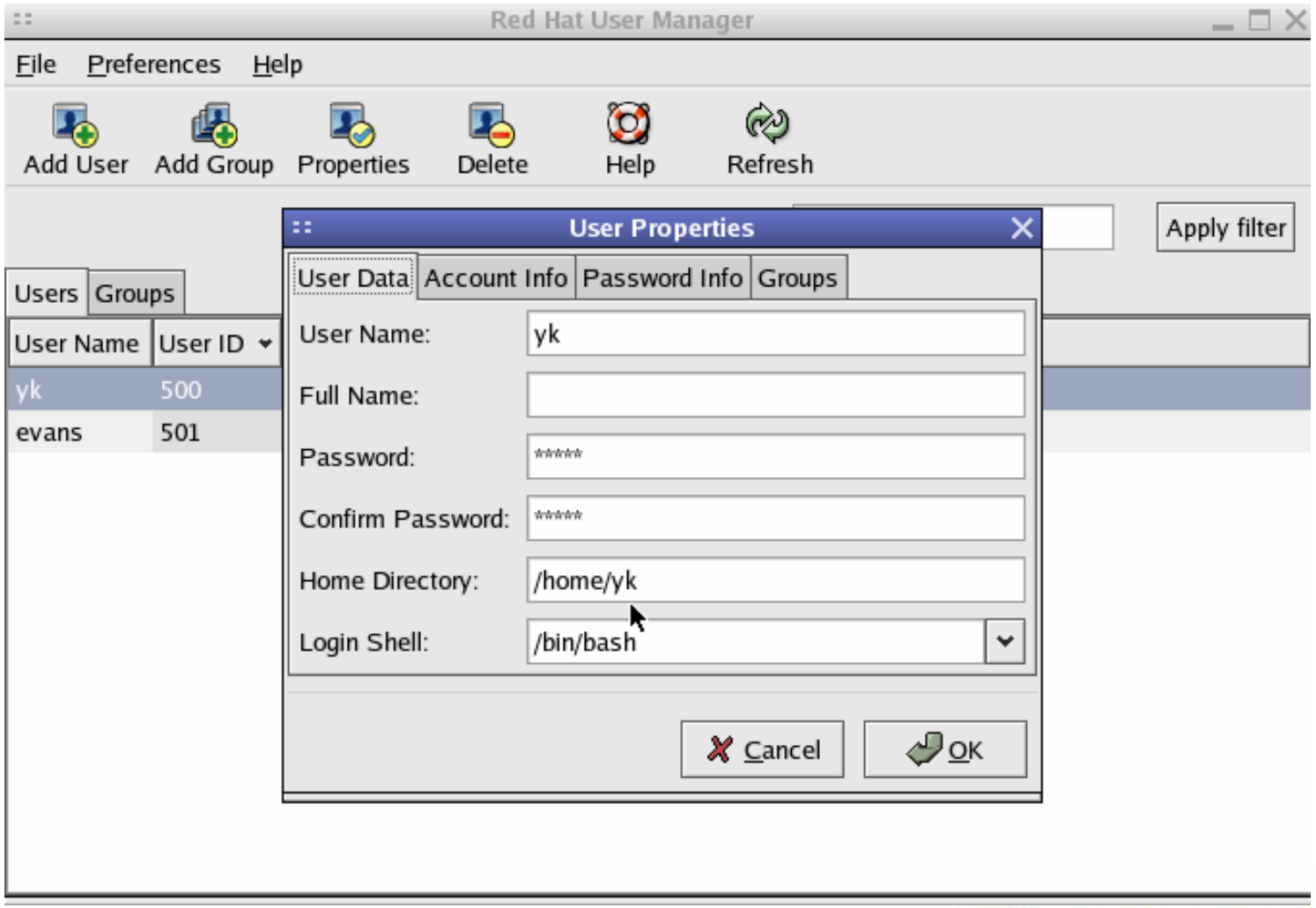
使用redhat-config-users进行用户管理

- 添加一个用户：



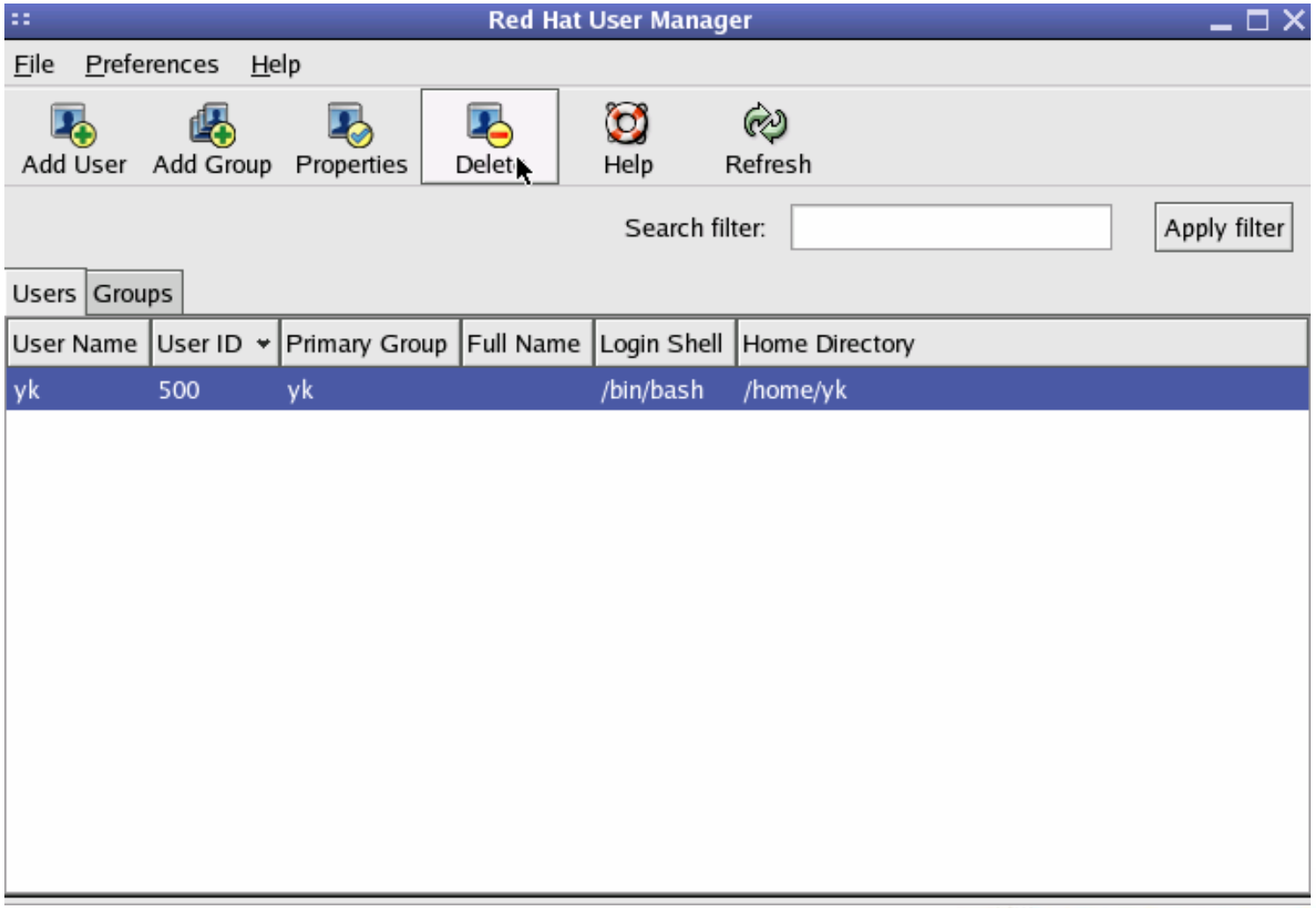
使用redhat-config-users进行用户管理

- 修改用户信息：



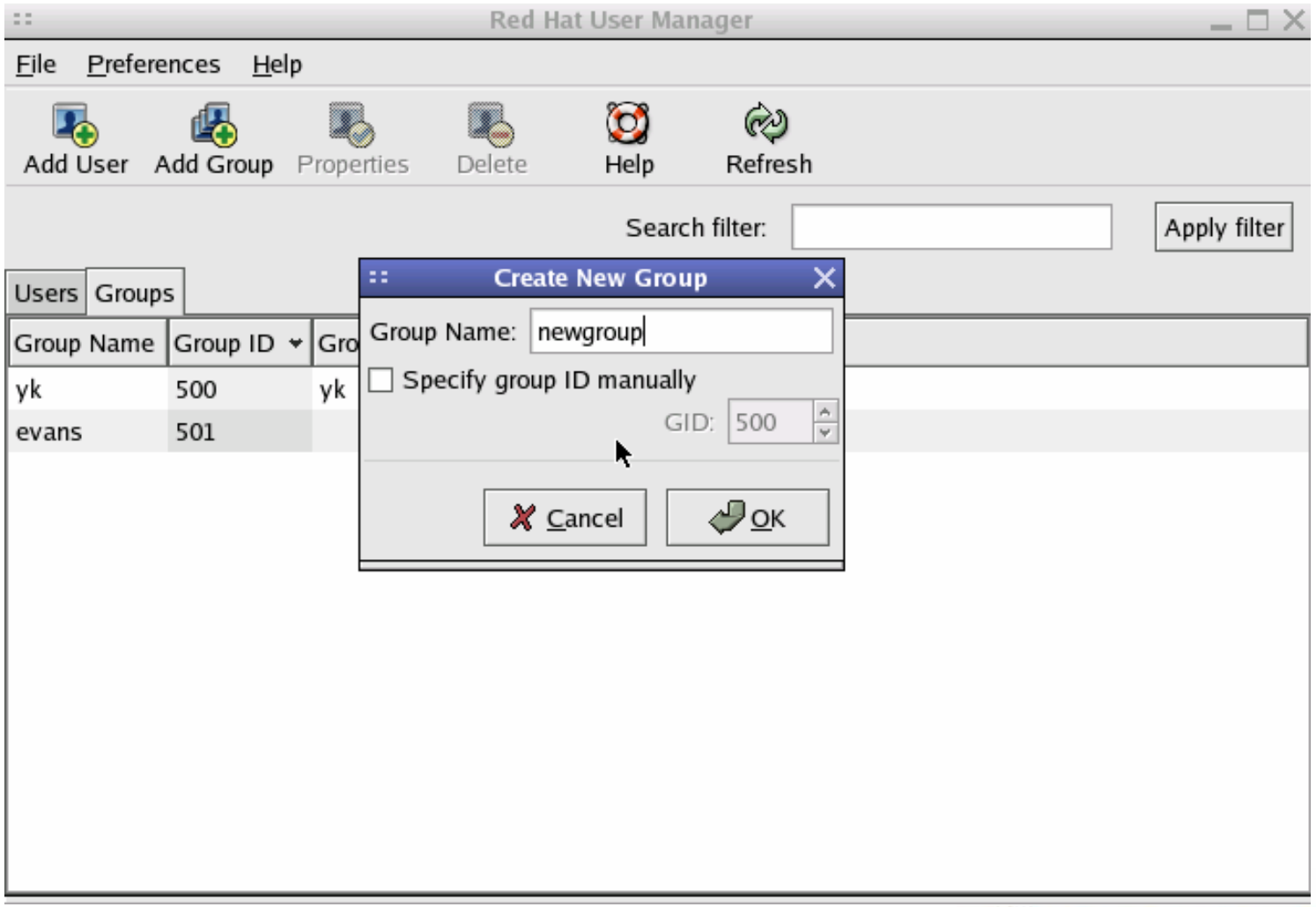
使用redhat-config-users进行用户管理

- 删除用户帐号：



使用redhat-config-users进行用户管理

- 添加用户分组：



lenovo联想

SetUID和SetGID程序

- 一般情况下，当用户运行一个应用程序的时候，这个程序将继承该用户所具有的全部权利（或者限制）。如果用户不能够读取**/var/log/messages**文件，那么他运行的程序也不能读取该文件。请注意这个权限可能会与该程序文件（通常叫做二进制文件）所有者所具有的权限有所不同。
- 但是，如果我们对**SetUID**或是**SetGID**的二进制位进行设置，使程序按照其所有者或文件分组的访问权限运行，不再受运行它的用户的访问权限的限制。
- 可以使用**chmod**命令激活SetUID或者SetGID位。
- 如果想把某个程序设置为SetUID状态，在打算分配给它的访问权限数值前面加上一个数字**4**。如果想把某个程序设置为SetGID状态，在打算分配给它的访问权限数值前面加上一个数字**2**。

文件的所有权

- 当一个用户被建立之后，它就得到了一个新的、唯一的UID值。该用户建立的任何文件都归这个用户所有。出于简单化的考虑， Linux并没有使用用户名而是使用了UID来设置文件的所有权。然后系统使用/etc/passwd文件在用户的UID和登录名之间做一个映射，这样就使子目录列表操作更容易阅读。
- 那么当用户从/etc/passwd文件中被删除后但是属于他的文件还依然存在的时候会发生什么样的事情呢？实际上不会发生什么事情。我们最能够观察到的现象将发生在问题中的文件进行子目录列表操作的时候。列表中不会出现文件的所有者，它将显示为一个号码。这个号码代表着拥有该文件的UID。如果有一个新用户在被建立的时候使用了与老用户相同的UID，这个相同的UID将被显示为所有者，使得新用户看起来就像是拥有着那些文件一样。因为会出现这种情况，所以当删除某个用户的账号时，确保同时删除了该用户拥有的全部文件是十分重要的。

用户管理小结

- 我们讨论过的重要内容主要有以下几个方面：
  - 每一个用户都将获得唯一的**UID**
  - 每一个用户分组都将获得唯一的**GID**
  - **/etc/passwd**文件把UID映射到用户名
  - Linux操作系统对加密口令有多种处理方法
  - Linux操作系统附带有帮助你对用户进行管理的工具软件

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- 核心级系统服务
- 编译Linux内核
- 提高单个服务器的安全性

命令行

- 命令行参数使得UNIX系统高效而又灵活，但是这种能力也使得UNIX对普通用户来说不容易理解和掌握。因此，对许多UNIX工具软件来说，图形化的用户界面（GUI）已经成为举足轻重的标准装备。
- 但是那些经验比较丰富的用户发现GUI工具很难提供全部的可选参数。而要想提供完备的参数，一般又会使操作界面变得与其对应的命令行程序同样复杂。GUI的设计思路就是要简化操作，因此有经验的用户基本都会返回到命令行灵活的能力上去。
- 每一种界面都有它各自的缺点与优点，而最终结果是，选择掌握两种方法的人将走在前面。
- 本章我们将深入介绍那些对日常工作最为关键的工具软件。

命令行（续）

- Bash简介
- 文档工具
- 文件列表、所有权和访问权限
- 文件管理和操作
- 进程管理
- 其他工具

Bash简介

- shell只不过是一个程序，提供了一个到系统的操作界面。具体到BASH shell（Bourne Again），它是一个只提供了命令行操作的界面，包括许多内建命令，具有启动其他程序和控制从它启动的程序的能力（作业控制）。

- 我们将仔细研究一些BASH的内建命令。关于BASH的完整介绍本身就能够很容易地写成一本书，因此我们将重点放在对系统管理员的日常工作影响最大的那些命令上。

Bash简介（续）

- 作业控制

  “&”符号：将作业放入后台运行

  CTRL+Z：暂停作业并放入后台

  jobs：查看后台进程（运行及暂停的）

  bg number：让后台进程继续运行

  fg number：将作业调回前台运行

- 环境变量

  查看环境变量：**printenv**

  设置环境变量：**export variable=value**

  取消已设置的环境变量：**unset variable**

Bash简介（续）

- 管道

  管道的机理：通过它可以把一个程序的输出发送到另一个程序作为输入，各自独立的程序可以一环连一环的组成功能极为强大的工具

- 重定向

  输出到文件：**>**

  附加到文件末尾： **>>**

  发送文件作为输入： **<**

- BASH的命令行快捷键

  可以使用**星号(*)**和**问号(?)**作为通配符

  可以使用环境变量作为命令行上的参数

  可以让多个命令在同一命令行上执行，命令间用**分号(;)**隔开

  可以使用**反单引号(`)**把一个命令的输出作为另一个命令的参数

文档工具

- Linux操作系统带有两个对阅读文档非常有用的工具程序：**man**和**info**。如今，这两种文档系统之间有许多转换工具，因为许多应用程序都在把它们的文档转换为info格式。这种格式是优于man的，因为它允许文档以类似于互连网应用的形式超连接在一起，但是又不必真的编写成HTML格式。
- 从另外一个角度看，man格式已经存在几十年了。有数以千计的软件以man使用手册页作为它们唯一的文档。进一步说，还有许多应用程序继续使用man格式，因为许多其他的UNIX族操作系统（比如Sun Solaris）使用着man格式。

文档工具（续）

- **man**

命令格式： `$ man program_name`

手册页的章节排列顺序：

man分类号	主题	man分类号	主题
1	用户工具	5	配置文件
2	系统调用	6	游戏
3	C函数库调用	7	软件包
4	设备驱动程序信息	8	系统工具

窍门　man命令有一个方便的参数-k，用在作为参数的命令前面。这个参数让man命令从全部的使用手册页中查找总结性的有关资料，并且把匹配了你指定命令的那些使用手册页、以及它们的小节序号列出来，如下所示：

```
[root@lenovo/root]# man -k printf
```

文档工具（续）

- **info**

命令格式： `$ info program_name`

文档的另外一种常见的格式是**texinfo**格式。作为GNU组织建立的标准， texinfo的文档系统与World Wide Web国际互连网上超链接的格式很类似。由于文档可以超链接在一起，texinfo通常都比较容易阅读、使用和检索。

文件列表、所有权和访问权限

- **ls**

ls命令列出一个子目录中的全部文件。它有26个命令行参数，下面列出来的是它最常用的几个。这些参数可以任意地组合使用。完整的命令行参数清单请阅读其texinfo文档。

ls命令的参数	说　明
-l	长列表。除了文件名之外，还列出文件的大小、日期 /时间、访问权限、所有者以及用户分组信息
-a	全部文件。列出该子目录中所有的文件，包括隐藏文件
-1	单列列表
-R	递归地列出所有的文件和下级子目录

使用长列表方式列出某个子目录中的全部文件，使用下面的命令：

```
[root@lenovo root]# ls -la
```

列出子目录中以字母A打头的全部非隐藏文件，使用下面的命令：

```
[root@lenovo root]# ls A*
```

文件列表、所有权和访问权限（续）

- 文件和子目录类型

普通文件－保存着数据和可执行程序，而操作系统本身对它们中的内容不做任何规定

子目录－子目录文件是普通文件的一种特殊形态。子目录文件中保存的是其他文件的存放位置。

硬链接(hard link)－Linux文件系统中的每一个文件都有它自己的i-结点（i-node）。每个i-结点都保存了一个文件的属性和它在硬盘上的位置。如果你需要使用两个不同的文件名代表同一个文件的话，就可以建立一个硬链接。硬链接具有与原始文件同样的i-结点，因此其外部表现和内部行为都和原始文件一模一样。每建立一个硬链接，就给“链接计数器”增加一个值；每删除一个硬链接，链接计数器就减去一个值。在链接计数器的值减到零之前，这个文件不会从硬盘上真正被删除。

文件列表、所有权和访问权限（续）

- 文件和子目录类型（续）

符号链接(symbolic link)－与通过i-结点指向某个文件的硬链接不一样，符号链接是通过文件名指向另外一个文件的。这就允许符号链接指向位于其他分区、甚至是其他网络硬盘上的某个文件。

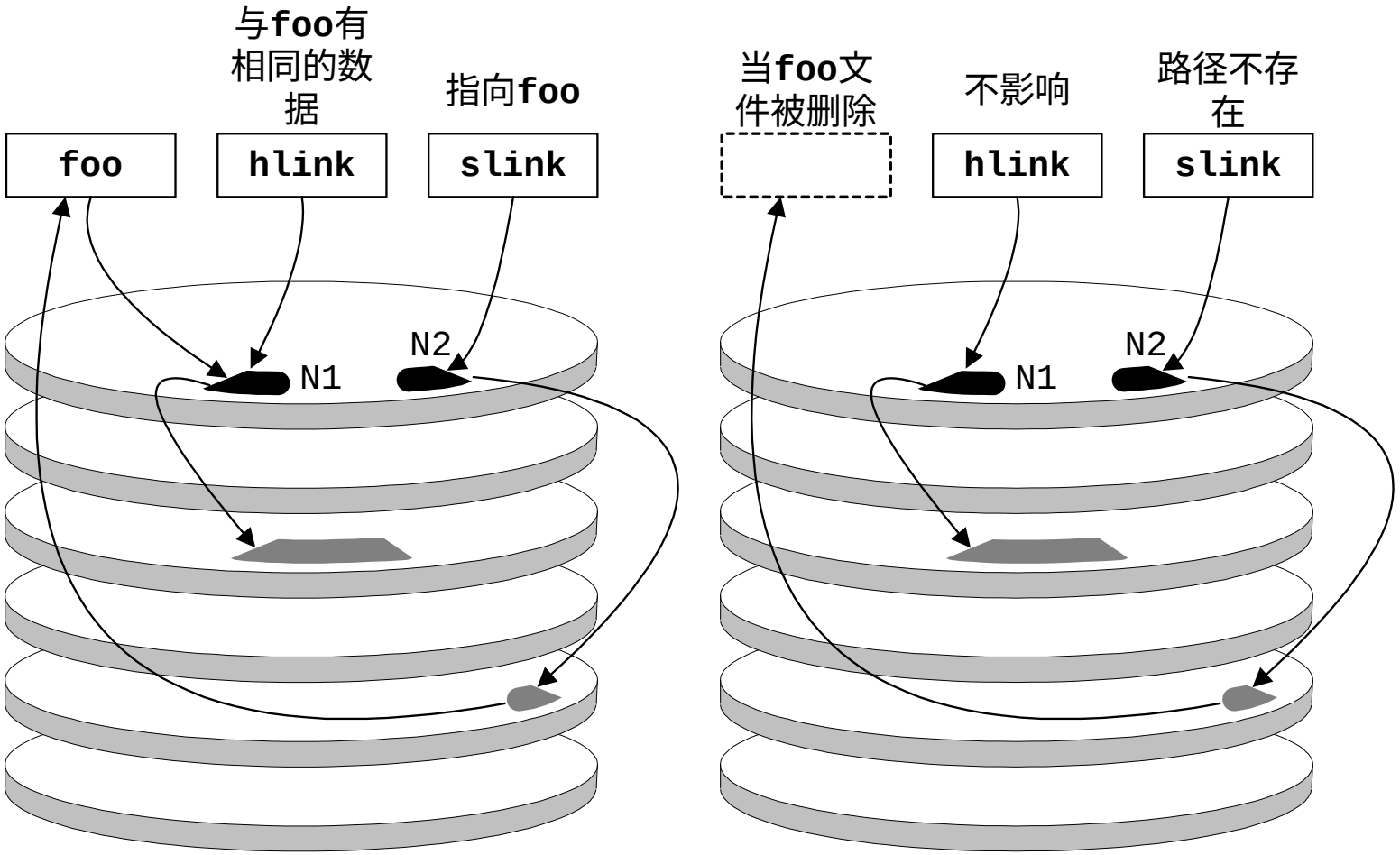
块设备－各种设备驱动程序都是通过文件系统进行访问的，块设备类型的文件被用做磁盘之类设备的接口。一个块设备文件有三个辨认特征：它有一个**主号码**、一个**从号码**，另外当使用ls -l命令列出它的时候，其**访问权限的头一个字母是b**。如下所示：

```
[root@lenovo /root]# ls -l /dev/hda
brw-rw----   1 root   disk     3,   0  May 5 2003 /dev/hda
```

请注意上面的列表中文件访问权限的头一个字母是b；3是主号码、0是从号码。

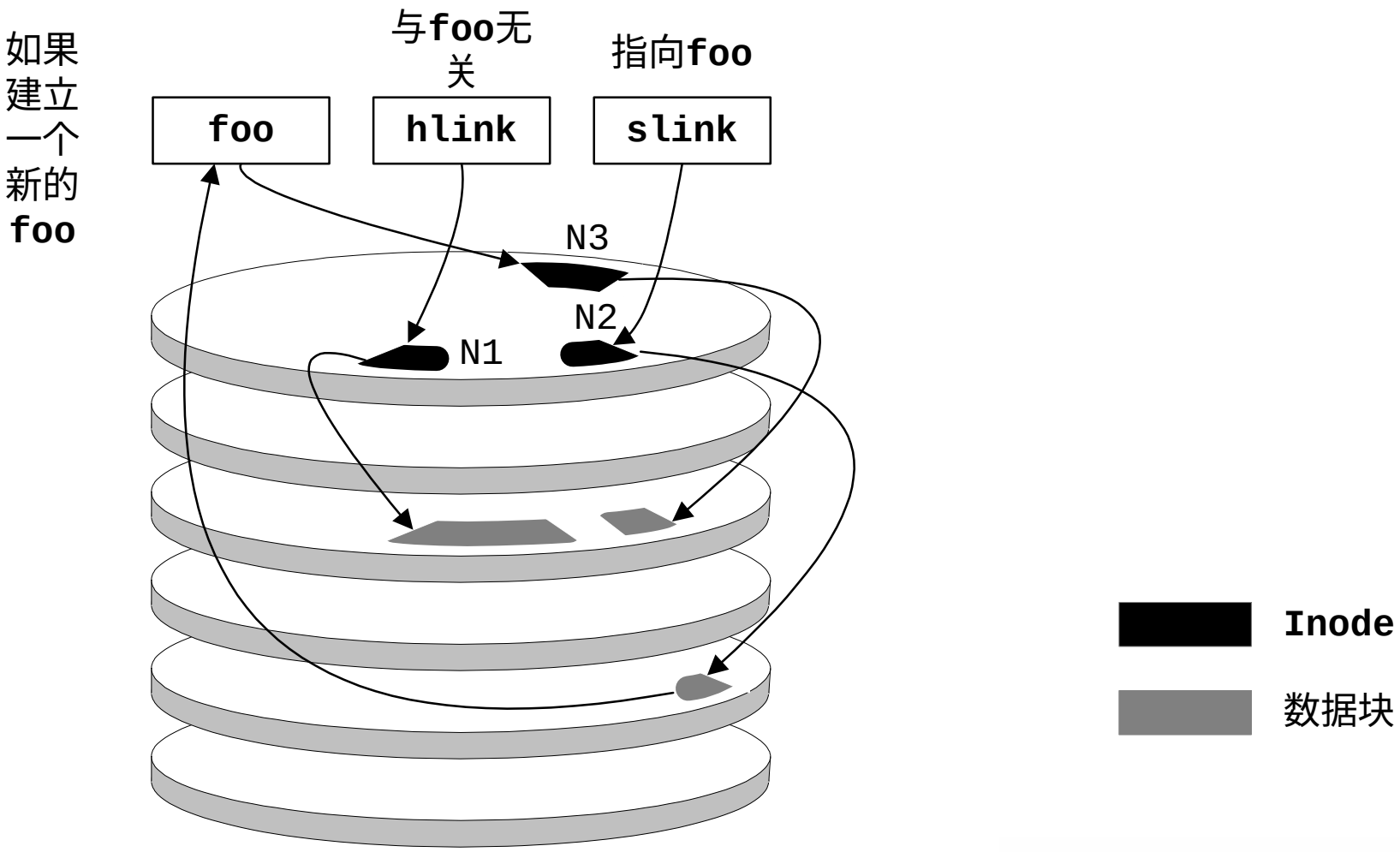
文件列表、所有权和访问权限（续）

文件foo有一个硬连接和一个软连接



lenovo 联想

文件列表、所有权和访问权限（续）



文件列表、所有权和访问权限（续）

- 文件和子目录类型（续）

  字符设备－与块设备类似，字符设备也是允许通过文件系统访问硬件设备的特殊文件。块设备和字符设备最明显的区别就是：**块设备与真实硬件设备之间的数据交换是以数据块为单位的**；而**字符设备每次只能交换一个字符的数据**（硬盘是一个块设备；调制解调器是一个字符设备）。字符设备的访问权限的**头一个字母是c**，并且文件具有一个**主号码**和一个**从号码**，如下所示：

```
[root@lenovo /root]# ls –l /dev/ttyS0
crw-------   1 root   tty    4,  64  May 5 2003 /dev/ttyS0
```

文件列表、所有权和访问权限（续）

- 文件和子目录类型（续）

命名管道－命名管道也是一种特殊的文件类型，它用来在进程之间进行通信。使用**mknod**命令就可以建立一个命名管道文件，一个进程可以打开它进行读操作，而另外一个进程则可以打开它进行写操作，这样就可以让这两个进程彼此进行通信。如果某个程序拒绝命令行管道读取的输入，另外一个程序又必须对这个程序输出数据，而磁盘上又没有足够临时文件使用的空间，在这种情况下，命名管道正好管用。一个命名管道文件的访问权限的**头一个字母是p**。如下所示：

```
[root@lenovo /root]# ls –l mypipe
prw-r--r--   1 root   root          0  May 5 10:47 mypipe
```

文件列表、所有权和访问权限（续）

- 改变文件的所有权命令**chown**

  命令格式： `$ chown [-R] username filename`

  chown命令可以把一个文件的所有权修改为别人的，只有根用户能够进行这样的操作。当指定的filename是一个子目录的名字的时候，就需要使用-R参数。这个参数告诉命令要递归地逐层进入这个子目录的树状结构，把新的所有权分配给这个子目录本身和其中全部的文件及下级子目录。

- 改变文件的用户分组命令**chgrp**

  命令格式： `$ chgrp [-R] groupname filename`

  chgrp与chown命令很像,用法上差不多。

文件列表、所有权和访问权限（续）

- 改变文件属性命令**chmod**

  访问权限分为四个部分：

  - 第一个部分就是访问权限的头一个字母。“普通”文件不具有任何特殊的值，头一个字母就使用一个短划线（-）字符。如果该文件具有特殊的属性，它就用一个字母来表示。我们在这里最感兴趣的两种特殊属性是子目录（d）和符号链接（l）。
  - 访问权限的第二、三、四部分是三个字符一组的数据段。第一个部分表示的是文件所有者的访问权限；第二个部分表示的是文件所在分组的访问权限；最后一个部分表示的是全系统的访问权限。在UNIX操作系统的术语里，“全系统”意味着系统中的全部用户，不管他们的分组设置情况如何。

文件列表、所有权和访问权限（续）

- 改变文件属性命令**chmod**（续）

代表访问权限的字母以及它们对应的数值。当你组合属性的时候，把它们的数值逐个相加。chmod命令用来设置访问权限的数值

r	**w**	**x**
读	写	执行
4	2	1

三种访问权限最常见的组合形式

---	0	无访问权限
r--	4	只读
rw-	6	读和写
rwx	7	读，写，执行
--x	1	只能执行

文件列表、所有权和访问权限（续）

- 改变文件属性命令**chmod**（续）

  对每一个文件来说，三个这样的三字符组组合在一起。第一个字符组表示的是文件所有者的访问权限；第二个字符组表示的是文件所在分组的访问权限；最后一个字符组表示的是系统中全部用户的访问权限。下面给出了一些常见的文件访问权限设置值。

（见下页）

文件列表、所有权和访问权限（续）

- 改变文件属性命令**chmod**（续）

访问权限	对应数值	说　明
-rw-------	600	所有者拥有读和写权限，大多数文件都有这个设置
-rw-r--r--	644	所有者拥有读和写权限；用户分组和全系统拥有只读权限。要确定你真的想让其他人读取这个文件
-rw-rw-rw-	666	人人都拥有读和写权限。不推荐这种做法，因为此组合允许任何人从系统中的任何地点访问该文件
-rwx------	700	所有者拥有读、写和执行权限。对所有者打算运行的程序来说（从C或者C++程序编译而来的结果文件），这是最佳的访问权限组合
-rwxr-xr-x	755	所有者拥有读、写和执行权限。其他人拥有读和执行权限
-rwxrwxrwx	777	任何人都拥有读、写和执行权限。和设置值666一样，这个组合必须避免使用
-rwx--x--x	711	所有者拥有读、写和执行权限。其他人只拥有执行权限。适用于打算让其他人执行但不想让他们拷贝的程序
drwx------	700	这是一个使用mkdir命令建立的子目录。只有所有者能够在这个子目录里进行读写操作。请注意：所有的子目录必须设置执行位x
drwxr-xr-x	755	这个子目录只能够由所有者进行改动，但是其他人可以查看它的内容
drwx--x--x	711	让子目录对全系统可读，但是限制使用ls命令的访问。只有那些知道其名字的人才能对子目录中的文件进行读操作

文件管理和操作

- 拷贝文件命令**cp**

  cp命令用来拷贝文件，它有数量相当多的参数。详细资料请查阅它的使用手册页。下面是cp命令最常见的参数：

  -f 强制性拷贝，不要求确认

  -i 交互式拷贝，在每一个文件被拷贝之前，要求用户确认

- 移动文件命令**mv**

  mv命令用来把文件从一个位置移动到另外一个位置。文件也可以从一个分区移动到另外一个分区。这会引起拷贝操作，这样文件移动命令的用时可能会长一点。下面是mv命令最常见的参数：

  -f 强制性移动

  -i 交互式移动

文件管理和操作（续）

- 链接文件：**ln**命令

ln命令用来建立硬链接和软链接。

ln命令的一般格式如下所示：

```
$ ln original_file new_file
```

ln命令最常用的参数是**-s**，建立一个符号链接而不是一个硬链接。

例子：建立一个符号链接，让**/usr/bin/myadduser**指向**/usr/local/bin/myadduser**，使用下面的命令：

```
$ ln -s /usr/local/bin/myadduser /usr/bin/myadduser
```

文件管理和操作（续）

- 查找文件命令**find**: find start_dir [option]

  find命令可以根据各种检索条件查找文件。部分参数如下：

find命令的参数	说　明
-mount	不搜索与搜索起点不同的文件系统
-atime *n*	至少在n*24小时以内没有访问过的文件
-ctime *n*	至少在n*24小时以内没有修改过的文件
-inum *n*	拥有i-结点值为n的文件
-amin *n*	n分钟之前访问过的文件
-cmin *n*	n分钟之前修改过的文件
-empty	文件为空
-mmin *n*	n分钟之前修改过的文件
-mtime *n*	n小时之前修改过的文件
-nouser	文件的UID值在/etc/passwd文件中没有对应的真正用户
-nogroup	文件的GID值在/etc/group文件中没有对应的真正用户分组
-perm *mode*	文件的访问权限被准确设置为mode
-size *n[bck]*	文件的长度至少为n块/字符/千字节。每块等于512字节
-print	列印找到的文件名
-exec *cmd* \;	对每一个找到的文件，运行cmd命令。重要事项：每一个cmd的后面必须跟上“\;”符号，要不然BASH会不知道如何继续操作
-name name	文件的名字必须是name。这里可以使用规则表达式

文件管理和操作（续）

- 转换并拷贝文件命令**dd**

dd命令读出一个文件的内容再把它送到另外一个文件中去。它与cp命令的不同之处在于：dd命令可以即时进行文件格式转换，还可以从磁带或者软驱甚至其他设备上接受数据。当dd命令访问每个设备的时候，它不对文件系统进行任何假设，只是把设备上的数据原样读下来。

如果想生成一个软盘的映像（特别适用于另类的文件格式），请使用下面的命令：

```
# dd if=/dev/fd0 of=/tmp/floppy_image
```

dd命令的参数	说　明
if = *infile*	把输入文件指定为infile
of = *outfile*	把输出文件指定为outfile
count = *blocks*	设定dd命令在退出之前至少应该对blocks个数据块进行操作
ibs = *size*	把输入设备的数据块长度设置为size
obs = *size*	把输出设备的数据块长度设置为size

文件管理和操作（续）

- 文件压缩命令**gzip**

在UNIX最早的发行版本里，用来进行文件压缩的工具程序叫做**compress**。但不幸的是这个算法被某些想利用它发大财的人申请了专利。为了避免付费，大多数站点寻找到另外一个非专利算法的压缩工具程序gzip。经常与gzip一起使用的可选参数如下所示：

-c 把压缩后的文件写到stdout标准输出（这样可以使输出经过管道送入其他程序）

-d 解压缩

-r 递归找出全部需要压缩的文件

-9 最大压缩效果

-l 最快压缩时间

gzip是“当场”对文件进行压缩的，也就是说在压缩操作结束之后，原始文件将被删除，剩下的就只有压缩好的文件了。

文件管理和操作（续）

- 建立子目录命令**mkdir**

  Linux中的mkdir命令与其他UNIX操作系统中同名的命令是完全一样的，与MS-DOS中的也一样。它唯一的参数是**-p**，使用这个参数可以在没有上级子目录的情况下完成操作。

  如果想建立一个名为mydir的子目录，请使用下面的命令：

  ```
  $ mkdir mydir
  ```

- 删除子目录命令**rmdir**

  rmdir命令对那些熟悉DOS命令的人们毫不陌生；它用来删除某个现有的子目录。它唯一的命令行参数是**-p**，使用这个参数可以把上级子目录一起删除掉。

- 显示当前工作子目录命令**pwd**

  只有两个参数：**--help,--version**

文件管理和操作（续）

- 磁带文件归档命令**tar**

  tar程序把多个文件合并成一个大文件。它独立于压缩工具程序，因此可以让你选择使用哪一种压缩工具或者是否需要使用压缩。另外，tar还可以与dd命令以几乎同样的方法对设备进行读写操作，这使得tar成为备份磁带设备的良好工具。

  下面是tar命令的结构、它最常用的参数和几个用法示例：

  ```
  $ tar [ commands and options ] filename
  ```

tar命令的参数	说　明
-c	建立一个新的档案文件
-t	查看档案文件的内容
-x	释放档案文件的内容
-f	定义档案文件所在文件（或者设备）的名字
-v	操作过程中显示流程信息
-z	假设该文件已经（或者将要）使用gzip进行压缩

文件管理和操作（续）

- 磁带文件归档命令**tar**（示例）
  - 建立一个包含/usr/src/apache子目录中全部文件的名为apache.tar的档案文件

```
$ tar -cf apache.tar /usr/src/apache
```

  - 建立一个包含/usr/src/apache子目录中全部文件的名为apache.tar的档案文件，并且在操作过程中显示流程信息

```
$ tar -cvf apache.tar /usr/src/apache
```

  - 建立一个包含/usr/src/apache子目录中全部文件的经过gzip压缩的名为apache.tar.gz的档案文件，并且在操作过程中显示流程信息

```
$ tar -cvzf apache.tar.gz /usr/src/apache
```

  - 释放一个名为apache.tar.gz的经过gzip压缩的tar档案文件，并且在操作过程中显示流程信息

```
$ tar -xvzf apache.tar.gz
```

文件管理和操作（续）

- 合并文件命令**cat**

  cat程序通常被用来显示文件信息，它也可以用来把许多文件合并成一个独立连续的大文件。

- 分屏显示文件命令**more**

  more命令执行的情况与DOS中的同名程序差不多是一样的。它使用一个输入文件，以每次一屏的方式显示其内容。输入文件可以来自它的标准输入**stdin**，也可以来自命令行参数。

- 磁盘操作工具命令**du**

  du命令能够检查子目录的磁盘工作性能。

文件管理和操作（续）

- 查找文件位置命令**which**

  which命令在用户的全部路径中对在它命令行上给出的文件进行查找，这个命令用来查找文件所在子目录的完整路径。

- 查找命令的保存位置命令**whereis**

  whereis命令搜索用户的路径，给出程序的名称和它所在的子目录、源文件（如果有的话）、以及该命令的使用手册页。

- 得到磁盘空间命令**df**

  df程序给出每一个分区自由空间的总量。为了得到这些信息，欲对之操作的硬盘驱动器或者分区必须已经挂装在系统上。NFS网络文件系统的信息也可以采用这个方法收集到。

文件管理和操作（续）

- 同步磁盘命令**sync**

与大多数其他现代的操作系统一样，Linux使用了磁盘缓存技术以提高性能。这样做的缺点是：你**打算写到磁盘上的东西并不都被及时地写到磁盘上**。

如果想把缓存中的内容写到磁盘上，则需要使用sync命令。如果sync检测到已经安排了把缓存中的内容写到磁盘上，内核就会立刻清空缓存。这个命令没有命令行参数。

如果想确保缓存中的内容立刻被写到磁盘上，请使用下面的命令：

```
# sync ; sync
```

进程管理

- 对Linux操作系统（以及UNIX操作系统）来说，每个运行中的程序至少由一个进程组成。从操作系统的立场出发，每个进程与其他进程都是彼此独立的。除非某个进程发出与其他进程共享资源的特殊请求，一般情况下它是被局限在分配给它的内存和CPU位置上的。跨出其分配内存的进程（它们可能会引起另外一个运行程序的崩溃并使系统不稳定）将立刻被终止。管理进程的这个方法对UNIX系统的稳定性起了非常大的作用：一个用户的应用程序不会干扰到其他用户的程序或者操作系统本身。
- 下面将介绍列出和管理进程的工具程序。对一个系统管理员的日常工作来说，它们是非常重要的元素。

进程管理（续）

- 列出进程清单命令**ps**

  ps命令列出系统中全部的进程，包括他们的状态、大小、名称、所有者、CPU时间、已运行时间等方面的信息。它有许多命令行参数，下表列出了其中最常用的一些。

ps命令的参数	说　明
-a	列出带有控制终端的全部进程，不仅仅是当前用户的进程
-r	只列出正在运行中的进程（请参考本小节后面对进程状态的说明）
-x	列出没有控制终端的那些进程
-u	列出进程的所有者
-f	给出进程之间的父/子关系
-l	按长格式显示清单
-w	显示进程的命令行参数（最多半行）
-ww	显示进程的全部命令行参数，不管其长度是多少

进程管理（续）

- 列出进程清单命令**ps**（续）

ps命令最常用的参数组合是**-auxww**。这些参数将列出全部的进程（、每个进程的所有者、以及进程全部的命令行参数。我们来看看一个**ps -auxww**的部分输出结果。

```
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3   1096   476 ?        S    Jun10   0:04 init
root         2  0.0  0.0      0     0 ?        SW   Jun10   0:00 [kflushd]
root         3  0.0  0.0      0     0 ?        SW   Jun10   0:00 [kpiod]
root         4  0.0  0.0      0     0 ?        SW   Jun10   0:00 [kswapd]
root         5  0.0  0.0      0     0 ?        SW<  Jun10   0:00 [mdrecoveryd]
root       102  0.0  0.2   1068   380 ?        S    Jun10   0:00 /usr/sbin/apmd -p 10 -w 5
bin        253  0.0  0.2   1088   288 ?        S    Jun10   0:00 portmap
root       300  0.0  0.4   1272   548 ?        S    Jun10   0:00 syslogd -m 0
root       311  0.0  0.5   1376   668 ?        S    Jun10   0:00 klogd
daemon     325  0.0  0.2   1112   284 ?        S    Jun10   0:00 /usr/sbin/atd
root       339  0.0  0.4   1284   532 ?        S    Jun10   0:00 crond
root       357  0.0  0.3   1232   508 ?        S    Jun10   0:00 inetd
root       371  0.0  1.1   2528  1424 ?        S    Jun10   0:00 named
root       385  0.0  0.4   1284   516 ?        S    Jun10   0:00 lpd
root       399  0.0  0.8   2384  1116 ?        S    Jun10   0:00 httpd
xfs        429  0.0  0.7   1988   908 ?        S    Jun10   0:00 xfs
root       467  0.0  0.2   1060   384 tty2     S    Jun10   0:00 /sbin/mingetty tty2
root       468  0.0  0.2   1060   384 tty3     S    Jun10   0:00 /sbin/mingetty tty3
```

进程管理（续）

- 列出进程清单命令**ps**（续）

输出结果的第一行给出了清单的内容标题，它们是：

- **USER** 谁拥有这个进程
- **PID** 进程的标识号码
- **%CPU** 进程占用CPU的百分比。对一个多处理器系统来说，这一行数字相加的结果可能会大于100%
- **%MEM** 进程占用内存的百分比
- **VSZ** 进程占用虚拟内存的总量
- **RSS** 进程占用真实（驻留）内存的总量
- **TTY** 进程的控制终端。在这一列中出现的问号（?）意味着那个进程不再与某个控制终端相关连

进程管理（续）

- 列出进程清单命令**ps**（续）
  - **STAT**　　进程的状态

S 进程休眠中。所有准备运行的进程（即那些被安排为多任务的进程，但是CPU当前正在处理其他事情）都是休眠状态的

R CPU正在处理的进程

D 不可中断休眠状态（通常与输入输出有关）

T 正在被纠错程序跟踪或者已经被终止的进程

Z “昏迷”的进程

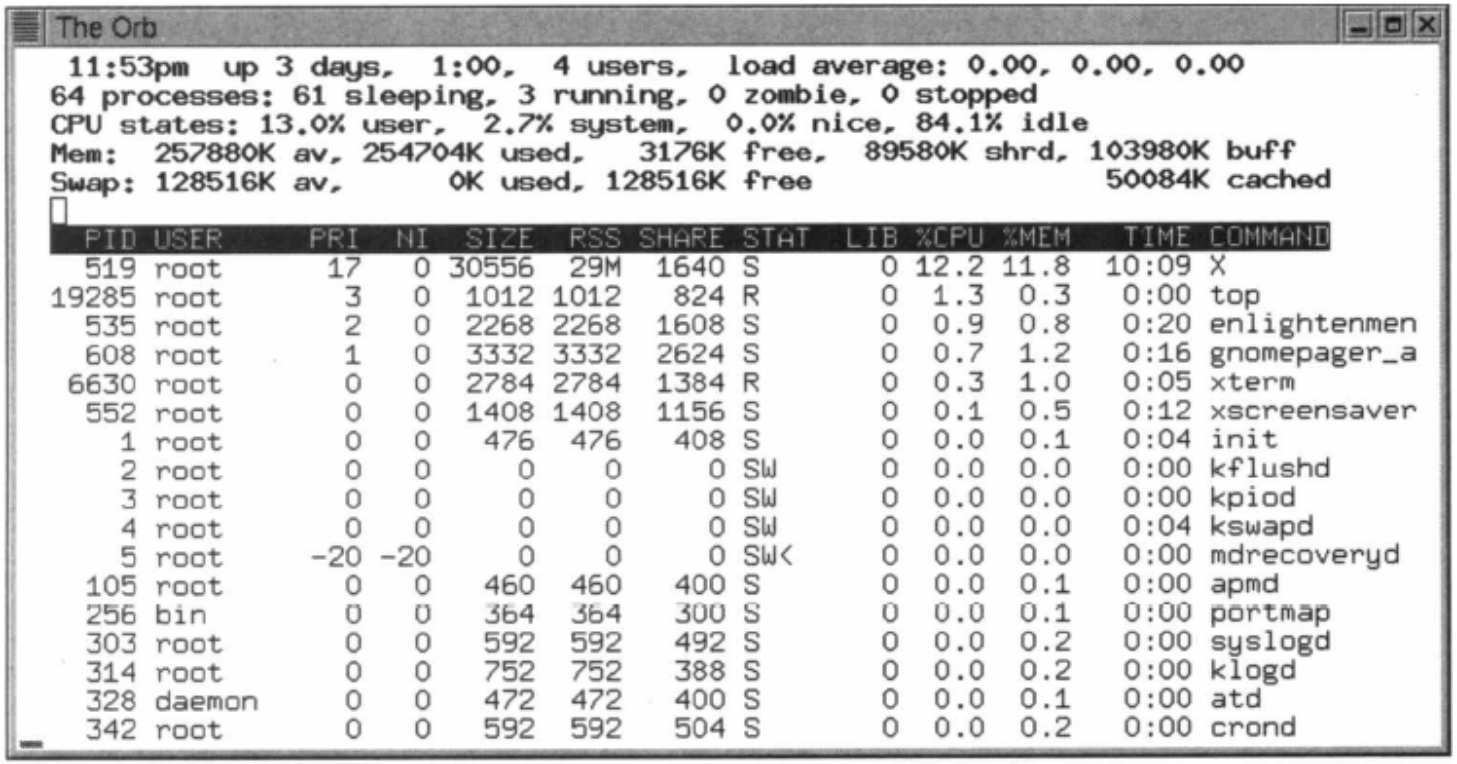
另外，每个进程的STAT数据项还可以有如下所示的说明符： W = 内存中没有驻留页面（它已经全部交换出内存）； < = 高优先权进程； N = 低优先权进程； L = 内存页面被锁定在那里（通常就表示需要实时操作功能）

  - **START**　　进程开始的时间
  - **TIME**　　进程已经使用的C P U时间
  - **COMMAND**　　进程名称和它的命令行参数

进程管理（续）

- 交互列出进程清单命令**top**

top命令是一个交互式操作的ps命令版本，top命令每隔2-3秒钟（用户可调）就会刷新进程清单的显示画面。top命令的严重不足是它会占用CPU。在一个拥挤的系统上，这个程序会使系统管理方面的问题复杂化。



进程管理（续）

- 向某个进程发送消息命令**kill**

这个程序的名字有一些误导：其实它并不真的“杀死”进程。它的作用是向正在运行的进程发送消息。

kill命令的可选参数是**-n**，其中的n是信号的编号。作为一名系统管理员，我们最感兴趣的是信号**9**（中断程序运行）和**1**（挂起）。

能够中断进程运行的能力明显是强大的，根用户可以对系统中的全部进程发送信号，这意味着在使用kill命令的时候，根用户必须非常谨慎。

下面是kill命令可以发送的消息列表：

```
# kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGIOT       7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR
```

其他工具

- 显示系统名称命令**uname**

uname程序给出一些系统细节信息，在某些场合是很有帮助的。下面是uname命令的命令行参数：

uname命令的参数	说　明
-m	给出机器的硬件类型（比如“i686”表示Pentium Pro或者更好的体系结构）
-n	给出机器的主机名
-r	给出操作系统的发行名称
-s	给出操作系统的名称
-v	给出操作系统的版本
-a	列出以上全部的信息资料

- 查看用户命令**who**

在允许用户登录进入到其他用户的计算机或者特殊目的的服务器主机上的系统时，系统管理员需要了解都有哪些人正在上机。可以使用who命令产生这样一个报告。

其他工具（续）

- 改变用户身份命令**su**

  su命令可以用来切换用户身份，而不需要退出再重登录。

- 编辑器程序

  vi编辑器

  emacs编辑器

  joe编辑器

  pico编辑器

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- **开机和关机**
- 文件系统
- 核心级系统服务
- 编译Linux内核
- 提高单个服务器的安全性

开机和关机

- 操作系统正在变得越来越复杂，开机引导和关机下电的过程也越来越智能化。从简单的DOS系统转移到Windows NT系统的人们已经亲身感受到了这些变化—这已不仅仅是核心操作系统的启动引导和关闭了，还包括必须要同时启动或者关闭相当数量的服务项目。类似于Windows NT，Linux系统作为启动过程的一个组成部分需要打开的服务项目也是数量极大的。
- 下面我们将逐步介绍Linux操作系统环境的启动和关闭过程。我们将讨论使这个过程能够自动进行的脚本程序，以及这个过程的哪些部分可以被修改。

开机和关机

- LILO
- 添加新内核
- 开机引导的步骤

LILO

- **LILO是LI**nux **LO**ader(Linux加载程序)的缩写，是一个引导管理程序。它可以引导多个操作系统，还可以选择引导时使用不同的内核配置。
- **LILO的概念**：准备一个配置文件（**/etc/lilo.conf**），定义哪一个硬盘分区是可以引导的。当程序**/sbin/lilo**开始运行的时候会记下这个分区信息，然后把必要的数据再写到引导扇区上，按配置文件中的定义提供引导参数。

配置LILO

LILO的配置文件是/etc/lilo.conf

```
boot=/dev/had
prompt
timeout=50
image=/boot/vmlinuz-2.2.5-15
        label=linux
        root=/dev/hda2
        read-only
other=/dev/hda1
        label=dos
        table=/dev/hda
```

附加的LILO参数

- 1、lilo.conf文件中的全局参数

  全局参数的作用范围是整个配置文件，不仅局限于某个特定的数据块

default = *name*	定义将要引导的缺省操作系统。如果没有此语句，第一个数据块就被认为是缺省值
message = *message-file*	“lilo :”提示符出现之前可以在屏幕上先显示一段信息（它的内容保存在message-file文件里）。这段信息的长度不得超过64K；而且如果它发生了变化，映射文件也要重建（请参考后面的“运行 LILO”一节）
prompt	强制LILO显示“lilo :”提示符并等待用户作出响应。如果定义了这个参数但是没有定义timeout参数，重启动就无法自动进行
timeout = *deciseconds*	以十分之一秒为单位定义一个倒计时数字，如果在倒计时期间提示符下没有输入，则使用缺省的引导映象继续完成引导过程

附加的LILO参数（续）

- 2、数据块参数

  数据块参数的作用范围仅限于它们所在的数据块，不同的数据块定义了不同的引导操作系统

image = *image file*	定义引导Linux操作系统的内核映象文件
other = *image file*	定义引导其他各种操作系统的映象文件
table = *device name*	说明本数据块的硬盘分区表保存在哪一个设备上
label = *name*	定义所在数据块的名称，当用户在“lilo:”提示符下查看引导菜单时，看到的就是这条语句定义的名称
password = *password*	表示只有当用户正确地输入了口令后才能引导本数据块中定义的操作系统
restricted	如果有命令行参数需要传递到内核，用户必须输入正确的口令（当需要口令来保护单用户模式的操作环境时，这个参数尤其重要）

附加的LILO参数（续）

- 3、内核参数

  参数可以从LILO传到内核。其中包括请求引导至单用户模式的参数。这些参数的作用范围仅限于Linux内核。

append = string	把string添加到用户输入的每一个命令行参数后面
literal = string	类似于append，但是string并不是附加到用户通过命令行传递的参数后面，而是完全代替之
read-only	定义根文件系统需要挂装为只读属性。在内核加载完成并使用 fsck工具程序对根文件系统进行检查之后，再把它重新挂装为读/写属性

添加引导用的新内核

第一步：编译好的新内核**vmlinuz-2.3.12**保存在**/boot**子目录中，把有关的信息添加到**/etc/lilo.conf**文件中去。但是如果只是把这个数据块添加到**/etc/lilo.conf**的末尾，那末**vmlinuz-2.3.12**还不是缺省引导的内核。若想让它成为缺省引导的内核，有两个办法：

一、把这个数据块移动到第一个数据块的位置

```
image=/boot/vmlinuz-2.3.12
    label=linux-2.3.12
    root=/dev/hda2
    read-only
```

二、使用default命令

```
default = vmlinuz-2.3.12
```

添加引导用的新内核（续）

最终的/etc/lilo.conf文件的内容为：

```
default=vmlinuz-2.3.12
boot=/dev/had
prompt
timeout=50
image=/boot/vmlinuz-1.2.5-15
        label=linux
        root=/dev/hda2
```

添加引导用的新内核（续）

运行LILO，大多数情况下，直接运行LILO：

```
[root@lenovo /boot] # lilo
Added linux *
Added dos
```

LILO命令的几个重要参数

LILO命令的参数	说　明
-t	对配置情况进行测试，但是不真正加载。如果单用 -t参数并不会告诉你很多东西，但是如果和 -v参数（见本表下面的定义）一起使用，就可以看到LILO到底做了些什么
-C *config-file*	缺省情况下，LILO会查找/etc/lilo.conf作为其配置文件。使用这个命令行参数，你可以为它另外指定一个配置文件
-r *root-directory*	通知LILO在开始任何操作之前使用chroot命令把根用户目录切换到指定的子目录去。chroot命令会把根用户目录切换到 root-directory定义的子目录去。这个参数的典型用法是通过软盘引导开机，以便修复发生故障的系统（举例来说，如果你是从软盘引导开机的，并且把根文件系统挂装在/mnt子目录下，就可能需要使用lilo -r /mnt命令来运行LILO）
-v	让LILO详细报告它执行的每一步操作

开机引导的步骤

- **加载内核**：LILO启动以后，若选择Linux作为准备引导的操作系统，第一个被加载的就是内核。此时计算机**内存**中还不存在任何操作系统；而PC还没有办法存取机器上全部的内存。因此，内核就必须完整的加载到可用RAM的第一个兆字节之内。内核被压缩了。
- **执行内核**：内核在内存中解压缩后就可以运行了。内存必须有足够的代码设置它自己的虚拟内存子系统和根文件系统。
- **init进程**：它是非内核进程中第一个被启动运行的；PID是1。init读它的配置文件**/etc/inittab**，决定它需要启动的运行级别。

rc命令脚本程序

- 需要管理的服务器很多，因此需要使用**rc**命令脚本程序。最主要的一个是**/etc/rc.d/rc**，它负责为每一个运行级别按照正确的顺序调用相应的命令脚本程序。
- 两种方法实现修改的目的：
  - 一、如果所做的修改只在引导开机的时候起作用，并且改动不大的话，可以简单的编辑一下**/etc/rc.d/rc.local**脚本
  - 二、如果修改很细致或要求关闭进程的操作必须使之明确的停止运行，需要在**/etc/rc.d/init.d**子目录中添加一个命令脚本程序。这个命令脚本程序必须可以接受**start**和**stop**参数并完成相应的操作。

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- 核心级系统服务
- 编译Linux内核
- 提高单个服务器的安全性

文件系统

- 虚拟文件系统
- 文件系统的构成
- 管理文件系统
- 对硬盘进行分区
- 建立文件系统
- 网络文件系统
- 使用Automounter自动挂装子系统
- 硬盘空间配额的管理

虚拟文件系统

- 文件系统分类
  - 基于磁盘的文件系统
    - Linux ext2/ext3/Reiser
    - SYSV fs/BSD UFS/minix/VxFS
    - MSDOS/VFAT/NTFS
    - iso9660/UDF(DVD)
    - HPFS/HFS(Mac)/AFFS
    - JFS/XFS

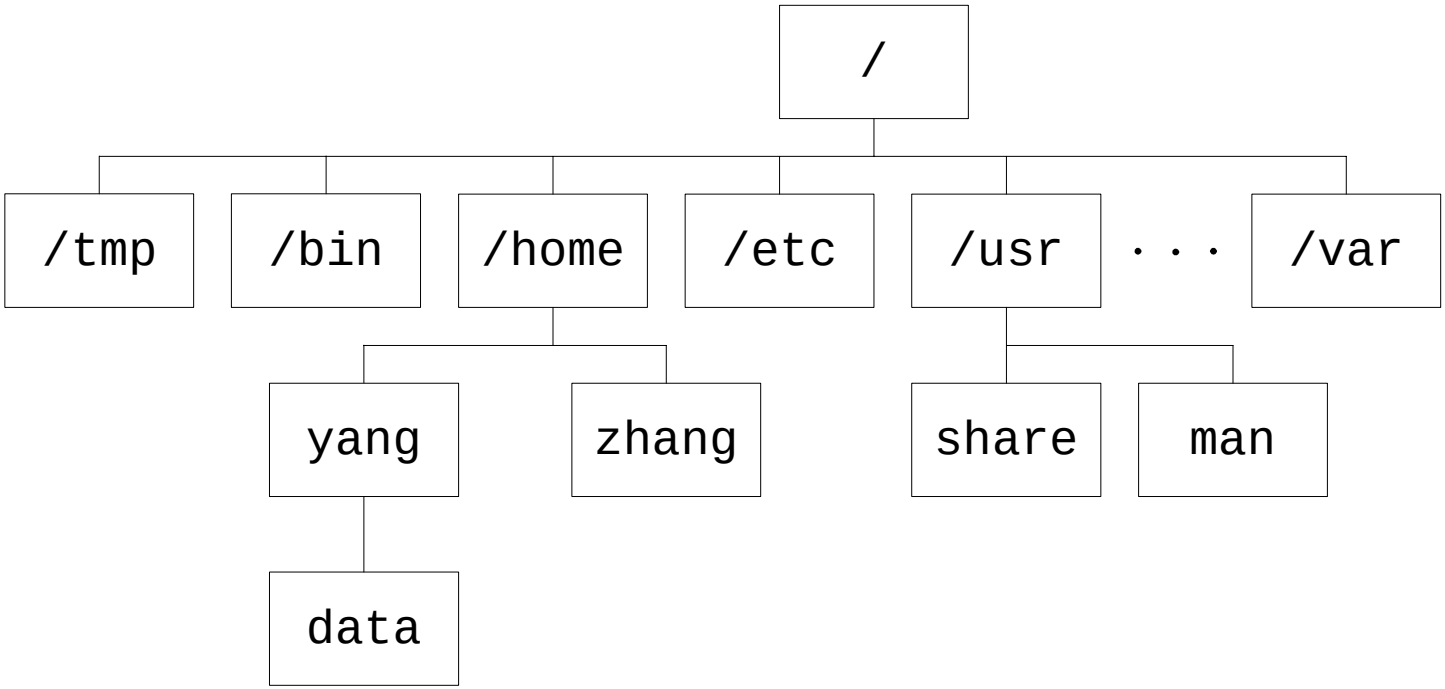
虚拟文件系统（续）

- 文件系统分类（续）
  - 网络文件系统
    - NFS
    - AFS/Coda
    - SMB/CIFS
    - NCP
  - 特殊文件系统
    - /proc
    - Devfs
    - Sock fs
    - Tmpfs

虚拟文件系统（续）

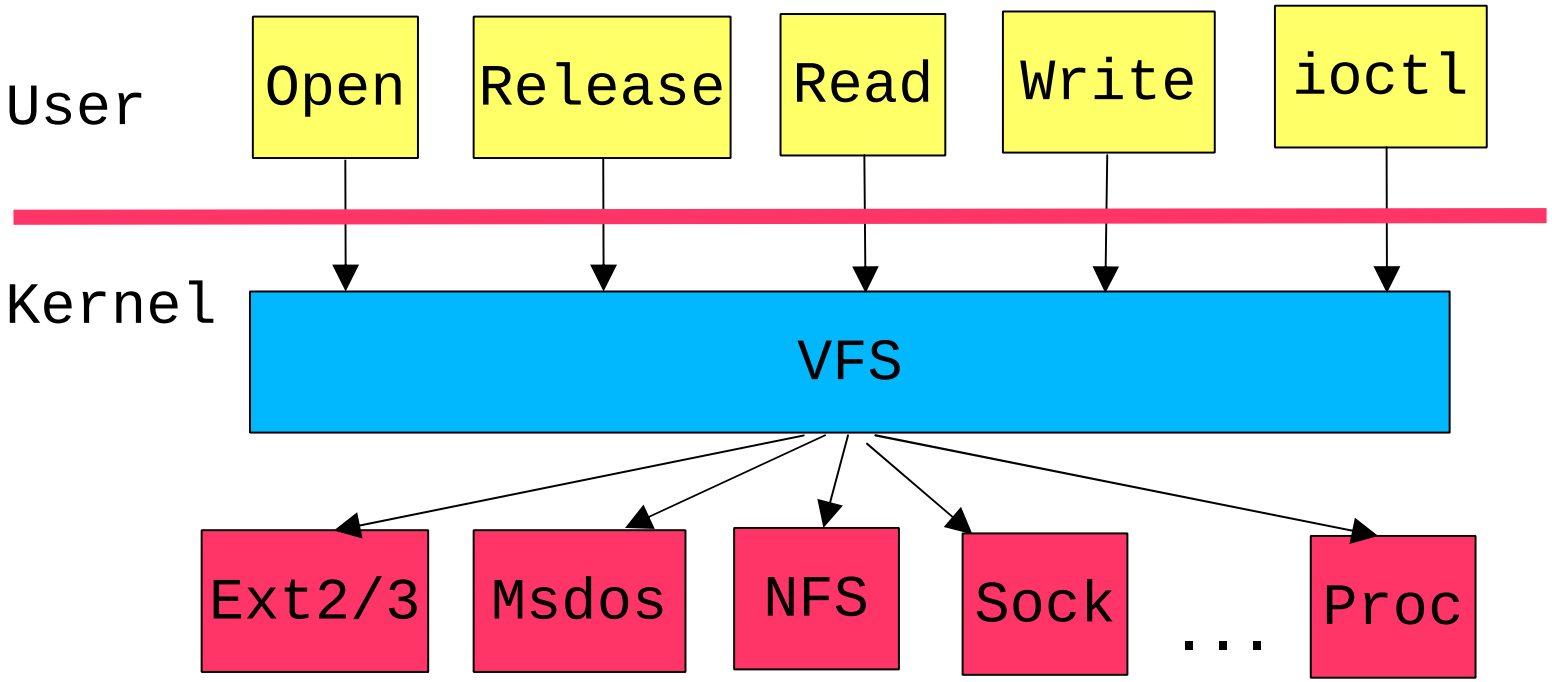
- 多种文件系统支持的基石-**VFS**
  - **V**irtual **F**ile **S**ystem
  - From SUN
  - vnode/inode
  - Provide Universal FS access interfaces

虚拟文件系统（续）



虚拟文件系统（续）

- Linux VFS Layer



虚拟文件系统（续）

- VFS Key Data Structure
  - **Superblock**

    Describe file system
  - **Inode**

    Describe file metadata
  - **File**

    Describe file data
  - **Dentry**

    Describe directory

文件系统的构成

- **i-节点**：它是一个包含着指针的控制结构，它的控制信息包括文件的所有者、访问权限、长度、最后一次存取时间、建立时间、用户分组GID号等等。

- **超级块**：从磁盘上读出来的第一块信息就是它的超级块。保存的信息包括：磁盘的几何尺寸、可用空间容量、以及最重要的第一个i节点的位置

管理文件系统

- **挂装和卸载本地磁盘**：文件系统的管理工作是从根目录开始的。包含着操作系统内核及核心目录结构的分区是在系统引导时挂装的，这个分区上必须存放有使系统进入单用户模式所必须的全部工具程序和配置文件，其上许多子目录都是空的。
- 随着引导脚本程序的执行，其他分区也挂装到文件系统结构上了。挂装命令用它正在挂装的分区上的目录树覆盖掉最初的某个子目录。
- 另外：当挂装上一个新子目录的时候，mount命令会把原来挂装在这个位置的全部内容隐藏起来。

管理文件系统（续）

- 1、使用mount命令

```
# mount [options] device directory
```

mount命令的部分参数

mount命令的参数	说　明
-a	把/etc/fstab文件（本小节后面介绍）中列出的文件系统都挂装上
-t *fstype*	定义挂装的文件系统类型。Linux可以挂装其他非ext2标准的文件系统，比如FAT、VFAT以及FAT32等等。mount命令通常可以自己检测出这类信息
-o *options*	定义作用于挂装过程的选项。它们通常是一些与文件系统类型有关的选项（挂装网络文件系统的选项不能够用来挂装本地文件系统）

管理文件系统（续）

- 2、使用**mount -o**命令

  例如：把**/dev/hda3**分区以只读属性挂装到**/usr**子目录上

  # mount –o ro /dev/hda3 /usr

与**mount**命令的**-o**参数连用的部分参数

ro	以只读属性挂装该分区
rw	以读-写属性挂装该分区（缺省值）
exec	允许二进制代码的执行（缺省值）
noatime	禁止刷新i-结点上的存取时间。用于存取时间不重要的分区（比如新闻队列），可以提高性能
noauto	如果使用了-a参数，禁止这个分区的自动挂装（只作用于/etc/fstab文件）
nosuid	禁止setuid程序的应用程序对此挂装分区置位
sb = n	告诉mount命令对一个ext2文件系统使用第n个数据块作为超级块

管理文件系统（续）

- 3、使用**umount**命令卸载文件系统

```
# umount [-f] directory
```

umount命令有一个**不足**之处：如果文件系统正在使用中（也就是说有人在那个分区上打开了文件），就无法把这个文件系统卸载下来。有三种解决办法：

- 一、 使用 **lsof**程序或者**fuser**程序检查有哪些进程打开了文件，终止那些进程的运行或者让进程的所有者停止操作
- 二、 使用**umount**命令和**-f**参数强制执行卸载操作。任何在这个分区上打开的进程都将被挂起，可能会造成数据丢失
- 三、 最安全和适当的办法是把系统调整为单用户模式，然后再卸载这个文件系统

管理文件系统（续）

- 4、**/etc/fstab**文件

/etc/fstab是一个**mount**命令可以利用的配置文件。这个文件包含着一个系统中全部已知硬盘分区的清单。在引导过程中，这个清单被读出，其中包含的各个分区都被自动挂到系统上。

/etc/fstab文件数据项中的各个数据元素：

/etc/fstab文件的数据项	说　明
/dev/device	将被挂装的分区（比如/dev/hda3）
/dir/to/mount	分区挂装到其上的子目录（比如/usr）
fstype	文件系统的类型（比如ext2）
parameters	mount命令-o参数的附加参数
fs_freq	告诉dump命令备份这个文件系统的频率
fs_passno	告诉fsck程序在引导时确定文件系统的检查顺序（请注意所有文件系统在挂装之前都要被检查）

管理文件系统（续）

- **使用fsck程序** **fsck**工具程序的名字是**F**ile **S**ystem **C**hec**k**（文件系统检查）的缩写，它被用来诊断和修复在日常操作中可能已经损坏的文件系统。
- 通常，系统在引导过程中如果发现有某个分区没有正常地卸载，就会自动运行**fsck**工具程序。
- 不足之处是：为了执行这个程序，需要诊断的文件系统必须先卸载下来。
- **e2fsck**

```
# e2fsck /dev/hda3
```

管理文件系统（续）

- **e2fsck**的可用参数

e2fsck的参数	说　明
-b *superblock*	让e2fsck读取分区信息的超级块编号。大多数情况下，e2fsck可以在第一个数据块中找到它，但是如果那个块损坏了，就需要指定另外一个号码。超级块每隔8192个出现一次，因此第二个超级块在8193、然后是16385等等
-c	在运行e2fsck之前先执行badblocks程序。它对整个硬盘按块查找并校验该块的完整性。这是检查硬盘最彻底的方法，但是花的时间比较多
-f	强制进行检查，即使认为文件系统已经没有问题了
-y	告诉e2fsck对e2fsck提示的问题全部自动回答为“Yes”

- **lost+found**子目录

e2fsck找到了一些文件碎片，但是没有办法把他们恢复到原始文件中去。这种情况下，它会把这些碎片放到该分区的**lost+found**子目录。这个子目录就在该分区挂装的位置。

任何东西都可以放到**lost+found**子目录里—文件碎片、子目录、一些特殊文件

硬盘分区

- **硬盘的表示方法** Linux中，每个硬盘都分配一个自己的设备名。IDE硬盘的名称以**/dev/hdX**开始，其中的**X**是从a到z的小写字母，每个字母代表一个物理设备。
- 到建立分区的时候，就需要用到新的设备了。它们的形式是**/dev/hdXY**，其中**X**是设备字母，而Y就是分区编号。这样，在**/dev/hda**硬盘上的第一个分区就是**/dev/hda1**。
- SCSI硬盘遵守与IDE一致的基本机制，但设备名不使用**/dev/hd**的形式而是**/dev/sd**打头。因此第一个SCSI硬盘上的第一个分区就是**/dev/sda1**,第三个SCSI硬盘上的第二个分区就是/dev/sdc2。

硬盘分区（续）

- **建立硬盘分区** 各种Linux发行版本上都有的一个基本程序是**fdisk**。它是一个可靠的分区工具。
- 假设准备对**/dev/hdb**设备、一个340MB的IDE硬盘进行分区。从执行带**/dev/hdb**参数的**fdisk**开始。输入下面的命令：

```
# fdisk /dev/hdb
```

- 屏幕上出现一个简单的提示符：

```
Command ( m for help) :
```

硬盘分区（续）

```
Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
Command (m for help):
```

硬盘分区（续）

- 现在从查看现有的硬盘分区开始，使用**p**命令（显示分区表）：

```
Command (m for help): p
Disk /dev/hdb: 16 heads, 63 sectors, 665 cylinders
Units = cylinders of 1008 * 512 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1   *           1         664      334624+   6  FAT16
Command (m for help):
```

硬盘分区（续）

- 给系统升级—先用**d**命令删除现有的分区：

```
Command (m for help): d
Partition number (1-4): 1
Command (m for help):
```

- 再使用**p**命令检验操作结果：

```
Command (m for help): p
Disk /dev/hdb: 16 heads, 63 sectors, 665 cylinders
Units = cylinders of 1008 * 512 bytes
   Device Boot      Start         End      Blocks   Id  System
Command (m for help):
```

硬盘分区（续）

- 需要建立下表的分区

分　区	说　明
/	根分区，存放用来把系统引导到单用户模式所必须的核心系统文件。一旦建立好以后，这个分区中的内容就不应该变化，也绝不需要增长。目的是把这个文件系统与其他文件系统隔离开来，防止影响核心操作
/usr	这个分区用来保存系统软件，比如用户工具程序、编译器、 X-Windows等等。因为今后可能需要我们为这些系统软件找一个更大的“家”，所以现在就把它放在一个单独的分区里
/var	/var分区用来保存变化很大的文件——通常包括队列子目录（电子邮件、打印等等）和日志文件。这个分区令人担心的情况是：外部活动可能会使其内容增长到分配给它的空间以外去。举例来说， Web服务器上的日志文件就可能增长得很快，让你不好控制。为了防止这些文件散布到系统的其他部分，把这个分区独立出去是明智的（网络上有一种攻击类型其原理是这样的：在被攻击方的服务器上人为地制造出数量巨大的活动，使得其硬盘被系统日志占满，影响系统工作的可靠性）
/tmp	类似于/var，/tmp中的文件也有可能消耗大量的空间。当用户不照管自己的程序或者应用程序生成了巨大的临时文件的时候，就会出现这种情况。不管是哪一种情况，给它分配一个分区是一个安全的好办法
/home	如果需要在硬盘上保存登录子目录，特别是有一些需要限制其硬盘空间使用量的用户时，肯定就需要把它单独安排在一个分区
swap	swap分区是用来保存虚拟内存的。虽然它并不是必须的，但保留 swap以备物理RAM全部消耗殆尽的情况发生是个不错的主意。通常，可以把这个分区设置为与RAM同样的大小

硬盘分区（续）

- 先建立根分区。因为只有340MB的工作空间，要把根分区设置小一点儿，假设25MB：

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-665, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-665, default 665): +25M
Command (m for help):
```

硬盘分区（续）

- 为swap建立第二个分区：

```
Command (m for help): m
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (52-665, default 52): 52
Last cylinder or +size or +sizeM or +sizeK (52-665, default 665): +16M
Command (m for help):
```

硬盘分区（续）

- 缺省情况下，**fdisk**建立的是**ext2**类型的分区，用t命令（改变分区类型）把这个分区设置为**swap**类型：

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): L
 0  Empty           16  Hidden FAT16    61  SpeedStor       a6  OpenBSD
 1  FAT12           17  Hidden HPFS/NTF 63  GNU HURD or Sys a7  NeXTSTEP
 2  XENIX root      18  AST Windows swa 64  Novell Netware  b7  BSDI fs
 3  XENIX usr       24  NEC DOS         65  Novell Netware  b8  BSDI swap
 4  FAT16 <32M      3c  PartitionMagic  70  DiskSecure Mult c1 DRDOS/secFAT-
 5  Extended        40  Venix 80286     75  PC/IX           c4 DRDOS/secFAT-
 6  FAT16           41  PPC PReP Boot   80  Old Minix       c6 DRDOS/secFAT-
 7  HPFS/NTFS       42  SFS             81  Minix / old Lin c7  Syrinx
 8  AIX             4d  QNX4.x          82  Linux swap      db CP/M / CTOS
 9  AIX bootable    4e  QNX4.x 2nd part 83  Linux           e1  DOS access
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 84  OS/2 hidden C:  e3  DOS R/O
 b  Win95 FAT32     50  OnTrack DM      85  Linux extended  e4  SpeedStor
 c  Win95 FAT32 (LB 51  OnTrack DM6 Aux 86  NTFS volume set eb  BeOS fs
 e  Win95 FAT16 (LB 52  CP/M            87  NTFS volume set f1  SpeedStor
 f  Win95 Ext'd (LB 53  OnTrack DM6 Aux 93  Amoeba          f4  SpeedStor
10  OPUS            54  OnTrackDM6      94  Amoeba BBT      f2 DOS secondary
11  Hidden FAT12    55  EZ-Drive        a0  IBM Thinkpad hi fe  LANstep
12  Compaq diagnost 56  Golden Bow      a5  BSD/386         ff  BBT
14  Hidden FAT16 <3 5c  Priam Edisk
Hex code (type L to list codes): 82
Changed system type of partition 2 to 82 (Linux swap)
Command (m for help):
```

硬盘分区（续）

- 下面建立/usr。大小设置为100MB:

```
Command (m for help): m
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (85-665, default 85): 85
Last cylinder or +size or +sizeM or +sizeK (85-665, default 665): +100M
Command (m for help):
```

硬盘分区（续）

- 现在建立容纳/tmp、/var和/home的扩展分区。使用n命令：

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
e
Partition number (1-4): 4
First cylinder (289-665, default 289): 289
Last cylinder or +size or +sizeM or +sizeK (289-665, default 665): 665
Command (m for help):
```

硬盘分区（续）

- 创建扩展分区中的三个分区/tmp、/var和/home

```
Command (m for help): n
First cylinder (289-665, default 289): 289
Last cylinder or +size or +sizeM or +sizeK (289-665, default 665): +100M
Command (m for help): n

First cylinder (493-665, default 493): 493
Last cylinder or +size or +sizeM or +sizeK (493-665, default 665): +45M
Command (m for help): n
First cylinder (585-665, default 585): 585
Last cylinder or +size or +sizeM or +sizeK (585-665, default 665): 665
Command (m for help):
```

硬盘分区（续）

- 使用**w**命令（把分区表写到硬盘并退出）使修改生效并退出**fdisk**程序
- 根据配置情况自行编写一个**/etc/fstab**文件

```
/dev/hdb1       /          ext2      defaults    1 1
/dev/hdb2       swap       swap      defaults    0 0
/dev/hdb3       /usr       ext2      defaults    1 2
/dev/hdb5       /home      ext2      defaults    1 2
/dev/hdb6       /var       ext2      defaults    1 2
/dev/hdb7       /tmp       ext2      defaults    1 2
none            /proc      proc      defaults    0 0
none            /dev/pts   devpts    mode=0622   0 0
```

建立文件系统

- 两个工具程序：
  - **mke2fs建立ext2文件系统**

```
[ root@lenovo /root ] # mke2fs /dev/hdb3
```

  - **mkswap建立swap文件系统**

```
[ root@lenovo /root ] # mkswap /dev/hdb2
```

网络文件系统

- 网络文件系统可以让你在使用客户端处理用户大计算量任务的同时对外提供磁盘存储服务
- 集中的硬盘意味着简单化的备份解决方案和现实的安全性
- Linux中，硬盘的集中是通过网络文件系统（Network File System，**NFS**）来实现的

网络文件系统（续）

- 挂装NFS分区

  挂装NFS分区与挂装本地分区的过程几乎是一样的。唯一的区别在于分区是如何确定的。

```
mount technics:/export/SL1200/MK2 /projects/topsecret1
```

下面是/etc/fstab文件中的一个NFS 挂装数据项示例：

```
denon:/export/DN2000F /proj/DN2k nfs bg, intr, hard, wsize=8192, rsize=8192 0 0
```

网络文件系统（续）

- NFS又引入了一些可以与mount命令的-o参数联合使用的参数

表8-6　与mount命令的-o参数联合使用的参数

mount -o命令的参数（NFS分区）	说　明
soft	“软挂装”该分区。如果服务器没有响应，客户端会倒计时一个预定的期间，并撤消请求的操作
hard	“硬挂装”该分区。如果服务器没有响应，客户端会一直等下去直到服务器恢复为止。如果服务器恢复了，不会有任何数据丢失
timeo=n	把倒计时时间设置为n秒
wsize=n	把写缓冲区大小设置为n个字节。缺省值是1024；推荐值是8192
rsize=n	把读缓冲区大小设置为n个字节。缺省值是1024；推荐值是8192
bg	让挂装操作在后台运行。如果在开始的时候挂装不上，就可以把挂装进程放到后台去继续尝试。如果在 /etc/fstab文件中有NFS挂装点，就必须包括这个参数

使用Automounter自动挂装子系统

- 1. 启动Automounter

```
# /etc/rc.d/init.d/autofs start
# /etc/rc.d/init.d/autofs reload
```

- 2. 配置/etc/auto.master文件

```
#
# Sample /etc/auto.master file
# (lines which begin with a '#' are comments)
#
/mount/point            map-file        global-options
/home               auto.home
/usr/local          auto.local
/misc               auto.misc
```

使用Automounter自动挂装子系统（续）

- 3．配置/etc/auto.misc文件

```
#
# auto.misc
#
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage
kernel          -ro,soft,intr           ftp.kernel.org:/pub/linux
cd              -fstype=iso9660,ro      :/dev/cdrom
# the following entries are samples to pique your imagination
#floppy         -fstype=auto            :/dev/fd0
#floppy         -fstype=ext2            :/dev/fd0
#e2floppy       -fstype=ext2            :/dev/fd0
#jaz            -fstype=ext2            :/dev/sdc1
```

硬盘空间配额的管理

- 1. 设置引导过程
- 2. 配置各个分区
  - **usrquota**参数
  - **grpquota**参数
  - 配额数据库

硬盘空间配额的管理（续）

- 设置配额
  - 软限制
  - 硬限制
  - 限制期（时间限制）
  - **edquota**的命令行参数

参　数	说　明
-u *logtn*	为参数定义的用户设置配额数据
-t	为硬盘分区设置限制期。把它与-u或者-g参数联合使用可以分别为用户或者用户分组全部设置好限制期。请注意：如果用户 /用户分组在同一个分区上，它们就不能有不同的限制期
-g *group*	为参数定义的用户分组设置配额数据
-p *logtn*	允许把一个用户的配额资料（用户名为 login）克隆给另外一个用户。这个参数必须与-u参数合用

硬盘空间配额的管理（续）

- 管理配额
  - 三个管理硬盘配额的工具程序

    quotacheck、repquota和quota
  - 可以传递到**quotacheck**的参数如下

-v	打开报告模式，再检查配额数据库的时候就会看到许多有用又有意思的信息
-u *uid*	检查UID是uid的用户的配额情况
-g *gid*	检查GID是gid的用户的配额情况
-a	检查所有设置了配额的文件系统（以/etc/fstab文件中的设置为准）
-R	与-a参数合用。检查所有设置了配额的硬盘分区，但是不包括根分区

硬盘空间配额的管理（续）

- repquota命令用来生成系统上配额使用情况的统计报告，参数如下

-a	统计所有文件系统的配额使用情况
-v	统计所有配额的使用情况，没有用到的也要统计
-g	以用户分组为单位统计配额使用情况
-u	以用户为单位统计配额使用情况

- 可以传递到quotacheck的参数如下

-g	给出用户所在分组的配额使用情况
-u	给出该用户的配额使用情况（缺省操作）
-v	给出支持配额的所有文件系统里与该用户有关的配额使用情况
-q	如果该用户已经超标，显示一个消息给他

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- **核心级系统服务**
- 编译Linux内核
- 提高单个服务器的安全性

核心级系统服务

- init服务
- telinit命令
- inetd进程
- syslogd守护进程
- cron程序

init服务

- **init**的进程编号（process ID，PID）永远是**1**
  - **/etc/inittab**文件

    id : runlevels: action : process
  - **/etc/inittab**文件语句格式组成部分的说明

/etc/inittab元素	说　明
id	由1到4个字符组成的、在/etc/inittab文件中对本数据项进行定义的独一无二的字符串
runlevel	必须调用此进程的运行级别。有的活动是如此的特殊，以至于在全部运行级别中都可以被捕获（比如用于重启动的 Ctrl+Alt+Del组合键）。如果想表示某个活动在全部运行级别中都需要处理，将 runlevel项空着就可以了。如果想定义在某几个运行级别中发生一些事情，只需要把这些运行级别全部列在该数据域中就可以了。举例来说，如果runlevel数据域的值是123，就表示在运行级别1、2、和3中需要进行操作
action	定义需要进行的操作。这个数据域的参数马上会在下面说明
process	定义进入某个运行级别后需要执行的进程（程序）

init服务（续）

- 定义在/etc/inittab文件中action数据域中的参数

/etc/inittab文件里action数据域中的参数	说　明
respawn	被终止时立刻需要重新启动的进程
wait	一进入某个运行级别就必须运行的进程，init会等待它运行结束
once	一进入某个运行级别就必须运行的进程，init不需要等待它运行结束就可以执行需要在此运行级别中运行的其他程序
boot	需要在系统引导时运行的进程。此时运行级别域中的数据将不起作用
bootwait	需要在系统引导时运行的进程，init需要等待它运行结束之后才能继续到下一个需要运行的进程
ondemand	当某个特定运行级别请求出现时需要运行的进程（这些运行级别是a、b和c），运行级别不发生变化
initdefault	定义init启动时的缺省运行级别。如果没有定义缺省值，就会在控制台上提示用户输入一个运行级别
sysinit	系统引导过程中，需要在任何其他boot或者bootwait数据项之前运行的进程
powerwait	如果init从另外一个进程收到电源出现问题的信号，则将运行这个进程。init会等到这个进程结束之后再继续向下执行
powerfail	相当于powerwait，但init不必等到这个进程结束之后才继续向下执行
powerokwait	如果init收到与powerwait类型相同的信号，并在etc/powerstatus文件中保存有字符串“OK”，就将运行这个进程。init会等到这个进程结束之后才继续向下执行
ctrlaltdel	如果init收到一个信号，表明用户按下了CTRL-ALT-DEL组合键，就将运行这个进程。需要注意的是大多数X-Windows服务器会捕获这个组合键，因此如果激活了X-Windows，init就不会收到这个信号

telinit命令

- 两个命令行参数
  - 一个参数用来通知**init**准备切换过去的运行级别
  - 一个是**-t sec** ，其中的**sec**是在通知**init**之前需要等待的以秒计算的时间

inetd进程

- inetd的角色是作为telnet和ftp等与网络服务器相关的进程的“超级服务器”
  - etc/inetd.conf文件

```
srvce_name  sock_type  protocol  [no]wait  user  srvr_prog  srvr_prog_args
```

  - etc/inetd.conf文件中服务定义语句中的元素

名　称	说　明
srvce_name	所提供服务的名称。这个名称是与 /etc/services文件中的定义相关联的，该文件把服务的名称与提供该服务的端口联系在一起。如果打算在 etc/inetd.conf文件中添加新的内容，那么还必须在 /etc/services文件中添加一个相应的数据项（不用有什么顾虑，因为/etc/services文件的格式是不连续的）

inetd进程（续）

- /etc/inetd.conf文件中服务定义语句中的元素

名　称	说　明
sock-type	套接字类型可以是stream或者dgram。stream值代表面向连接的（即TCP）数据流（比如：telnet和ftp）。dgram值代表数据报（datagram，即UCP）流，比如：tftp服务就是一个基于数据报的协议）。不同于TCP/IP范围以外的协议确实是存在的；但是，很少会遇到它们
protocol	这是一个在/etc/protocols文件中定义的值——典型值是tcp、udp，或者是定义在RPC基础上的TCP服务的rpc/tcp，以及定义在RPC基础上的UDP服务的rpc/udp
[no]wait	可以是wait或者nowait。对任何流（TCP）式连接，必须使用nowait。对其他连接来说，这个值取决于进程支持的数据报连接的类型。 wait代表的意思是：客户将连接到服务器，在断开连接之前必须等待一个回答（这通常被称为“单线”服务器，因为它一次只能处理一个请求）。nowait代表的意思是：客户在发送完自己的数据之后就立刻断开连接
user	进程将作为本数据项中定义的服务运行。使用这个数据项的时候要十分谨慎——一个不正确的选择有可能导致安全方面的漏洞。一般来说，选择服务的时候要尽量选那些能够以非根用户的身份来运行的。如果你必须运行一个只能以根用户身份启动的服务，一定要保证程序有良好的设计，并且要阅读全部的文档以保证已经对它进行了正确的配置
srvr_prog	这是连接到本语句定义的服务时将要执行的程序的完整路径
srvr_prog_args	程序的名称，加上该程序的参数。举例来说，我们假设srvr_prog参数定义（见上面的说明）的完整路径是/usr/bin/in.fingerd，并且准备把-l参数传递给它。那么，srvr_prog参数就将是/usr/bin/in.fingerd，而srvr_prog_args就是in.fingerd -l

inetd进程（续）

- **/etc/inetd.conf**文件中的一个数据项示例

有两个数据项：**ftp**和**telnet**。它们两个都是流式TCP服务通过被称为“TCP打包器”的**/usr/sbin/tcpd**程序调用TCP打包器程序将接受对该程序的连接，记录连接请求并检查**/etc/hosts.allow**和**/etc/hosts.deny**两个文件，确定客户是否被允许连接到这些服务上

```
#
# These are standard services.
#
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

inetd进程（续）

- 安全性与**inetd.conf**文件
  - 关掉尽可能多的服务
  - 把**/etc/inetd.conf**文件里所有用不着的服务性说明语句都改为注释语句
  - 向守护进程报告其配置文件已经被修改
    - 通过向该守护进程发送**HUP**信号来实现
    - **# ps auxw | grep inetd | grep -v grep**

root 359 0.0 0.1 1232 168 ? S Jun21 0 : 00 inetd

    - **# kill -1 359**

syslogd守护进程

- 系统全局性的日志记录守护进程
  - 记录特殊事件和消息的标准机制
  - **syslogd**保存数据用的记录文件都是简明的文本文件
    - **/var/log**目录
    - 每个数据项构成一行，包括日期、时间、主机名、进程名、进程的PID、以及来自该进程的消息
    - 使用**logger**命令在记录文件中生成数据项

syslogd守护进程（续）

- 调用syslogd的命令行参数

参　数	说　明
-d	调试状态。在启动的时候，syslogd一般会交出当前终端的控制权，让自己在后台运行。使用了-d参数之后，syslogd将保留对终端的控制权，并在记录消息的同时显示调试信息。一般几乎没有机会用到这个参数
-f *config*	定义另外一个配置文件代替缺省的 /etc/syslog.conf文件
-h	在缺省的情况下，syslogd不会把发送给它但实际目的地是另外一个主机的消息转发出去。如果使用了这个参数，就有可能被用来作为实施拒绝服务攻击的一个部分
-l hostlist	这个参数列出愿意记录其活动的主机名单。每个主机使用的是它的简单名称，不是它的完全授权域名（fully qualified domain name，FQDN）。允许列出多个主机，彼此用冒号隔开，如下所示： `-l toybox : ford : oid`
-m interval	缺省情况下，syslogd每隔20分钟生成一个记录数据项，以此作为“让你知道我正在运行”的消息。这只适用于并不繁忙的系统（如果你正在观察系统的记录，要是过了20分钟还没有看到一条消息，则说明有什么地方出现了问题）。如果定义了一个数值作为时间间隔，就可以指定syslogd需要每隔多少分钟生成一条消息
-r	缺省情况下，作为安全预警措施，syslogd守护进程会拒绝从网络上发送给它的消息。这个参数将激活本功能
-s domainlist	如果你正在接收的syslogd数据项给出了FQDN，可以让syslogd滤除域名，只保留主机名。只需要在一个以冒号隔开的清单中简单地列出域名作为 -s选项的参数就可以了。如下所示： `-s x-files.com : conspiracy.com : wealthy.com`

syslogd守护进程（续）

- /etc/syslog.con文件中功能值的等价字符串

功能值等价字符串	说　明
auth	身份验证消息
authpriv	基本上与auth相同
cron	由cron子系统（参见后面的章节）产生的消息
daemon	各种服务守护进程的基本信息
kern	内核消息
lpr	打印子系统消息
mail	电子邮件子系统消息（包括对每一个电子邮件的记录）
mark	已弃用，但是有的书里还介绍它；syslogd只是简单地忽略它
news	通过NNTP子系统的消息
security	与auth相同的东西（不应再使用）
syslog	来自syslog的内部消息
user	来自用户程序的消息
uucp	来自UUCP（UNIX to UNIX CoPy的缩写，意思是UNIX到UNIX的复制）的消息
local0 - local9	基本功能级别，它们的重要性可以根据需要来决定

syslogd守护进程（续）

- /etc/syslog.conf文件中优先权值的等价字符串

优先权级别等价字符串	说　明
debug	调试语句
info	杂项信息
notice	重要语句，但并不一定是坏消息
warning	潜在危险情况
warn	与warning相同（不应再使用）
err	出现一个错误
error	与err相同（不应再使用）
crit	严重事件
alert	指明发生重大事件的消息
emerg	紧急事件

syslogd守护进程（续）

- 保存目的地数据项示例

目的地格式	说　明
/var/log/logfile	一个文件 注意，如果你在文件名前面加上短划线字符（-），则syslogd在完成了写操作之后不会同步文件系统。这也就是说如果使用了短划线字符（-），在系统有机会清空其缓冲区之前有可能会丢失一些数据。但是从另外一方面来说，如果某个应用程序需要记录的消息太多太噜嗦，使用这个参数可以改进性能 记住：如果用户打算把记录消息发送到控制台上，就必须指定 /dev/console设备
| /tmp/mypipe	一个管道。这种类型的文件是使用 mknode命令（参见第6章）建立的。当syslogd在管道的一端读入数据的时候，可以运行另外一个程序从管道的另一端读取数据。这是使用程序分析记录消息输出、查找严重问题情况的有效方法，这样在必要的时候可以及时收到报警
@loghost	主机名。这个例子将把消息发送到 loghost主机。loghost主机上的syslogd守护进程将记录那些消息

cron程序

- cron程序允许系统中的任一用户安排某个程序在任何日期、时间、或者星期几准时运行，时间可以精确到分钟
  - cron是由开机引导命令脚本程序启动
  - cron服务的工作原理是
    - 每隔一分钟唤醒一次，检查每个用户的crontab文件
    - 用来编辑由cron调入执行的设置项工具是crontab
    - crontab检查/etc/cron.allow和/etc/cron.deny这两个文件确定用户适当的权限
    - 文件格式：

```
Minute  Hour  Day  Month  DayOfWeek  Command
```

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- 核心级系统服务
- **编译Linux内核**
- 提高单个服务器的安全性

编译Linux内核

- 简介
- 获取内核源代码
- 建立内核
- 安装内核
- 给内核打补丁

内核简介

- 开放源代码

  linux内核源代码的开放性**GNU GPL**(**G**eneral **P**ublic **L**icense)

- Linux内核 = 纯粹意义上的linux

  各个版本的linux中同样的东西

- 系统软件的核心

  系统的基础

获取内核源代码

- **http://www.kernel.org**

  内核的web官方站点

- 选择正确的版本

  偶数的比较稳定，没有test标识 **2.4.18**

- 解包源代码

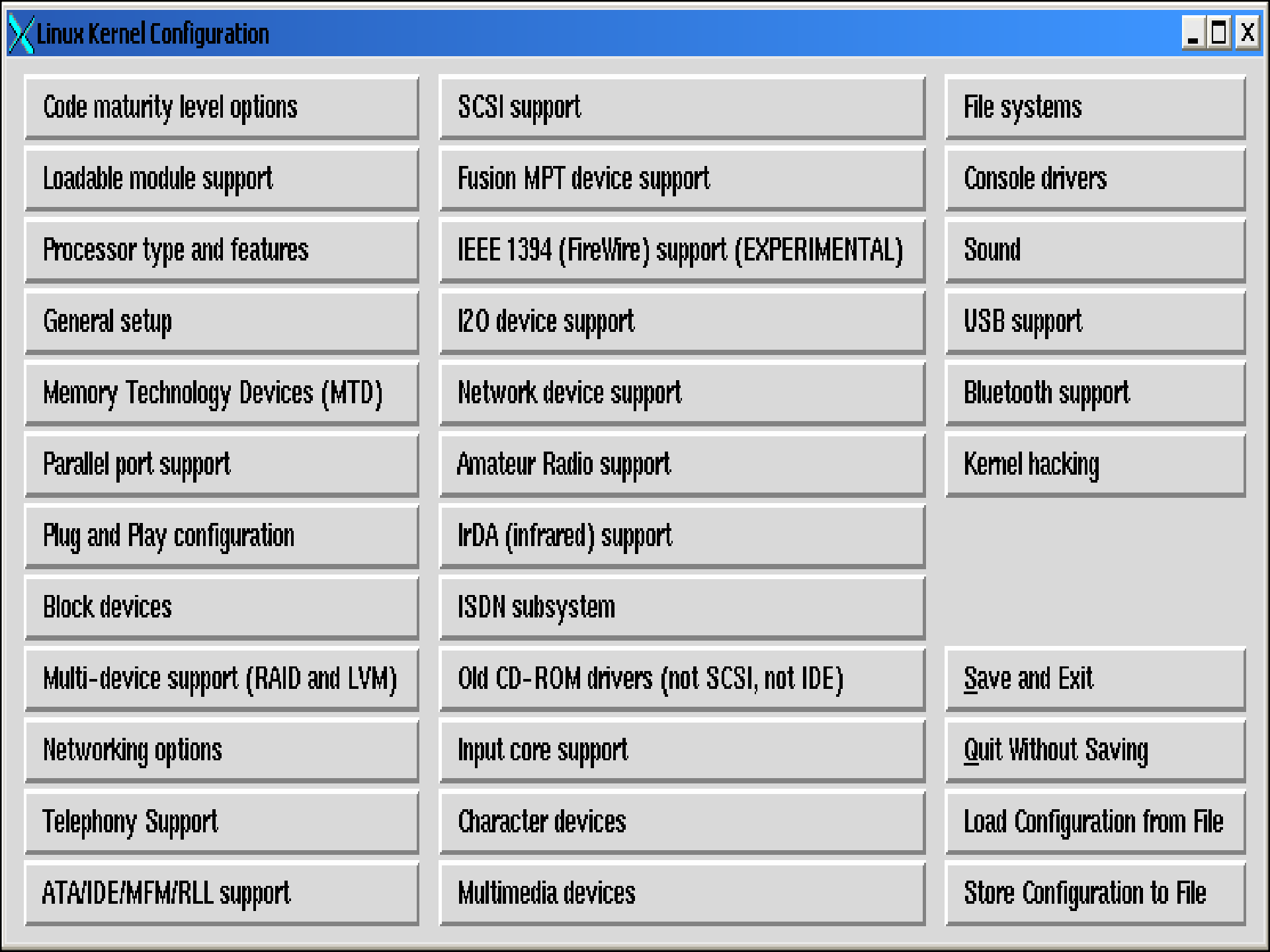
  /usr/src/linux-version

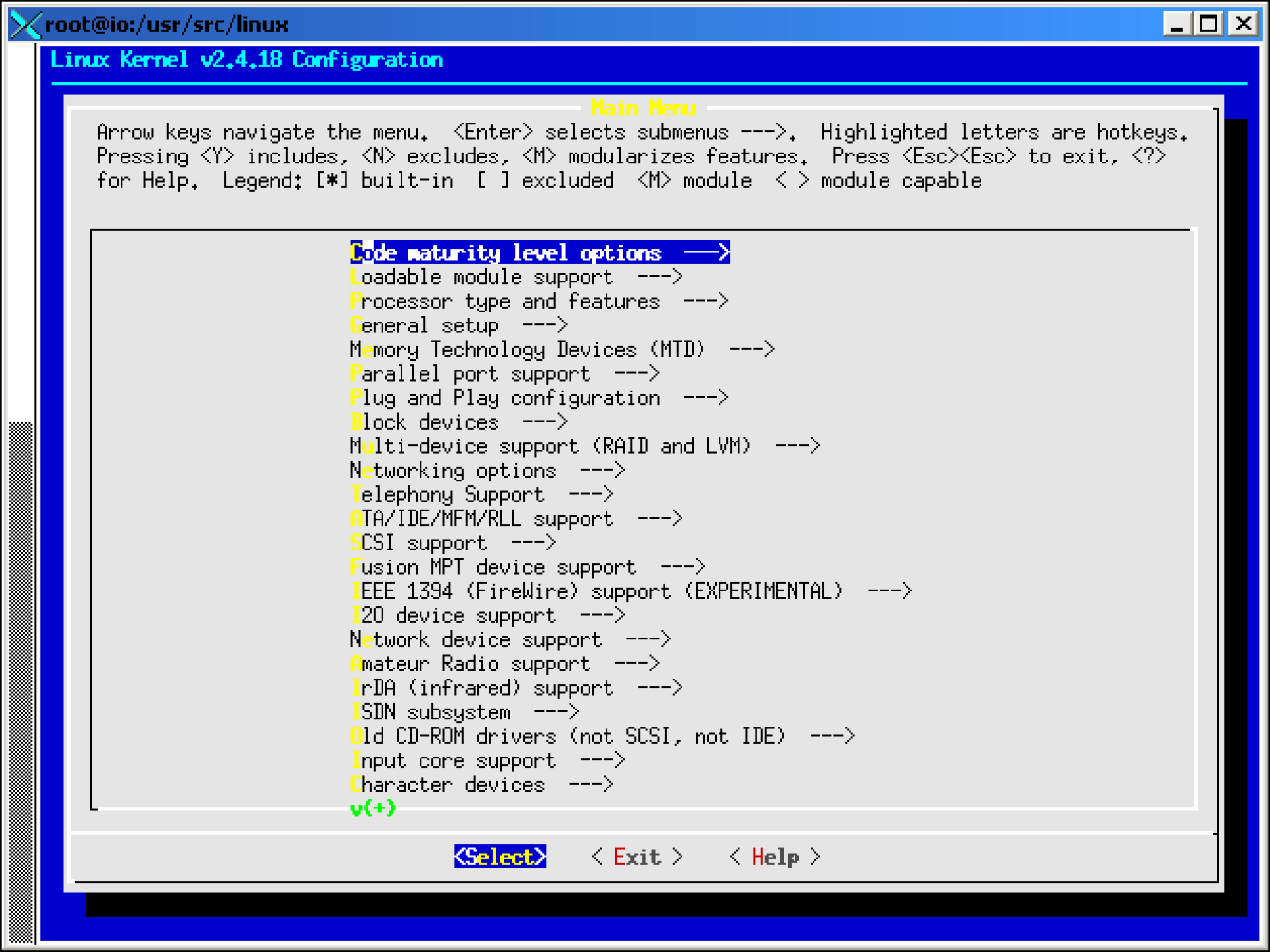
建立内核–配置

- 通过配置，选择自己最需要的模块和功能
- 了解你自己的硬件

  /proc/pci
- **Make *config**

  xconfig ,menuconfig ,config


Linux Kernel Configuration
Code maturity level options
Loadable module support
Processor type and features
General setup
Memory Technology Devices (MTD)
Parallel port support
Plug and Play configuration
Block devices
Multi-device support (RAID and LVM)
Networking options
Telephony Support
ATA/IDE/MFM/RLL support
SCSI support
Fusion MPT device support
IEEE 1394 (FireWire) support (EXPERIMENTAL)
I2O device support
Network device support
Amateur Radio support
IrDA (infrared) support
ISDN subsystem
Old CD-ROM drivers (not SCSI, not IDE)
Input core support
Character devices
Multimedia devices
File systems
Console drivers
Sound
USB support
Bluetooth support
Kernel hacking
Save and Exit
Quit Without Saving
Load Configuration from File
Store Configuration to File

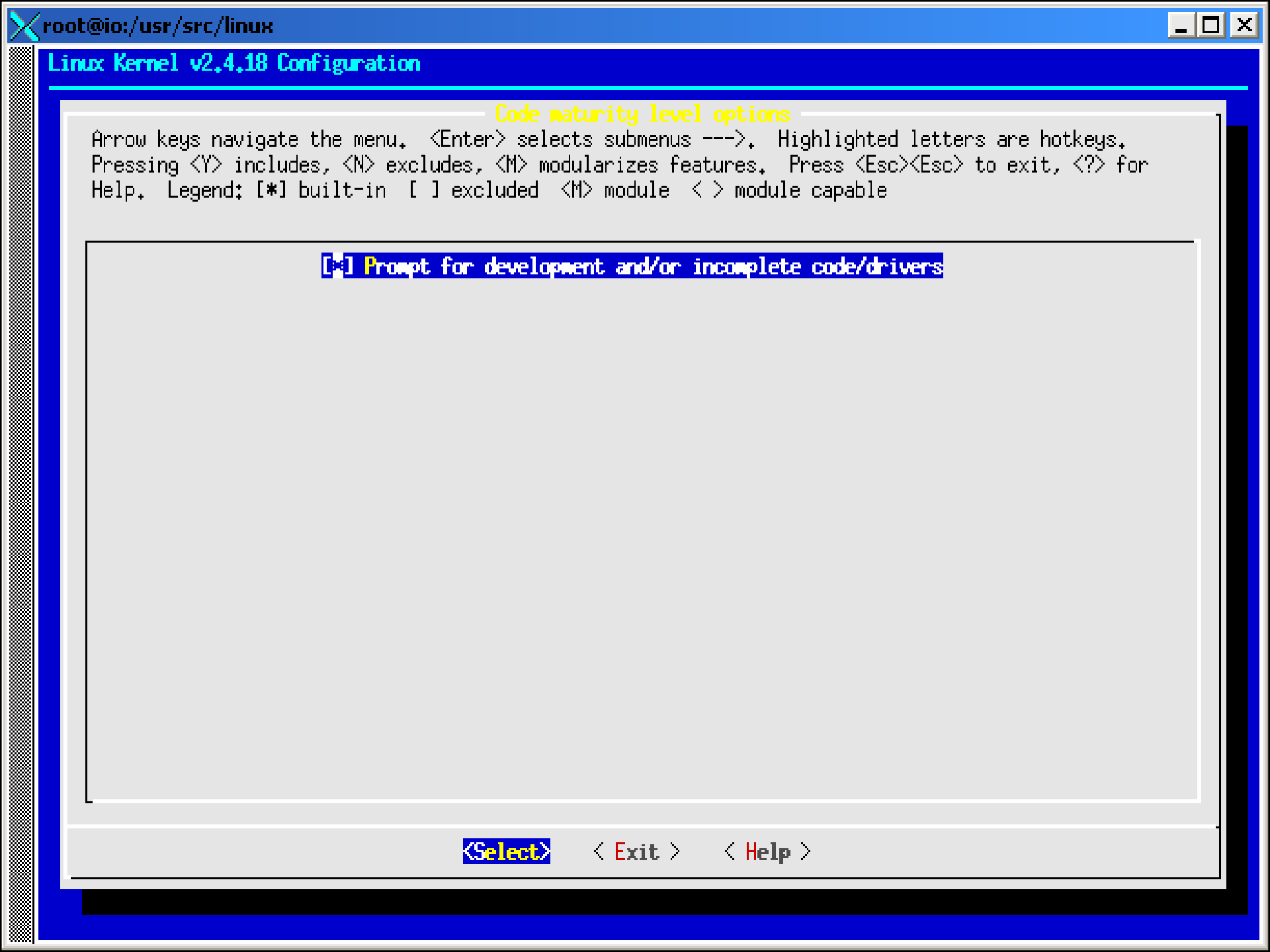

root@io:/usr/src/linux
Linux Kernel v2.4.18 Configuration
Main Menu
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded <M> module < > module capable
Code maturity level options --->
Loadable module support --->
Processor type and features --->
General setup --->
Memory Technology Devices (MTD) --->
Parallel port support --->
Plug and Play configuration --->
Block devices --->
Multi-device support (RAID and LVM) --->
Networking options --->
Telephony Support --->
ATA/IDE/MFM/RLL support --->
SCSI support --->
Fusion MPT device support --->
IEEE 1394 (FireWire) support (EXPERIMENTAL) --->
I2O device support --->
Network device support --->
Amateur Radio support --->
IrDA (infrared) support --->
ISDN subsystem --->
Old CD-ROM drivers (not SCSI, not IDE) --->
Input core support --->
Character devices --->
v(+)
<Select>
< Exit >
< Help >

建立内核–配置（续）

- 代码成熟水平(code maturity level options)

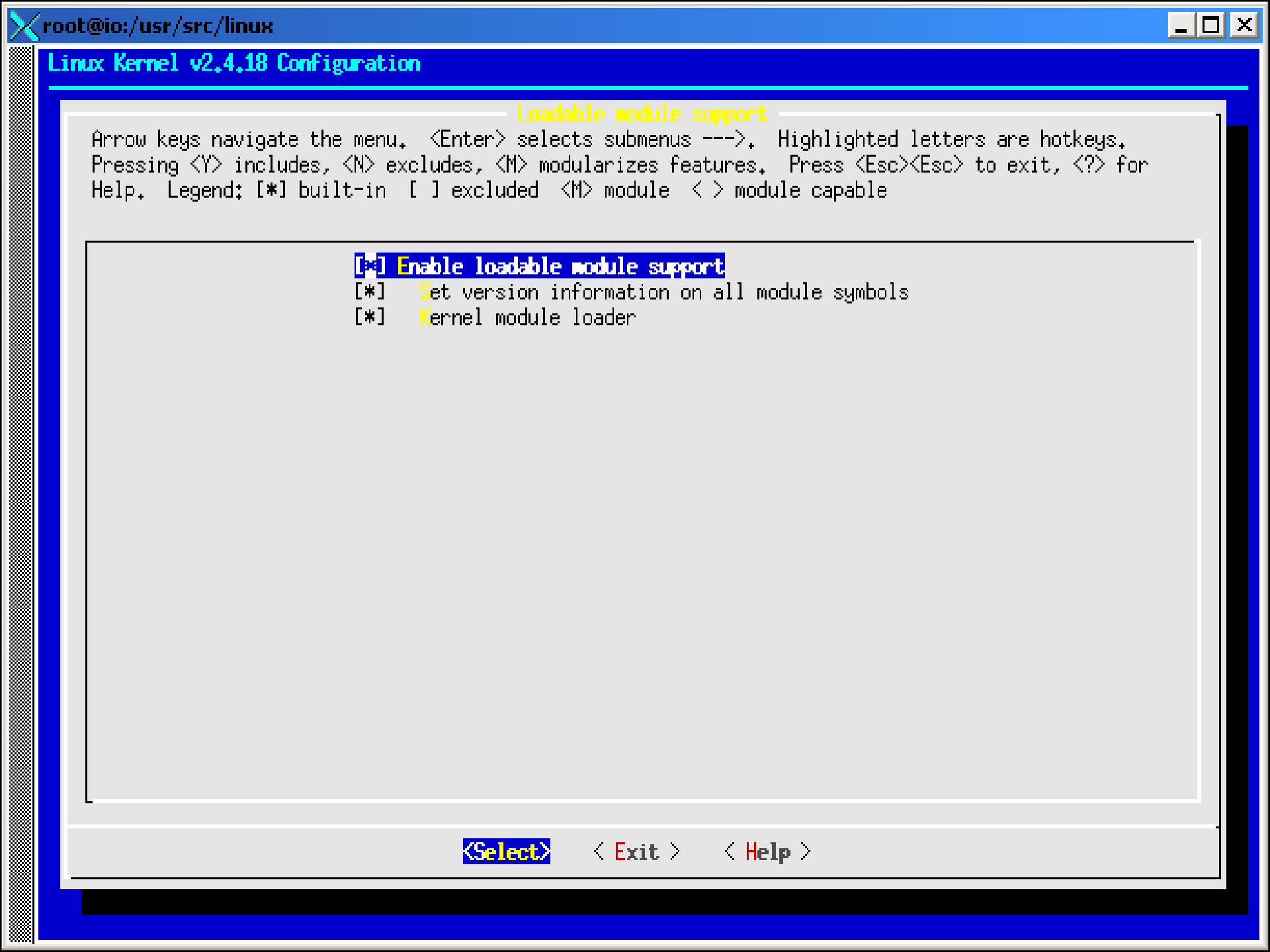
  提供进行实验的新特征

  如果不选，这些新特性不会出现在下面的选择中


root@io:/usr/src/linux
Linux Kernel v2.4.18 Configuration
Code maturity level options
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] Prompt for development and/or incomplete code/drivers
<Select>
< Exit >
< Help >

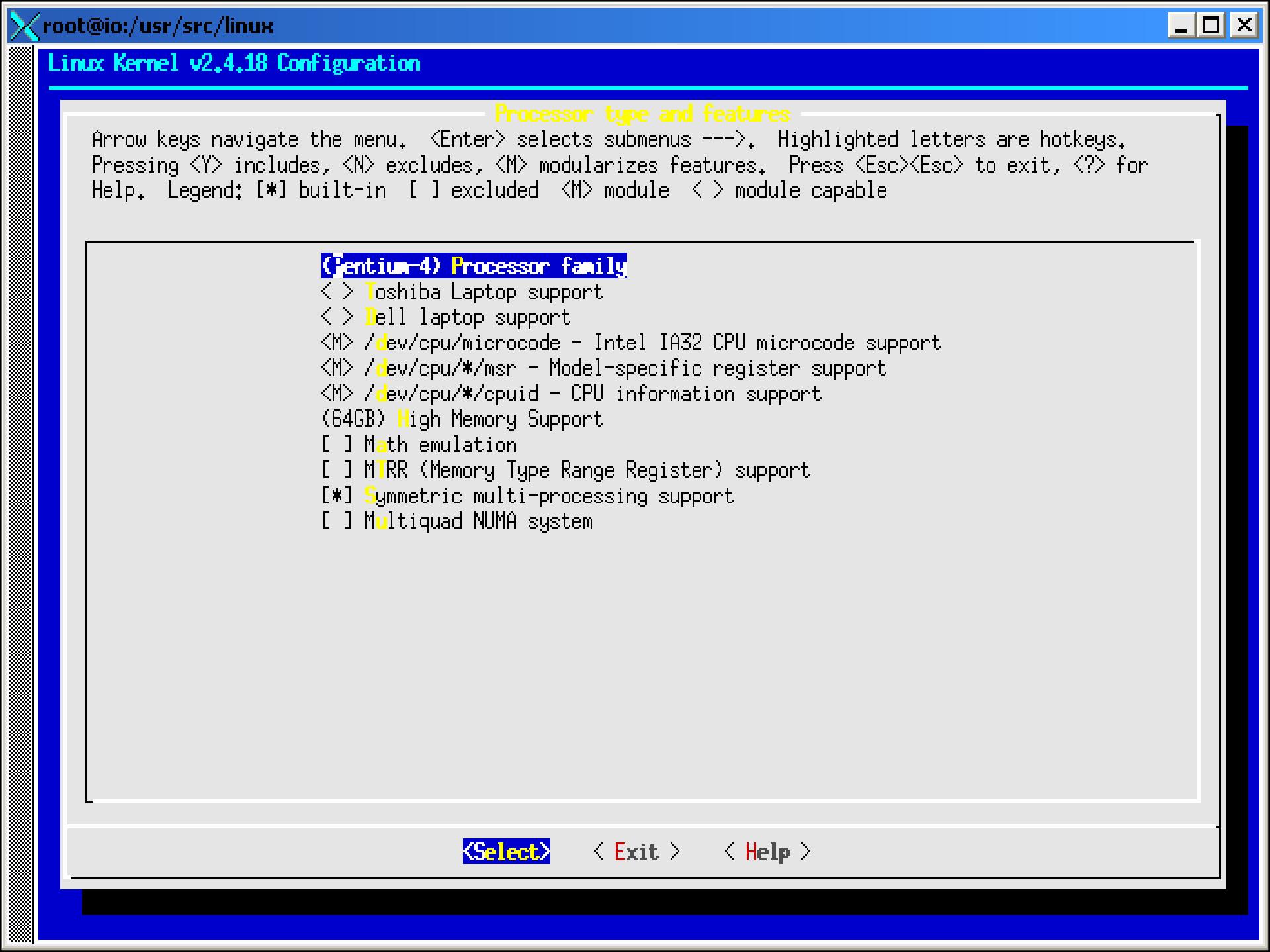
建立内核–配置（续）

- 可调入模块支持(loadable module support)：
  - 模块使得在不引起系统开销的情况下随时调入内核需要的对设备和文件系统的支持
  - 保持模块的版本的一致
  - 模块一般是由**init**脚本调用，或使用其他显式调用**insmod**和**rmmod**工具的shell脚本在需要时调入或调出模块


root@io:/usr/src/linux
Linux Kernel v2.4.18 Configuration
Loadable module support
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for
Help. Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] Enable loadable module support
[*] Set version information on all module symbols
[*] Kernel module loader
<Select>
< Exit >
< Help >

建立内核–配置（续）

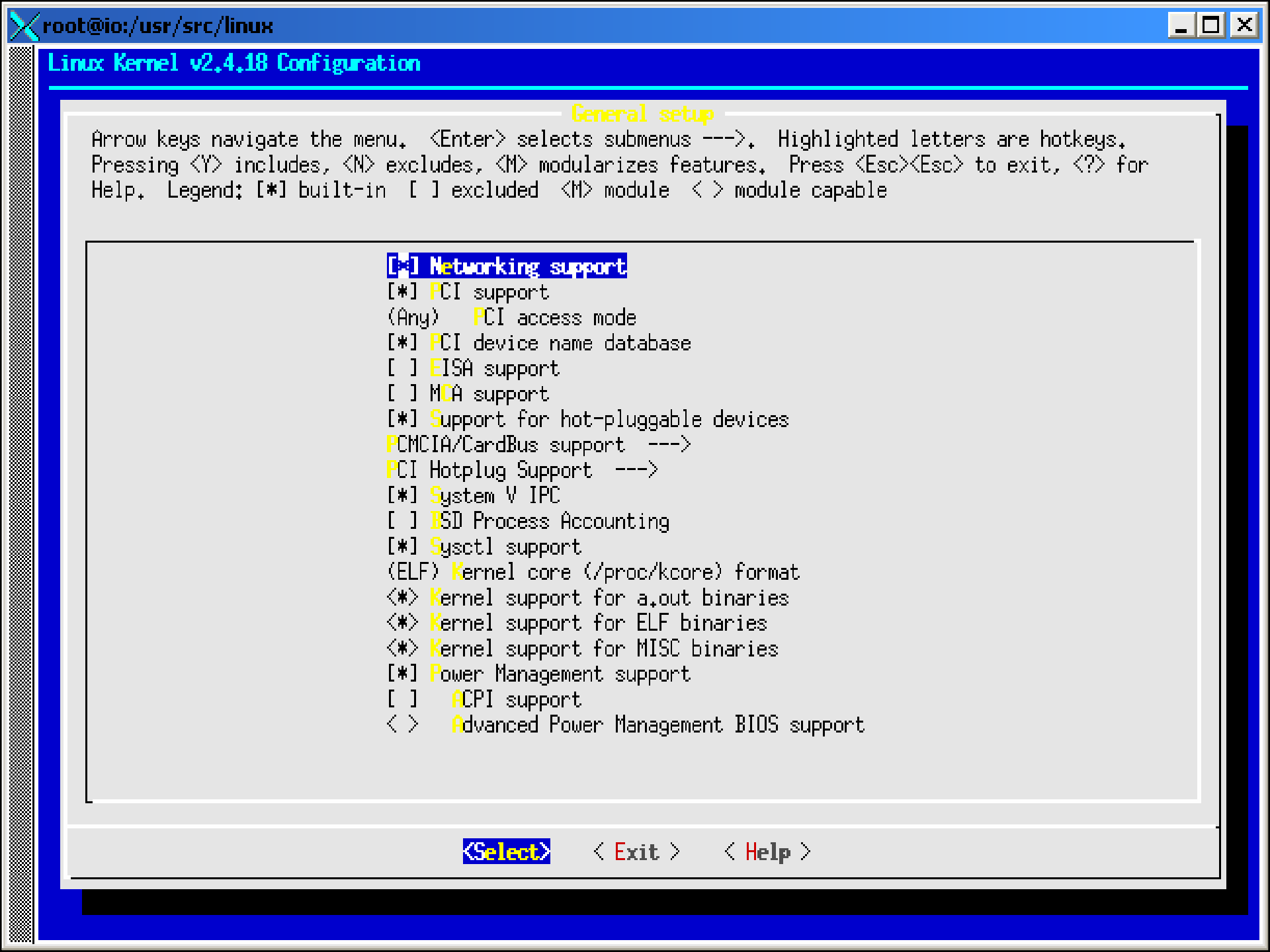
- 处理器类型和特征(processor type and features)：
  - SMP支持设置
  - 处理器类型的设置
  - 最大内存支持设置


root@io:/usr/src/linux
Linux Kernel v2.4.18 Configuration
Processor type and features
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded <M> module < > module capable
(Pentium-4) Processor family
< > Toshiba Laptop support
< > Dell laptop support
<M> /dev/cpu/microcode - Intel IA32 CPU microcode support
<M> /dev/cpu/*/msr - Model-specific register support
<M> /dev/cpu/*/cpuid - CPU information support
(64GB) High Memory Support
[ ] Math emulation
[ ] MTRR (Memory Type Range Register) support
[*] Symmetric multi-processing support
[ ] Multiquad NUMA system
<Select>
< Exit >
< Help >

建立内核–配置（续）

- 一般设置(general setup)：

  一般设置包含起用网络，PCI硬件，支持二进制文件类型，高级电源管理等。


root@io:/usr/src/linux
Linux Kernel v2.4.18 Configuration
General setup
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys.
Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for
Help. Legend: [*] built-in [ ] excluded <M> module < > module capable
[*] Networking support
[*] PCI support
(Any) PCI access mode
[*] PCI device name database
[ ] EISA support
[ ] MCA support
[*] Support for hot-pluggable devices
PCMCIA/CardBus support --->
PCI Hotplug Support --->
[*] System V IPC
[ ] BSD Process Accounting
[*] Sysctl support
(ELF) Kernel core (/proc/kcore) format
<*> Kernel support for a.out binaries
<*> Kernel support for ELF binaries
<*> Kernel support for MISC binaries
[*] Power Management support
[ ] ACPI support
< > Advanced Power Management BIOS support
<Select> < Exit > < Help >

建立内核–配置（续）

- 并行端口支持(parallel port support)
- 内存技术设备(memory technology devices)

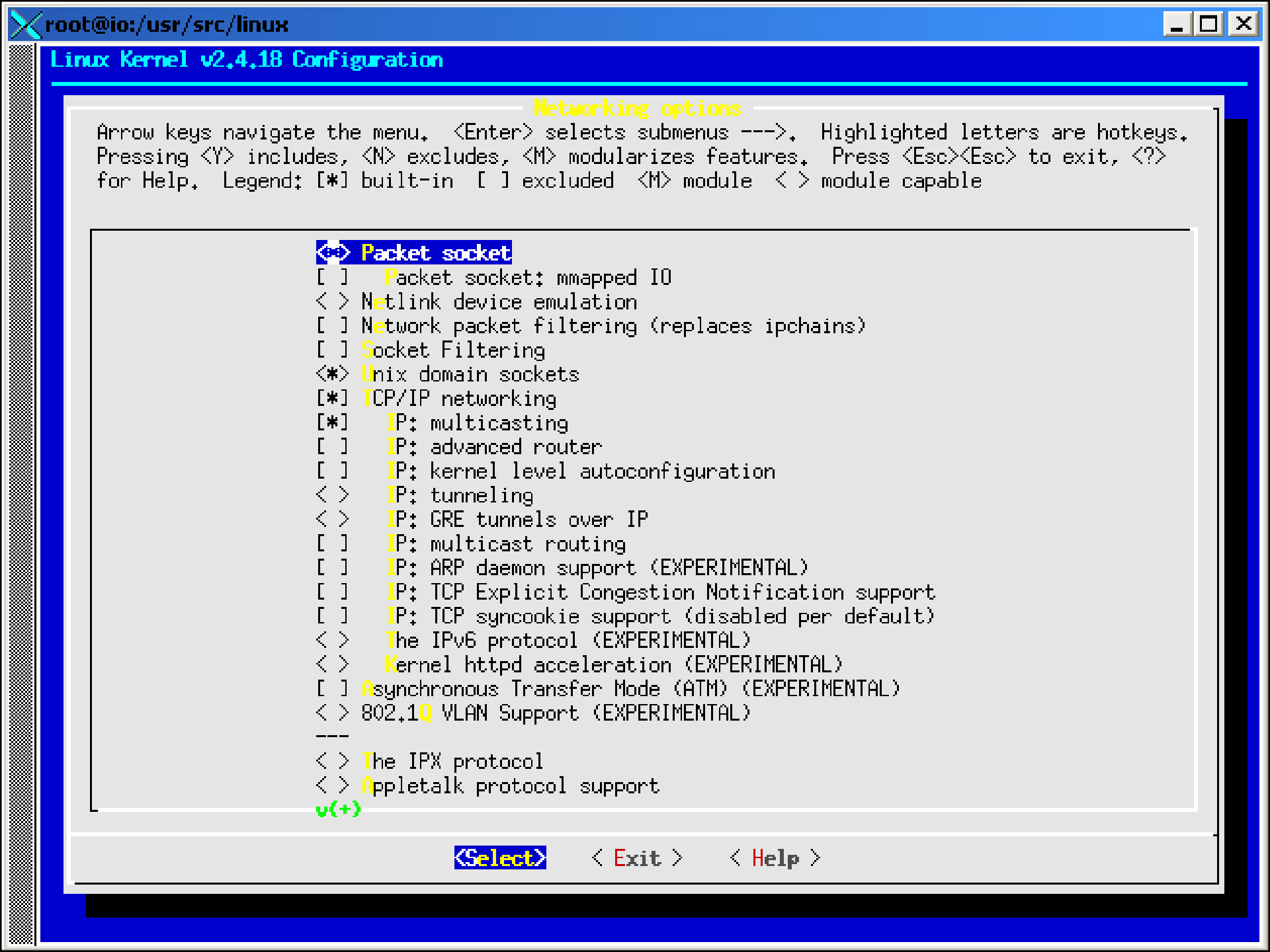
  支持特殊的内存设备，如闪存卡等
- 即插即用的支持(plug and play support)
- 块设备(block devices)

  列出linux对硬盘和**cd－rom**等设备的支持选项
- 多设备的支持

  raid和**lvm**等
- 对IP电话的支持(telephony support)
- **ATA/IDE/MFM/RLL** support

  linux的缺省可用于所有的IDE硬盘
- 网络选项(networking options)

  至少需要激活**tcp/ip**网络选项


root@io:/usr/src/linux
Linux Kernel v2.4.18 Configuration
Networking options
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*] built-in [ ] excluded <M> module < > module capable
<*> Packet socket
[ ] Packet socket: mmapped IO
< > Netlink device emulation
[ ] Network packet filtering (replaces ipchains)
[ ] Socket Filtering
<*> Unix domain sockets
[*] TCP/IP networking
[*] IP: multicasting
[ ] IP: advanced router
[ ] IP: kernel level autoconfiguration
< > IP: tunneling
< > IP: GRE tunnels over IP
[ ] IP: multicast routing
[ ] IP: ARP daemon support (EXPERIMENTAL)
[ ] IP: TCP Explicit Congestion Notification support
[ ] IP: TCP syncookie support (disabled per default)
< > The IPv6 protocol (EXPERIMENTAL)
< > Kernel httpd acceleration (EXPERIMENTAL)
[ ] Asynchronous Transfer Mode (ATM) (EXPERIMENTAL)
< > 802.1Q VLAN Support (EXPERIMENTAL)

< > The IPX protocol
< > Appletalk protocol support
v(+)
<Select> < Exit > < Help >

建立内核–配置（续）

- 对SCSI的支持(SCSI support)
- 对IEEE 1394的支持(IEEE 1394 (firewire) support)
- 对I2O设备的支持(I2O device support)
- 网络设备的支持(network device support)
- 对业余无线电的支持(amateur radio support)
- 对IrDA红外线设备的支持(IrDA support)
- ISDN子系统(isdn subsystem)
- 老式CD-ROM驱动程序(old cd一rom drivers)
- 对输入核心代码的支持(input core support)

  指linux域键盘进行通信所需要的必要代码量

建立内核–配置（续）

- 字符设备(character devices)
- 多媒体设备(multimedia devices)
- 文件系统(file system)

  包括对ext2，ext3，ntfs等的支持
- 控制台驱动程序(console drivers)
- USB的支持(usb support)
- 内核修改(kernel hacking)

  进行内核再开发

建立内核–编译

- `# make dep`
- `# make clean`
- `# make bzImage`
- `# make modules`
- `# make modules_install`

安装内核

- **/usr/src/linux-version/arch/i386/boot/bzImage**
- **/boot/vmlinuz-x.x.x**
- **/boot/System.map-x.x.x**
- **/etc/lilo.conf**

安装内核（续）

- lilo.conf

```
prompt
timeout=50
default=linux-2.4.18
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
message=/boot/message
linear
```

安装内核（续）

- lilo.conf（续）

```
image=/boot/vmlinuz-2.4.7-10smp
        label=linux
        initrd=/boot/initrd-2.4.7-10smp.img
        read-only
        root=/dev/sda3
image=/boot/vmlinuz-2.4.7-10
        label=linux-up
        initrd=/boot/initrd-2.4.7-10.img
        read-only
        root=/dev/sda3
```

给内核打补丁

- 下载补丁文件

  以**patch**打头 **patch-x.x.xx.gz**

- 解压补丁文件

- **# Patch –p0 < patch-x.x.xx**

- 编译安装

第二部分 单主机系统的管理

- 用户的管理
- 命令行
- 开机和关机
- 文件系统
- 核心级系统服务
- 编译Linux内核
- 提高单个服务器的安全性

提高单个服务器的安全性

- TCP/IP与网络安全
- 追踪服务
- 对系统实施监控
- 利用软件来实现监控
- 安全问题总结

TCP/IP与网络安全

- 监听特定端口的服务程序
- 扫描端口号发现不安全因素
- **netstat**命令查看当前连接

追踪服务

- **netstat**命令查看当前连接
  - **# netstat –natu**
  - 查看那些端口被打开、那些端口上有进程在监听
  - 缺省情况**netstat**命令列出网络和本地套接字上全部已建立的连接状态
- **netstat**命令输出与系统的安全性
  - **/etc/services**文件显示启动的服务和它们关联的端口
  - **# netstat –p**
- 利用**netstat**的输出结果分析

追踪服务（续）

- 关闭服务
  - 关闭**xinetd**和**inetd**服务

    /etc/xinetd中将**disable**设置为**Yes**

    /etc/inetd.conf中注释掉不要的服务

    运行 **# /etc/rc.d/init.d/xinetd reload**
  - 关闭非**inetd**服务

    改变**/etc/rc.d/**目录中的符号连接，将符号连接的第一个字母由**S**改为**X**
  - **syslogd**服务

    以 **# syslogd –r**启动时记录来自其他主机的请求

    缺省情况下，**syslogd**不会使用**-r**参数启动

对系统实施监控

- 充分利用**syslog**命令
  - 日志分析
  - 保存日志文件的数据项
  - 将日志中重要部分直接打印出来

利用软件来实现监控

- 使用MRTG监控带宽
  - **M**ulti **R**outer **T**raffic **G**rapher
  - 多路由通信负荷图形监控；生成网络性能报告
- COPS
  - **C**omputer **O**racle and **P**assword **S**ystem
  - 计算机访问权限和口令检查系统；查看系统中不正常的访问
- TripWire
  - TripWire监控系统（商务软件包）
  - 对系统中每一个文件生成**MD5**校验和**checksum**，并保存

利用软件来实现监控（续）

- SATAN
  - **S**ystem **A**dministrators **T**ool for **AN**alyzing
  - 系统管理员分析工具；监测网络中是否存在潜在的安全漏洞
- CERT
  - **C**omputer **E**mergency **R**esponse **T**eam
  - 计算机应急反应小组；定期公布大量的公告的工具软件
  - http://www.cert.org

安全问题总结

- 模糊安全法
  - PGP(**P**retty **G**ood **P**rivacy)
  - 源代码和加密算法都公开；反而提供了最高的安全性
- 社会安全问题
  - 警惕内部人员的泄密问题
- 物理安全问题
  - 只有值得信赖的人才能接触机器

第一部分　安装Linux操作系统作为服务器软件
第二部分　单主机系统的管理
第三部分　Internet网络服务
第四部分　内部网(Intranet)网络服务
第五部分　Linux操作系统的高级网络功能

lenovo 联想

第三部分 Internet网络服务

- DNS域名服务
- FTP文件传输服务
- Apache Web服务
- SMTP邮件发送服务
- POP邮件接受服务
- SSH安全登录服务

第三部分 Internet网络服务

- **DNS域名服务**
- FTP文件传输服务
- Apache Web服务
- SMTP邮件发送服务
- POP邮件接受服务
- SSH安全登录服务

DNS域名服务

- DNS的简化版本：/etc/hosts文件

```
#
# Host table for Intranet network
#
127.0.0.1 localhost     localhost.localdomain
192.168.1.171    node171        c0201
……
192.168.1.178    node178        c0208
192.168.3.171    node371        t0201
……
```

DNS域名服务（续）

- **DNS**的组成部分
  - 域和主机

    倒置的树装结构；分级查询
  - 子域

    `tsinghua.edu -> cs.tsinghua.edu`
  - 正向解析与反向解析

    `www.cs.ucr.edu -> 138.23.169.15`

    `138.23.169.15 -> www.cs.ucr.edu`
  - 三种服务器

    主控、辅助、缓冲

DNS域名服务（续）

- 安装**DNS**服务器

  BIND 9.0

  - 下载、解包、阅读

    tar ; ./configure ; make ; make install

  - 安装文件

    /usr/local/bin/host 执行简单查询

    /usr/local/bin/dig 跟踪服务器路径

    /usr/local/sbin/named DNS服务器程序

  - 启动时引导DNS

    /etc/rc.d/rc3.d中指向/etc/rc.d/init.d/named脚本的链接

DNS域名服务（续）

- 配置**DNS**客户机
  - **/etc/resolv.conf**文件

    第一行指明确省的搜索域

    第二行指明该主机域名服务器的IP地址

    ```
    search      planetoid.org
    nameserver  127.0.0.1
    ```
  - **/etc/nsswitch.conf**文件

    告诉系统到哪里查找特定类型的配置信息

    ```
    hosts: files nisplus nis dns
    ```

    上述文件表明了搜索的依次位置

DNS域名服务（续）

- DNS服务器配置文件/named.conf的格式
  - 注释
  - statement关键字
  - ACL语句
  - include语句
  - logging语句
  - 配置参数选项
  - server语句
  - zone语句

DNS域名服务（续）

- 配置DNS服务器
  - 配置主域

```
zone domain-name{
  type master;
  file path-nam
}
```

  - 配置从域

```
  type slave;
  masters IP-address-list;
```

  - 配置缓冲域

```
  type hint;
  file “named.ca”
```

DNS域名服务（续）

- DNS记录的类型
  - SOA：授权开始
  - NS：域名服务器
  - A：地址记录
  - PTR：指针记录
  - MX：邮件交换器
  - CNAME：标准名称
  - RP/TXT：文档数据项

DNS域名服务（续）

- 完整的named.conf文件数据项

  $TTL开头

  每个记录的生存时间值

  SOA纪录

  NS纪录

  MX纪录

  A/CNAME纪录

  /var/named/named.ca

  /var/named/named.local

  /var/named/named.domain.com

  /var/named/named.rev

DNS域名服务（续）

- **DNS工具箱**
  - **kill-HUP**

    发送HUP信号给named进程，重新读取配置文件
  - **host**

    host theorb.com

    host 209.133.83.16
  - **dig**

    收集关于DNS服务器的信息
  - **whose**

    确定某个域的拥有者身份

第三部分 Internet网络服务

- DNS域名服务
- **FTP文件传输服务**
- Apache Web服务
- SMTP邮件发送服务
- POP邮件接受服务
- SSH安全登录服务

Ftp文件传输服务

- 基本原理
  - 服务端口 **21**
  - 数据传输端口大于1024
  - **passive transfer**
- **wu-ftpd**安装
  - 备份文件

    /etc/ftpaccess

    /etc/ftpgroups

    /etc/ftphosts

    /etc/ftpusers

Ftp文件传输服务（续）

- wu-ftpd安装

```
# ./build lnx
# ./build install
```

  - 修改inetd.conf文件

```
ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd
-l -a
```

  - 修改xinetd.conf文件

```
/etc/xinetd.d/wd-rtpd
# /etc/rc.d/init.d/xinetd reload
```

Ftp文件传输服务（续）

- **wu-ftpd**配置
  - **/etc/ftpaccess**控制访问权限
  - 访问控制命令

`class classname type address ……`	定义新的用户分类
`autogroup groupname class ……`	设置匿名用户的分组
`deny address message-file`	禁止登录
`guestgroup groupname ……`	降低用户权限
`limit class n times message_file`	限制登录时间数量等
`loginfials n`	限制失败次数
`private`	设置私有目录

Ftp文件传输服务（续）

- **wu-ftpd**配置
  - 前导信息控制

    banner filename

    email address

    message path when class ……

    readme path when class
  - 系统记录

    log commands

    log transfers
  - 其它

    alias ; cdpath ; compress

Ftp文件传输服务（续）

- **wu-ftpd配置**
  - 访问权限

```
chmod switch class
  switch 为 ON/OFF ; class 为 anonymous/guest/real
delete
overwrite
rename
umask
passwd-check
path-filter
upload
```

Ftp文件传输服务（续）

- 系统日志文件
  - **/var/log/xferlog**

    包含以下数据项：

    Current-time ; Transfer-time

    Remote-host ; File-size

    Transfer-type ; Special-action-flag

    Direction ; Access-mod

    Username ; Service-name

    Authentication-method

    Authenticated-user-id

Ftp文件传输服务（续）

- 即时文件转换
  - **/etc/ftpconversions**

  比如以压缩方式传送文件

  ```
  strippre : strippost ; addonpre ; addonpost ;
  external ; type ; options ; desc
  ```
- 配置主机访问权限
  - **/etc/ftphosts**

  ```
  allow username address
  deny username address
  ```

Ftp文件传输服务（续）

- 多种访问设置方案
  - 只允许匿名访问
    - 建立匿名用户
    - 建立匿名FTP目录
    - 建立**/etc/ftpaccess**文件
    - 建立**/incoming**目录
  - 混合式访问

    同时支持匿名用户和实际用户
  - 只允许注册用户访问

    /etc/passwd中取消**ftp**用户

    /etc/ftpaccess中使用**limit**命令

Ftp文件传输服务（续）

- 建立虚拟FTP服务器

  确认新IP地址在**/etc/hosts**和DNS清单中存在

  执行**ifconfig**命令

  ```
  ifconfig eth0:0 192.168.1.42 netmask 255.255.255.0
  broadcast 192.168.1.255
  ```

  建立ARP表

  ```
  arp –s earth 00:10:4B:CB:15:9F pub
  ```

  在**/etc/ftpaccess**文件中加**virtual**命令

  ```
  virtual IP ftype directory
  ```

第三部分 Internet网络服务

- DNS域名服务
- FTP文件传输服务
- **Apache Web服务**
- SMTP邮件发送服务
- POP邮件接受服务
- SSH安全登录服务

Apache Web服务

- Liunx下的Web服务器：**Apache**服务器
  - **Apache HTTP**服务器的优势

    稳定性强

    开放源代码

    灵活性高

    安全性高

    支持平台多

Apache Web服务（续）

- HTTP的基本原理

  HTTP（HyperText Transfer Protocol)超文本传输协议

  - 国际互联网**WWW**（World Wide Web）的一个重要组成部分。
  - 标准端口（默认端口）：**80**

    http://www.lenovo.com
  - 非标准端口：配置一个其他端口给Apache服务器

    http://www.lenovo.com ：端口号
  - 不使用80端口可以在一定程度上提高安全性，但是安全性的提高极为有限。

Apache Web服务（续）

- 进程所有者和安全性

  Apache对所有权的处理

  - Apache的启动必须由root用户执行。Apache只支持80端口的请求。完成连接后，Apache放弃它的权限，再根据自己配置文件中的定义，转换到一个非root用户运行，一般来说是nobody用户。

  - 作为nobody用户，Apache只能读取nobody用户有权限读取得文件。如果想让用户能够读取某个文件，就需要把这个文件设置为对整个系统可读：

```
# chomd a+r filename
```

Apache Web服务（续）

- 安装Apache HTTP服务器

  配置文件：**/etc/httpd/conf**

  顶级总控主页：**/var/www**

  服务器控制命令：**/etc/rc.d/init.d/httpd start/…**

- 安装软件

```
# tar –xzf apache.1.3.14.tar.gz
# ./configure –prefix = /usr/local/apache
# make
# make install
```

Apache Web服务（续）

- 安装Apache HTTP服务器
  - Apache模块
  - 在线升级Apache服务器

    /usr/local/apache_version

    修改符号链接
  - 确保nobody用户存在

    缺省作为nobody用户来运行

    /etc/passwd中

Apache Web服务（续）

- 安装Apache HTTP服务器
  - 控制Apache服务器

    /usr/local/apache/bin/apachectl start/stop
  - 开机运行Apache服务器

    /etc/rc.d/rc3.d/S85httpd

    /etc/rc.d/init.d/httpd
  - 测试Apache服务器

    /etc/local/bin/apache/bin/apachectl start

    缺省主页：

    /usr/local/apache/htdocs/index.html

Apache Web服务（续）

- 配置Apache HTTP服务器
  - 建立简单的顶级主页

```
# cd /usr/local/apache/htdocs
# echo ‘Welcome to Lenovo’ > index.html
# chmod 644 index.html
```

  - Apache配置文件

/usr/local/apache/conf/srm.conf

/usr/local/apache/conf/access.conf

/usr/local/apache/conf/httpd.conf 实际设置文件

Apache Web服务（续）

- 配置Apache HTTP服务器
  - 常见的配置修改

    将nobody用户改为其他用户

    建立新的用户www

    建立对应的用户组www

    将用户的Shell设置值定义为**/bin/false**： 该用户无法登录

    将口令设置为* ：该用户不能ftp访问

    编辑httpd.conf文件

    ```
    User www
    Group www
    ```

    修改**/usr/local/apache**目录中所有文件的存取权限

Apache Web服务（续）

- 配置Apache HTTP服务器
  - 常见的配置修改
    - 修改主机名

      Apache提供Web服务器的假名

      ServerName www.eng.domain.org

    - 服务器系统管理员

      Apache提供服务器系统管理员的Email假名

      ServerAdmin www@domain.com

Apache Web服务（续）

- 配置Apache HTTP服务器
  - 设置虚拟域

    允许从同一个IP来托管多个域，修改**httpd.conf**文件

```
#
#  Virtual Host Settings for Legend
#
<VirtualHost Legend>
  ServerAdmin TAR@Legend.com
  DocumentRoot /www/htdocs/Legend
  ServerName Legend
</VirtualHost>
```

Apache Web服务（续）

- 基本的故障诊断

  Apache有非常出色的错误日志文件

  /var/log/httpd/logs/access_log　Web站点访问文件

  包括：

  数据的传输是否成功完成

  访问站点的请求从哪个IP发出

  传输了多少数据

  数据在什么时间传输

  /var/log/httpd/logs/error_log　Apache发生的错误

  # tail -100 error_log | more　　查看最后的记录

第三部分 Internet网络服务

- DNS域名服务
- FTP文件传输服务
- Apache Web服务
- **SMTP邮件发送服务**
- POP邮件接受服务
- SSH安全登录服务

SMTP邮件发送服务

- SMTP基本原理

  Simple **M**ail **T**ransfer **P**otocol

  简单邮件传输协议

  - SMTP协议定义了电子邮件在主机之间传递的方法
  - SMTP的优势在于它的简单性
  - SMTP不依赖于具体的操作系统
- 基本SMTP命令

```
# telnet mailserver 25
HELO ; MAIL FROM ; RCPT TO ; DATA
```

SMTP邮件发送服务（续）

- 编译Sendmail软件

```
# tar –xzf sendmail.8.9.3.tar.gz
# ./Build
```

- 外围程序
  - MAILSTATS　　性能流量统计
  - MAKEMAP　　保存表格为数据库格式
  - PRALIASES　　设置aliasese假名文件
  - SMRSH　　增强Sendmail软件安全性

SMTP邮件发送服务（续）

- Sendmail软件设置
  - M4宏命令语言

    divert ; dnl ; define ; undefine
  - Sendmail软件使用的宏命令

    OSTYPE ; DOMAIN ; MAILER ; MASQUERADE_AS ; FEATURE
  - 可配置参数选项

    查看cf/README文件获得
  - 将宏命令编译到.mc配置文件里

    # m4 ../m4/cf.m4 ourconfig.mc > ourconfig.cf

SMTP邮件发送服务（续）

- 安装运行Sendmail软件

```
# ./Build install
# cp ourconfig.cf /etc/sendmail.cf
# /usr/sbin/sendmail –bd –p1h
```

-bd参数设置Sendmail进入守护进程状态

-p1h设置每个一个小时回到前台处理积压的电子邮件（因对方主机原因未能及时发送而放入队列）

SMTP邮件发送服务（续）

- 其它配置文件
  - aliases文件
  - “access”（访问权限）数据库
- 运行中的问题
  - mailq命令

```
# mailq
```

列出队列中的全部消息，包括编号，来源，目的地

  - Sendmail在运行吗

```
# /etc/rc.d/init.d/sendmail start
# sendmail –bd –q15m
```

SMTP邮件发送服务（续）

- 运行中的问题
  - 队列和缓冲池的位置：

    /var/spool/mqueue
  - 删除队列中的项目

    # ls –l /var/spool/mqueue | grep messageID

    删除引起麻烦的邮件文件，messageID是其文件编号

    # rm –I /var/spool/mqueue/*RAA735
  - 加速处理邮件队列

    # sendmail –bd –q15m
  - 对电子邮件进行系统记录

    /etc/syslog.conf -> /var/log/maillog

第三部分 Internet网络服务

- DNS域名服务
- FTP文件传输服务
- Apache Web服务
- SMTP邮件发送服务
- **POP邮件接受服务**
- SSH安全登录服务

POP邮件服务

- 基本原理

  Post **O**ffice **P**rotocal

  邮局协议

- 基本命令

```
# telnet pop3server.legend.com 110
USER yangkun
PASS mypassword
LIST 列出 ; RETR 阅读 ; DELE 删除
```

  - **POP协议与IMAP协议不一致，避免同时使用**

POP邮件服务（续）

- **Qpopper软件**
  - 安装

```
# tar –xzf qpopper2.53.tar.gz
```

  - 编译

```
# ./configure –help
# ./configure –enable-apop=/etc/pop.auth –with-
  popuid=bin –enable-specialauth –enable-servermode
# make
# make insatll
```

POP邮件服务（续）

- **Qpopper**软件
  - 设置

只保留一个POP数据项

```
# grep ‘110/tcp’ /etc/services
```

把Qpopper设置为POP3服务

```
/etc/inetd.conf
pop-3 stream tcp nowait root /usr/bin/popper popper
kill –l `cat /var/run/inetd.run`
```

  - 测试

```
# telnet localhost 110
```

POP邮件服务（续）

- **Qpopper**软件高级配置
  - 服务器工作模式
  - 特殊身份验证功能
  - 带身份验证加强功能的POP
  - 公告板
  - 命令行参数

    `-b bulldir`　　重新设置公告板目录设置值

    `-T timeout`　　修改缺省的倒计时时间设置

第三部分 Internet网络服务

- DNS域名服务
- FTP文件传输服务
- Apache Web服务
- SMTP邮件发送服务
- POP邮件接受服务
- **SSH安全登录服务**

SSH安全登陆服务

- 公共密钥密码学

  Public Key Cryptography

  两把密匙：一把公共、一把私有

  传输的数据被A的私有密匙和B的公共密匙加密后

  只能由A的公共密匙和B的私有密匙共同解开

  SSH定期更改私人密匙，可以每隔几分钟就改变

SSH安全登陆服务（续）

- SSH的发行版本

  Secure **SH**ell实现安全远程登录，防止被网络监听

  SSH协议成为IETF标准

  - OpenSSH/OpenBSD

    支持SSH1和SSH2两个版本的协议，两个版本不兼容
  - FreeSSH for Windows
  - SecureCRT for Windows
  - PliteSSH for Palm Pilot

SSH安全登陆服务（续）

- 安装SSH软件包

  需要**ZLib**包

```
# tar –xzvf zlib.tar.gz
# ./configure ; make ; make install
```

  需要**OpenSSL**包

```
# tar –xzvf openssl-0.9.6.tar.gz
# ./configure ; make ; make test ; make install
```

  需要**OpenSSH**包

```
# tar –xzvf openssh-2.2.0p1.tar.gz
# ./configure –with-ssl-dir=/usr/local/ssl –with-
tcp-wrappers ; make ; make install
```

SSH安全登陆服务（续）

- SSH服务的启动、关闭
  - 直接启动方法

    /usr/local/sbin/sshd
  - 自动启动方法

    修改**/etc/rc.d/rc.local**
  - 关闭

    kill 进程

SSH安全登陆服务（续）

- SSH客户端使用
  - 登录

  与rsh相同

  ```
  # ssh –l yangkun c0101.lenovo
  # ssh c0101.lenovo
  ```

  - 远程复制

  与rcp相同

  ```
  # scp c0101.lenovo .bashrc
  ```

- 有了SSH，就应该关闭telnet，rsh，rlogin!

第一部分　安装Linux操作系统作为服务器软件
第二部分　单主机系统的管理
第三部分　Internet网络服务
第四部分　内部网(Intranet)网络服务
第五部分　Linux操作系统的高级网络功能

lenovo 联想

第四部分 Intranet网络服务

- NFS网络文件系统
- NIS网络信息服务
- Samba服务
- 打印服务
- DHCP动态主机配置协议
- 备份

第四部分 Intranet网络服务

- **NFS网络文件系统**
- NIS网络信息服务
- Samba服务
- 打印服务
- DHCP动态主机配置协议
- 备份

NFS网络文件系统

- **NFS简介**

网络文件系统（Network File System）是Unix操作系统共享网络间的文件和应用程序的方法。它允许用户连接到一个远程硬盘，像操作本地硬盘一样的操作这个远程硬盘。

NFS网络文件系统（续）

- **NFS**原理

  远程过程调用RPC（**R**emote **P**rocedure **C**all）

  负责处理客户和服务器之间的请求，让它们能联系上服务器并且找出需要把调用转给哪一种服务器。当一项服务准备出现在服务器上的时候，需要把自己注册到RPC服务管理器**portmapper**上，再由**portmapper**通知客户实际的服务在服务器的什么位置。

NFS网络文件系统（续）

- 挂装和存取硬盘分区
  - 客户端联系服务器的**portmapper**，确定NFS挂装服务端口
  - 客户端联系挂装服务，申请挂装分区。该服务会检查客户的挂装权限。权限指定：**/etc/exports**文件
  - 客户端再联系**portmapper**，确定NFS服务器所在端口，默认为**2049**
  - 客户对NFS服务器发出申请时，发送一个RPC到NFS服务器
  - 客户操作完成，更新挂装表，但不通知服务器，因为服务器和客户端并不保存对方的状态信息。

NFS网络文件系统（续）

- **NFS**安全性：

  必须做到保证非root用户不能变成root用户！

- **NFS**版本：

  NFS最新版本为3.0，但是Linux只支持到2.0。

NFS网络文件系统（续）

- **NFS**的激活

  检测是否激活：

  ```
  # rpcinfo –p
  ```

  如果在下边返回列表中有**nfs**和**mountd**两项，那么nfs已经在运行中。

```
program   ver    proto   port
100000     2     tcp     111    rpcbind
……         …     …       …      ……
100003     2     udp     2049   nfs
100005     2     tcp     995    mountd
……         …     …       …      ……
```

NFS网络文件系统（续）

- **NFS组件：**

  rpc.statd　守护进程，处理客户和服务器的文件锁定问题

  rpc.quotad　NFS和配额管理的接口。控制用户的磁盘配额

  rpc.mountd　检查用户是否具有挂接权限

  rpc.nfsd　处理NFS请求，是NFS的核心进程

  NFS启动：

  ```
  # /etc/rc.d/init.d/nfs start
  ```

  NFS停止：

  ```
  # /etc/rc.d/init.d/nfs stop
  ```

NFS网络文件系统（续）

- **NFS**的服务器配置：
  - 建立**/etc/exports**文件，该文件定义了服务器的共享规则，比如：共享目录，该目录对某一用户的权限等等。
  - 启动**NFS**服务器进程

/etc/exports文件：

```
/dir/to/export    client1(permissions) client2…
```

表示对客户机client1和client2共享**/dir/to/export**这个目录，这两个客户机的权限由括号内的权限定义来说明。

NFS网络文件系统（续）

- **NFS客户对共享分区的权限**
  - **secure** 客户挂装端口号必须小于1024，默认为激活
  - **noaccess** 客户只能访问**/dir**目录，不能访问**/dir/to**目录，但是可以访问**/dir/to1**目录。
  - **ro** 对该分区只读操作
  - **no_root_squash** 让客户机上root用户能访问NFS挂接的子目录
  - **squash_uids=uid_list** 禁止UID为特定值的用户访问

    squash_uids=4,8-15,45
  - **squash_gids=gid_list** 禁止GID为特定值的用户访问
  - **rw** 给予正常的读写权限

NFS网络文件系统（续）

- **/etc/exports**文件实例

```
#/ect/exports for nfsserver.domain.com
#
/export/data1     node1(rw) node2(ro) node3(rw) \
/export/data2     node1(rw,no_root_squash)
```

- 让NFS服务器重新读取**/etc/exports**文件

  # exportfs –a　（开放exports文件所有设置）

  -r　（重新开放exports文件所有设置）

  -u client（**/dir/to/mount**不向client开放该目录）

  # exportfs –o rw serv1:/export/data1

NFS网络文件系统（续）

- **NFS**常见问题
  - 开放共享后，服务器拒绝客户访问
    - 检查**/etc/exports**文件中客户机是否为完全域名
  - 服务器认为主机名和IP不符
    - 检查**/etc/hosts**文件和**DNS**表，确认这些客户机的主机名和IP是否对应

NFS网络文件系统（续）

- **NFS的客户端配置**

只需要内核编译为支持NFS功能，一般为默认支持，不需要加载其他软件，只需要修改mount命令参数。

```
#mount –o rw,bg,intr,soft  serv1:/export/data1 /data1
```

也可以在**/etc/fstab**文件中指定上述mount参数，比如：

```
serv1:/export/data1 /data1 nfs  rw,bg,intr,soft 0 0
```

- **mount参数：**

bg　后台mount

intr　可中断挂接（当内核存取操作出现问题时，自动放弃操作）

soft　分区软挂接（防止硬挂接不上时的系统停止，视情况而定）

retrans=n　软挂接时最大重试次数

rsize=x　读操作数据块大小（8192为佳）

wsize=x　写数据块大小

NFS网络文件系统（续）

- **NFS**分区常见用途：
  - 保存常用程序，使常用程序可以为网络中所有客户机使用，节约空间，方便管理。
  - 保存用户的登陆子目录，使得可以配置自动挂装程序和**NIS**服务。用户可以登陆到网络中任何一台机器上并且进入自己的目录，实现在不同系统间的切换。
  - 共享的电邮缓冲，可以将全体用户的电子邮件放在一个**NFS**分区里，供所有用户进行访问。

NFS网络文件系统（续）

- **NFS故障排除**
  - 拒绝访问（Permission Denied）
  - 孤立的文件句柄（Stale NFS FileHandle）

NFS网络文件系统（续）

- 小结
  - NFS进程：**rpc.statd、rpc.nfsd、rpc.mountd**和**rpc.quotad**
  - NFS激活查看：**# rpcinfo –p**
  - NFS的开放： **# exportfs –a**
    # exportfs –r
  - NFS不考虑服务器和客户机的运行状态

第四部分 Intranet网络服务

- NFS网络文件系统
- **NIS网络信息服务**
- Samba服务
- 打印服务
- DHCP动态主机配置协议
- 备份

NIS网络信息服务

- **NIS**简介

Network **In**formation **S**ervice（网络信息服务），通常还称为Yellow Pages（黄页），是一种集中管理系统通用访问文件（如**/etc/passwd**、**/etc/group**或**/etc/hosts**）的分布式数据库系统。服务器存放这些文件，而客户端则通过网络访问其中的信息。

在**NIS**中被访问的信息存放在多个文件（**maps**）中。对于位于中心的主控服务器，必须维护所有的文件，而客户端则访问它们。同时还有从属服务器也保存有备份，这些从属服务器可以处理客户端的访问请求，但不允许修改信息。所有对于maps的修改只能在主服务器上进行，然后再由主控服务器分发到各个从属服务器上（**server push**）。

NIS网络信息服务（续）

- **NIS主控服务器的配置**
  - 激活**NIS**服务（可将**ypserv**运行级别设置为3或5）
  - 建立域名
  - 启动**ypserv**守护进程
  - 编辑**Makefile**文件
  - 执行**ypinit**，建立数据库

NIS网络信息服务（续）

- 建立**NIS**域名

  # domainname ssb.lenovo.com

  - RedHat中编辑**/etc/sysconfig/network**文件，加入语句：

    NIS_DOMAIN = ssb.lenovo.com

  - 非RedHat编辑**/etc/rc.d/init.d/ypserv**文件，查找**domainname**语句，如没有，加入：

    Domainname ssb.lenovo.com

- **NIS**的启动和停止

  - **#/etc/rc.d/init.d/ypserv start**
  - **#/etc/rc.d/init.d/ypserv stop**

NIS网络信息服务（续）

- 编辑**Makefile**文件

进入**/var/yp**目录，找到**Makefile**文件

- 指定附属服务器：**NOPUSH**参数，并且编辑**/var/yp/ypservers**和**/etc/hosts**文件，加入附属服务器的主机名
- 设置最小UID和GID值：**MINUID**，**MINGID**。一般不共享root数据项，所以这两个参数不能设置为0。
- 整合shadow口令和常规口令：**MERGE_PASSWD**，一般**ture**。
- 整合shadow分组口令和常规分组口令：**MERGE_GROUP**，一般为**false**，除非有shadow加密用户分组文件。
- 指定文件名:**$(YPPWDDIR)**变量，一般为**/etc**

```
PASSWD = $(YPPWDDIR)/passwd
```

- 共享哪些数据：all参数，不共享的映射前边加#

```
all:passwd group hosts rpc mail\
    #shadow networks ethers
```

NIS网络信息服务（续）

- **ypinit**命令（初始化NIS服务器）
  - 使用前必须设置好NIS服务的域名。

    #/usr/lib/yp/ypinit –m(主控服务器)

    ……

    next host to add: ssb.lenovo.com（主控）

    next host to add: server1（附属）

    next host to add: server2（附属）

    ……
  - 所有信息保存在**/var/yp/ypservers**文件中，并且自动运行make程序，建立相应的映射关系。

NIS网络信息服务（续）

- **NIS**服务器调试
  - **Makefile**错误信息

    一般就是说找不到要映射的文件，建立该文件或者编辑Makefile文件就可以解决，重新make即可。

  - 更新**NIS**共享关系

    ```
    # cd /var/yp
    # make
    ```

NIS网络信息服务（续）

- **NIS**的客户端配置
  - 编辑**/etc/yp.conf**文件
  - 设置启动脚本
  - 编辑**/etc/nsswitch.conf**文件

- **/etc/yp.conf**文件
  - 使用广播

    加入**domain domainname broadcast**一句

    每个子网至少有一台NIS服务器
  - 直接定义服务器的主机名

    domain mydomainname server servername

NIS网络信息服务（续）

- 配置**NIS**客户
  - 设置启动脚本

    # /etc/rc.d/init.d/ypbind start

    # /etc/rc.d/init.d/ypbind stop
  - **/etc/nsswitch.conf**文件

    格式： ***filename : servicename***

    服务： files ; yp ; nis ; dns ; nis+ ;
    NOTFOUND=return

    示例： passwd: files nis
  - 测试NIS客户配置情况

    # ypcat passwd

NIS网络信息服务（续）

- 配置附属NIS服务器
  - 设置域名

    # domainname lenovo

    /etc/sysconfig/network中NIS-DOMAIN变量
  - 设置主控服务器对辅助服务器的推操作

    /etc/yp/ypservers文件中列出相应的辅助服务器名单。

    Makefile文件中设置NOPUSH=false
  - 执行ypinit命令

    # /usr/lib/yp/ypinit –s master

    # /etc/rc.d/init.d/ypserv start

NIS网络信息服务（续）

- **NIS软件工具**
  - **ypcat**

    给出NIS共享映射的全部关系
  - **ypwhich**

    返回本操作应答NIS服务器的名称
  - **ypmatch**

    通过用户给定的一个关键字查处与之对应的共享映射关系数据项

    ```
    # ypmatch sshah passwd
    ```

NIS网络信息服务（续）

- 在配置文件中使用**NIS**
  - 共享**/etc/passwd**文件
- 在实际网络中实现**NIS**
  - 小型网络
    一个NIS主控服务器
  - 带分支的网络
    一个NIS主控服务器加多个辅助NIS服务器
  - 规模庞大的网络
    多个NIS主控服务器

第四部分 Intranet网络服务

- NFS网络文件系统
- NIS网络信息服务
- **Samba服务**
- 打印服务
- DHCP动态主机配置协议
- 备份

Samba服务

- 基本原理

  Session Message Block 任务消息块

  Linux操作系统与Windows的基本问题

  - 用户名和口令
  - 加密口令
  - 守护进程**smbd**(139端口)与**nmbd**(137端口)之间的区别

    smbd守护进程负责为客户处理具体的文件、打印服务

    nmbd守护进程负责处理NetBIOS域名服务器申请

Samba服务（续）

- 编译安装**Samba**

```
# tar –xzf samba-2.0.5a.tar.gz
# cd samba-2.0.5a/source
# ./configure –with-smbmount –with-pam
# make
# make install
# cd ../examples
# cp smb.conf.default /usr/local/samba/lib/smb/conf
# cd /usr/local/samba
# chmod –R 755 bin lib man var
```

Samba服务（续）

- 设置**SWAT**工具

  Samba **W**eb **A**dministration **T**ool

  SWAT不需要依赖于Web服务器，通过**inetd**守护进程运行实现

  # chmod –R 755 /usr/local/samba/swat

  /etc/services文件中加入

  ```
  swat  901/tcp
  ```

  /etc/inetd.conf文件中加入

  ```
  swat stream tcp nowait . 400 root /usr/local/samba/bin/swat swat
  ```

  # kill –l `ps –C inetd|awk ‘ {if(/inetd/)print $1}’`

Samba服务（续）

- **Samba**系统管理
  - 启动和终止Samba

    启动：

    ```
    # /usr/local/samba/bin/smbd –D
    # /usr/local/samba/bin/nmbd -D
    ```

    终止：

    ps命令列出所有Samba进程，找出**smbd**进程kill之。

Samba服务（续）

- 使用**SWAT**工具
  - SWAT软件的用户界面
  - 建立共享关系
- 使用**smbclient**程序
  - 浏览服务器

```
# smbclient –L lenovo
```

  - 远程文件访问

```
# /usr/local/samba/bin/smbclient //lenovo/tools
```

  - 远程打印机访问

```
# smbclient //ssb-server/lp -p
```

Samba服务（续）

- 使用**smbmount**命令

```
# smbmount //lenovo/tools /mnt/lenovo
# umount /mnt/lenovo
```

- 对加密口令功能的支持

```
# cd /usr/local/samba/private
# ../bin/mksmbpasswd.sh < /etc/passwd > smbpasswd
# chmod 500 . ; chmod 600 smbpasswd
```

  - 允许使用**NULL**口令
  - 使用**smbpasswd**命令修改口令

Samba服务（续）

- 检查并排除**Samba**故障
  - 重启动Samba服务

    修改配置之后，重启服务才能使修改生效。
  - 检查配置参数选项是否正确

    smb.conf文件中及时通知用户变更情况。
  - 检查加密口令

    利用Samba源码中提供的**regedit**脚本程序禁止客户端加密。
  - 检查被缓冲保存的口令

    删除Windows中被缓冲起来的**pwl**文件。

第四部分 Intranet网络服务

- NFS网络文件系统
- NIS网络信息服务
- Samba服务
- 打印服务
- DHCP动态主机配置协议
- 备份

打印服务

- **lpd**守护进程的基本知识

  line **p**rinter **d**aemon 线性打印机守护进程

- Linux中打印过程
  - 打印作业放入缓冲池目录**/var/spool/lpd/printername**
  - 在**/etc/printcap**文件中查找该打印机的配置信息
  - 根据需要将作业发往打印过滤器
  - 对于直接连接打印机的服务器，直接将作业发给打印机
  - 对于远程连接打印机的服务器，与对应的**lpd**服务联系

打印服务（续）

- 启动**lpd**守护进程

  # ps –auxw

  查看哪些进程正在运行中

  如果没有启动lpd，用户自己创建**/usr/sbin/lpd**的链接

- 允许远程用户
  - 设置允许访问的打印客户主机

    /etc/hosts.lpd 或 **/etc/hosts.equiv**中列出
  - 对所有用户开放

    删除**/etc/hosts.lpd** 和 **/etc/hosts.equiv**

打印服务（续）

- 配置**/etc/printcap**文件

```
printname1|printname2|printname3…:\
   :command=value:\
   :command=:\
   :command=value:
lenovoprint|lp:\
   :sd=/var/spool/lpd/lp:\
   :sh:\
   :rm=intrepid:\
   :rp=engprint
```

打印服务（续）

- 使用Samba服务打印

  lpd调用一个过滤器

  -w width –l length –I indent –n login –h host accounting-file

- 让修改生效

  查找lpd守护进程的PID，终止后重启

  # ps –C lpd

  # kill xxxx ; /usr/bin/lpd

  或者

  # /etc/rc.d/init.d/lpd restart

打印服务（续）

- **lpd客户端工具**
  - **lpr**

    打印命令

    ```
    # lpr filename
    # lpr -P printername filename
    ```
  - **lprm**

    删除队列中未开始打印的作业

    ```
    # lprm -P lenovo 77
    ```
  - **lpq**

    按照即将打印的顺序列出队列中的作业

第四部分 Intranet网络服务

- NFS网络文件系统
- NIS网络信息服务
- Samba服务
- 打印服务
- **DHCP动态主机配置协议**
- 备份

DHCP动态主机配置协议

- 基本原理

  Dynamic **H**ost **C**onfiguration **P**rotocol

  动态主机配置协议

  客户端以DHCP申请的形式从服务器获得地址和相关设置

- 下载、编译和安装DHCP服务器

```
# tar –xzf dhcp-3.0b1p10.tar.gz
# ./configure
# make
# make install
```

DHCP动态主机配置协议（续）

- 配置**DHCP**服务器
  - **/etc/dhcpd.conf**
    - 一套定义集
    - 一套参数集

```
Global parameters;
Declaration1{
   [parameters related to declaration1]
   [nested sub declaration]
}
```

  - **dhcpd.lease**文件中纪录已经签发的地址

DHCP动态主机配置协议（续）

- **DHCP**客户端守护进程
  - 下载、编译和安装DHCP客户
  - 配置DHCP客户

    # dhcpcd
  - 参数

```
dhcpcd [-dkrDHR] [-t timeout] [-c filename]
[-h hostname] [-I vendorClassID] [-I
clientID] [-l leasetime] [interface]
```

第四部分 Intranet网络服务

- NFS网络文件系统
- NIS网络信息服务
- Samba服务
- 打印服务
- DHCP动态主机配置协议
- 备份

备份

- 评估备份需求
  - 有多少数据需要备份
  - 使用什么类型的硬件完成备份操作
  - 需要支持多大的网络流量
  - 数据恢复要求多快时间完成

备份（续）

- 管理备份设备和文件

  /dev/stX　　自动回绕SCSI磁带驱动器

  /dev/nstX　　非自动回绕SCSI磁带驱动器

- 使用**mknod**命令建立设备文件

  # mknod /dev/st0 c 9 0

  # mknod /dev/nst0 c 9 128

  9表示SCSI磁带驱动器的主编号

  0到15代表编号为0到15的自动回绕驱动器

  128到143代表编号为0到15的非自动回绕驱动器

备份（续）

- 使用**mt**命令操作磁带设备

  从备份的角度看，mt命令最适合作为回绕和检索机制

  - 回绕/dev/nst0设备中的磁带

    ```
    # mt –f /dev/nst0 rewind
    ```

  - 移动读写磁头，准备读取磁带上的第三个文件

    ```
    # mt –f /dev/nst0 asf 2
    ```

  - 参数：

    ```
    -f tape-device ; fsf count ; asf count ;
    rewind ; erase ; status ;
    offline ; load ; lock ; unlock
    ```

备份（续）

- 命令行工具程序
  - **dump**

    level 0: 完全备份

    level 1: 相对于level 0的增量式备份

    level n: 相对于level n-1的增量式备份(最高 n=9)

对/dev/hda1文件系统以备份级别0备份到/dev/st0设备上

```
# dump –0 –f /dev/st0 /dev/hda1
```

压缩备份并重定向输出到磁带设备

```
# dump –0 –f - /dev/hda1 | gzip –fast –c > /dev/st0
```

备份（续）

- 备份整个系统

```
# mt –f /dev/nst0 rewind
# dump -0uf - /dev/hda1 | gzip –fast –c > /dev/nst0
# dump -0uf - /dev/hda3 | gzip –fast –c > /dev/nst0
# dump -0uf - /dev/hda5 | gzip –fast –c > /dev/nst0
# dump -0uf - /dev/hda6 | gzip –fast –c > /dev/nst0
# mt –f /dev/nst0 rewind
# mt –f /dev/nst0 reject
```

备份（续）

- 命令行工具程序
  - **restore**

    ```
    # restore –ivf /dev/st0
    ```

  - **tar**

    适合对文件操作

- 系统完全恢复操作

```
# mke2fs /dev/sda1 ; mount /dev/sda1 /home ; cd /home
# restore –rf /dev/st0
# gzip –d –c /dev/st0 | restore –ivf -
```

第一部分　安装Linux操作系统作为服务器软件
第二部分　单主机系统的管理
第三部分　Internet网络服务
第四部分　内部网(Intranet)网络服务
第五部分　Linux操作系统的高级网络功能

lenovo 联想

第五部分 Linux操作系统的高级网络功能

- 系统管理员使用的TCP/IP
- 网络配置
- Linux的高级网络功能
- /proc文件系统

第五部分 Linux操作系统的高级网络功能

- 系统管理员使用的TCP/IP
- 网络配置
- Linux的高级网络功能
- /proc文件系统

系统管理员使用的TCP/IP

- TCP/IP的分层结构

  TCP/IP是采用分层结构来建立的，因此它称为TCP/IP栈

  - 都有哪些TCP/IP层？
  - 各个层相互之间有什麽样的关系？
  - 为什麽它们与ISO的7层OSI模型不一致？

系统管理员使用的TCP/IP（续）

- TCP/IP的分层结构
  - **数据包** 在分层系统的最低层是网络要处理的最小的数据单元，它既包括了我们要在各个系统之间传输的数据，也包含了一些控制信息，用于帮助网络传输机制确定数据包应该发送到什么地方去。

    数据包是按照协议来分层的。每个协议都使用一个首标来描述需要将数据从一个主机传输到另一个主机的信息。数据包首标很小，数据帧的其余1446个字节都是数据。

以太网首标	IP首标	TCP首标	数据（有效负载）
14字节	20字节	20字节	1446字节

系统管理员使用的TCP/IP（续）

- TCP/IP的分层结构

  TCP/IP的层次与OSI模型之间的关系

  第一层：物理层。用于描述数据传输所经过的实际介质

  第二层：数据链路层。用于描述以太网协议

  第三层：网络层（IP协议层）。负责通过多个子网络将数据包从一个主机传输到另一个主机

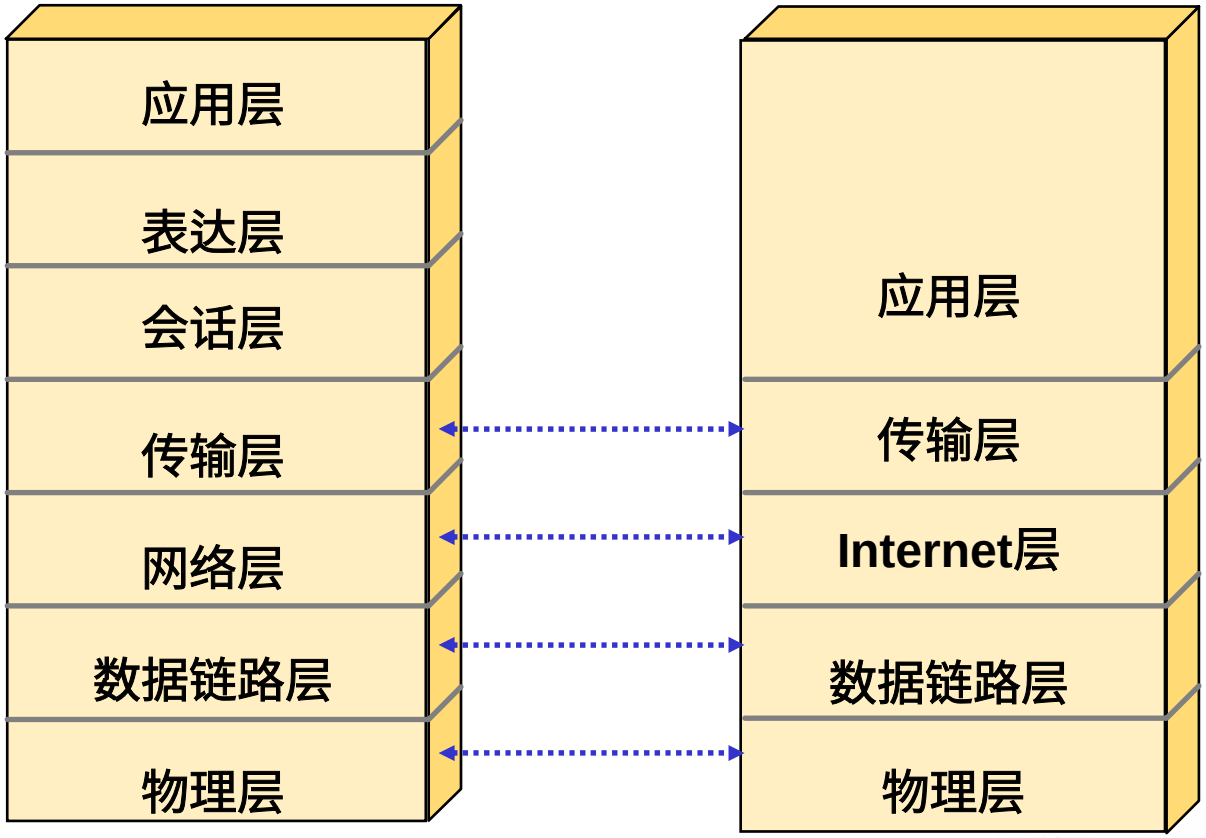
  第四层：传输层。负责从一个会话到另一个会话的可靠传输。

  第五至七层：应用层。

系统管理员使用的TCP/IP（续）

- TCP/IP的分层结构

  TCP/IP的层次与OSI模型之间的关系



系统管理员使用的TCP/IP（续）

- TCP/IP的各种协议
  - **以太网协议** 以太网协议很简单，它只负责将数据包从LAN（局域网）上的一个主机传输到LAN上的另一个主机。

    以太网没有全球网络的概念，因为它受到数据包传输时间的限制，同时也要受单个网段上可以存在的主机数量的限制。
  - **IP** Internet协议（IP）对于它周围的环境来说显得更聪明些。IP数据包能够传输到与同一个IP网络连接的任何其他主机，只要与该主机之间存在一个路由。

    但是，IP本身只能保证设法将数据包传输到它的目的地去。这就是说数据包可能不按照原来的顺序进行传输，也可能在传输中出现很长的延迟，甚至丢失数据包。

    此外，IP只知道如何将数据包传输到另一个主机。当数据包到达主机时，IP首标中没有任何信息来告诉它数据要传递给哪个应用程序。

系统管理员使用的TCP/IP（续）

- TCP/IP的各种协议
  - **TCP**　传输控制协议（TCP）使用IP来将数据包从一个主机发送到另一个主机。 TCP的另一个重要概念是端口号，它使得数据可以在特定的应用程序之间传输，而不只是从一个主机传输到另一个主机。

    TCP具备的最重要的特性是它采用了连接的思路。但是，它提供的一组功能是以牺牲速度为代价的。对连接信息进行管理，重新发送数据包，并且对数据包自动进行重新排序，都需要占用时间。
  - **UDP**　用户数据包协议（UDP）使用IP来将数据包从一个主机发送到另一个主机。但与 TCP不同的是， UDP不能保证数据包的传输的顺序，也不支持连接的概念。它给IP增加的是端口的思路，这样应用程序之间就能够互相进行通信，从而使得UDP的运行速度比TCP快，并且可以用于允许丢失数据包的传输环境。 UDP也提供了一个校验和，使得数据包在到达目的地时可以检查其数据的完整性。UDP的一个非常流行的用途是传输流式媒体。

系统管理员使用的TCP/IP（续）

- TCP/IP的各种协议
  - **ICMP**　因特网控制消息协议（ICMP）专门用于根据网络的情况使一个主机能够与另一个主机进行通讯。由于数据只是由操作系统使用，而不是用户使用，因此ICMP不支持端口号码、可靠的数据传递或者保证数据包的顺序等概念。

    每个ICMP数据包都包含一个消息类型的字段，它告诉接收者该消息属于什麽性质。

系统管理员使用的TCP/IP（续）

- 首标

以太网上的TCP/IP数据包是由每个协议的一系列首标和随后的要发送的实际数据组成的。数据包首标的作用只不过是告诉协议如何处理数据包的那些信息而已。

- 以太网首标

有两种以太网首标：802.3首标和Ethernet II首标

系统管理员使用的TCP/IP（续）

- Ethernet II首标

  以太网首标包含3个输入项，即目的地址、原地址和数据包的协议类型。

  以太网地址也称为MAC（介质访问地址）地址。它是个48-位（6个字节）的数，用于对世界上的每个以太网网卡进行独一无二的标识。

  数据包的协议类型是个2字节的值，它负责告诉我们该数据包被传递到接收端上的什麽协议。若是IP数据包，这个值是十六进制数0800。

  Ethernet II数据包通常称为以太网数据包。

系统管理员使用的TCP/IP（续）

- 802.3首标

802.3数据包的格式与Ethernet II数据包的格式略有不同。它的目的MAC地址和源MAC地址不变，但它们不是告诉我们数据包应该传递到接收端上的什麽协议，而是告诉我们它的长度。

区分这两种类型的数据包的方法是，如果协议类型的值小于1500，那麽这个以太网数据包实际上是个802.3数据包。

查看以太网首标：运行下面的命令

```
# tcpdump -e
```

系统管理员使用的TCP/IP（续）

- IP首标

  IP首标的第一个值是版本号（4位）。现在最常用的版本是版本4（IPv4）。

  首标本身的长度（4位）。我们必须知道首标的长度，因为在基本首标的结尾处还附加了一些选项参数。

  服务程序的类型（TOS）（8位）。它告诉IP协议栈，应该对数据包进行什麽类型的处理。

  总长度（16位）的值可以告诉你完整的数据包究竟是多长，包括IP首标和TCP首标，但不包括以太网首标。这个值以字节为单位。IP数据包的长度不能超过65536个字节。

系统管理员使用的TCP/IP（续）

- IP首标（续）

  标识号（16位）是一个独一无二的号码，用于标识一个特定的数据包。IP使用这个标识昊对分段的数据包进行组装。

  标记（3位）负责告诉我们该数据包是否进行了分段。分段是在IP数据包的长度大于两个主机之间的最小MTU(最大传输单元)是进行的。

  数据段位移(13位)的值是指我们接收的是整个数据包的那个部分。

  生存时间（TTL）（8位）字段是0到255之间的一个数字，用于标识数据包在被丢弃之前可以在网络上存在多长时间。

系统管理员使用的TCP/IP（续）

- IP首标（续）

  协议（8位）字段负责告诉我们该数据包该传递到哪个高层协议。这个值通常设置为TCP、UDP或ICMP。

  首标校验和（16位）是IP首标中的最后一个比较小的值。这个字段用于存放IP首标中的每个字节的和，包括所有选项的值。

  源IP地址（32位）和目的IP地址（32位）是IP数据包中最重要的数。

系统管理员使用的TCP/IP（续）

- TCP首标

TCP首标与IP首标很相似。

TCP开头的两个信息是源端口号（16位）和目的端口号（16位）。他们的值的范围是0到65535。通常源端口号的值要大于1024。

TCP首标中的下面两个号码是序号和确认号。这两个值供TCP用于保证数据包传输时的正确顺序，并且使发送端能够知道哪些数据包已经正确地接收到了。

TCP首标的长度（4位）将告诉我们首标有多长，这个长度包括了所有的TCP选项。

下面这个字段有点儿复杂。TCP使用一系列的标记来指明某个数据包是启动一个连接，还是包含数据，或是结束连接。这些标记是：Urgent(URG，紧急)，Acknowledge(ACK,确认)， Push(PSH,迫使)， Reset(RST，撤销连接)，Synchronize (SYN，同步)，Finish(FIN,结束)。

系统管理员使用的TCP/IP（续）

- TCP首标（续）

TCP首标的下一个项目是窗口大小（16位）。TCP使用一种称为滑动窗口的技术，使连接的每一端能够告诉另一端，它还有多少缓存空间可以用来处理这个连接上传输的信息。

TCP首标的下一个元素是校验和（16位）。它的作用是为接收端提供一种方法，以便检验接收到的数据是否遭到了破坏。与IP校验和不同的是，TCP校验和运行时既要考虑TCP首标，也要考虑发送的数据。

TCP首标的最后一个元素是紧急指针（16位）。这个值是在设置了URG标记时查看的值，用于告诉接收端的TCP协议栈，一些非常重要的数据是从紧急指针指向的位置开始的。TCP协议栈要把这个信息转发给应用程序，这样它就知道它应该将这些数据视为特别重要的数据。

另外，如果一个数据包使用了URG位，那麼你就必须查看这个数据包。

系统管理员使用的TCP/IP（续）

- UDP首标

  UDP首标的第一个字段是源端口号（16位）和目的端口号（16位）。它们都是TCP端口号。

  UDP首标中的下一个字段是数据包的长度（16位）。

  最后一个字段是UDP校验和。它供UDP用来检验到达的目的地的数据是否被破坏。

系统管理员使用的TCP/IP（续）

- 建立一个完整的TCP连接

  打开连接：

  要打开一个连接，TCP要通过一个“三次握手”的过程。目的是要使连接的两端能够互相发送它们的状态信息，并且使对方有机会确认数据已经接收到了。

  第一个数据包是由想要打开与服务器的连接的主机发送的。

  第二个数据包是从服务器发送给客户机的数据包。

  第三个数据包是从客户机发送给服务器的。

系统管理员使用的TCP/IP（续）

- 建立一个完整的TCP连接（续）

  传输数据：

  完整的连接建立好以后，双方就可以发送数据了

  关闭连接：

  可以选择使用强制关闭的方式来关闭TCP连接。连接的一端告诉另一端“现在停止发送数据包”强制关闭的方式要使用RST（撤销连接）标记来实现，设置RST标记后，接收端对收到的数据将不进行确认。

系统管理员使用的TCP/IP（续）

- ARP如何运行

地址解析协议（**ARP**）是IP用来将以太网地址映射为IP地址时所用的一个机制。其运行的基本步骤如下：

1、 客户机查看它的ARP缓存，以了解它里面是否存在IP地址与它的以太网地址之间的对应关系。

2、 如果没有找到对应于所需要的IP地址的以太网地址，那麼就发送一个广播数据包，请求拥有这个IP地址的人做出应答。

3、 若拥有该IP地址的主机是在LAN上，它将对该ARP请求做出应答，通知发送端它的以太网地址/IP地址的组合。

4、客户机将这个信息保存在它的缓存中，准备建立一个谣传输的数据包 。

第五部分 Linux操作系统的高级网络功能

- 系统管理员使用的TCP/IP
- 网络配置
- Linux的高级网络功能
- /proc文件系统

网络配置

- 模块和网络接口卡

Linux操作系统中的网络设备打破了它通过文件访问所有设备的传统。在网络驱动程序初始化网络接口卡并向内核注册它自己之前，没什麽机制可以访问该网卡。

根据内核配置的不同，网络接口卡的设备驱动程序可能已经被便以为一个模块。

如果已经把驱动程序配置为一个模块，并且还设置好了模块的自动加载功能，下一步就需要在**/etc/conf.modules**文件里告诉内核设备名和准备加载的模块两者之间的映射关系，例如：

```
# alias eth0 eepro100
```

这表示eth0网卡使用的是eepro100驱动程序模块。

网络配置（续）

- **ifconfig程序**

ifconfig程序负责对网络接口卡（**NIC**）进行具体的设置，它的全部操作都可以通过命令行参数来实现。

- **简单用法**

ifconfig程序最简单的使用方法只需要输入网卡的设备名及其IP地址，ifconfig程序将会根据IP地址的情况（A或B或C类）推导出子网掩码及广播地址的信息。

如果设置的IP地址没有严格按照IP地址的类型设置子网掩码，那就需要在命令行上明确的指定：

```
# ifconfig dev ip netmask nmask broadcast bcast
```

提示：不带任何参数的ifconfig程序将列出全部运行的以太网设备，加-a参数将列出全部已驱动的网络设备，而不论它是否在运行。

网络配置（续）

- **ifconfig程序（续）**
  - **在RedHat下设置网卡**

在redhat发行版本中，系统开机引导时会去读取**/etc/sysconfig/network-scripts**目录里的有关配置文件，对网卡的ip地址进行设置。我们可以手动编辑这些文件，对网卡进行配置。系统上的每一个网卡都在上述目录中有一个对应的ifcfg文件，这个文件名的后缀是相应的设备名。如ifcfg-eth0对应eth0设备，ifcfg-eth1是eth1设备

文件中与ip协议相关的设置项如下所示：

```
DEVICE=“ethx”                    #设备名
IPADDR=“xxx.xxx.xxx.xxx”         #ip地址
NETMASK=“xxx.xxx.xxx.xxx”        #子网掩码
NETWORK=“xxx.xxx.xxx.xxx”        #网络地址
BROADCAST=“xxx.xxx.xxx.xxx”      #广播地址
ONBOOT=“yes”                     #是否在开机时启动这个网络设备
```

网络配置（续）

- **ifconfig程序（续）**
  - **其他参数**

ifconfig命令的语法格式：

```
# ifconfig device address options
```

部分options：

up	激活该设备
down	停用该设备
arp	激活此设备响应arp请求（缺省设置）
-arp	禁止此设备响应arp请求
mtu value	把该设备的最大传输单元设置为value。对于以太网，缺省值为1500。对于千兆以太网卡，我们可以设为9000
netmask address	设置子网掩码为address，不给出按ip类别自动设置
broadcast address	设置广播地址为address，不给出按ip类别自动设置

网络配置（续）

- 使用**route**

如果在网络中存在多个子网时，就会需要路由器。路由器安放在各个网络之间，把数据包重新定向到它们真正的目的地取。

一个典型的linux主机知道三个路由：第一个是**回跳路由**，它简单的指向回跳设备；第二个是指向本地**局域网的路由**，这样使发送给同一局域网中其他主机的数据包能够直接发送给它们；第三个就是**缺省路由**，这个路由供那些要离开本地局域网与其他网络进行通信的数据包使用。

当我们在一个主机中安装了多个网卡，每个网卡分别连接到不同的网络，这时我们必须手工添加路由，这样才能保证根据目的地的地址把数据包发送给正确的网络。

网络配置（续）

- 使用**route**（续）
  - **简单用法**

典型route命令的语法格式：

 # route cmd type addy netmask mask gw gway dev dn

各个参数的含义如下：

cmd	可取值是add或者del，要看是准备添加一个路由还是准备删除一个路由。如果是删除，其他必选参数只有addy了。
type	可取值是-net或者-host，要看addy参数选项表示的是网络地址还是路由地址。
addy	准备对它提供路由的目的网络
netmask mask	把addy地址处的子网掩码设置为mask。
gw gway	把到addy地址去的路由地址设置为gway。通常用于设置缺省路由
dev dn	把目的地是addy的所有数据包通过由ifconfig程序设置的网络设备dn发送出去。

网络配置（续）

- 使用**route**（续）
  - **显示路由**

显示自己的路由表有两种办法：route或netstat命令

1、**route**命令

运行不带参数的route命令就可显示路由表，如下所示：

```
# route
Kernel IP routing table
Destination  Gateway  Genmask        Flags  Metric  Ref  Use  Iface
10.10.2.0    *        255.255.255.0  UH     0       0    0    eth1
192.168.1.0  *        255.255.255.0  U      0       0    0    eth0
127.0.0.1    *        255.0.0.0      U      0       0    0    lo
default      *        0.0.0.0        UG     0       0    0    eth0
```

网络配置（续）

- 使用**route**（续）

1、**route**命令（续）

上表中部分项目的含义：

Flags 表示连接状态的符号，每个字母代表一种情况：

U：代表成功

H：目的地是主机

G：目的地是网关

Metric 路由的“成本”，通常使用路程段个数来计算。这表明系统有多条路径通向同一个目的地，但其中一条比其他路径能够更快到达。Metric计量数值比较低的路径通常更受欢迎。

Ref 此路由被引用的次数。Linux操作系统的内核不使用这项信息。把它列在这里是因为route工具程序是能够跨操作平台使用的。因为其他种类的操作系统确实会用到它，所以在这里也就把它列出来了。

Use 路由缓冲区检索成功的次数。如果想看到这个数值需要使用-F参数。

网络配置（续）

- 使用**route**（续）
  - **显示路由**

2、**netstat**命令

正常情况下，**netstat**程序被用来显示一台主机上全部网络连接的运行状态。但是，加上-r参数之后，它就可以用来显示内核的路由表了。必须注意的是，其他大多数基于UNIX的操作系统都要求使用这个方法来查看路由。

第五部分 Linux操作系统的高级网络功能

- 系统管理员使用的TCP/IP
- 网络配置
- **Linux的高级网络功能**
- /proc文件系统

Linux的高级网络功能

在这部分我们将讨论Linux操作系统网络技术的三个特定方面：**IP假名、数据包过滤和IP地址欺诈**。

IP假名说的是一个网络接口使用多个IP地址的能力；数据包过滤技术允许使用者根据各种条件（比如源地址和目的地址等）选择是否接受某个数据包；IP地址欺诈是网络地址翻译功能（network address translation，NAT）的一个应用，它允许用户只使用一个IP地址来代表整个网络。

这些功能都是由操作系统内核所支持的，但它们需要由用户使用工具程序进行必要的配置，这里我们将介绍一些供用户使用的工具程序。需要注意的是，如果自行编译过操作系统的内核，需要把以上功能的支持加到编译操作的配置中去；这些功能在内核配置工具中“Networking”菜单里。

Linux的高级网络功能（续）

- IP假名技术

Linux操作系统可以只使用一个网卡就对多个IP地址做出响应。

设置IP假名功能的具体做法是在ifconfig命令中使用一个特殊的设备名：在网卡真实的设备名（比如eth0）后加上一个冒号和数字作为后缀（比如eth0:0），还可以继续设置eth0:2、eth0:3等等。缺省情况下，操作系统内河允许每个设备最多设置127个IP假名。

ifconfig命令行的其余部分和设置普通设备没什么不同，如下例：

```
# ifconfig eth0:0 10.1.1.2 netmask 255.255.255.0 broadcast 10.1.1.255
```

Linux的高级网络功能（续）

- 数据包过滤
  - 数据包过滤的实现

每一个在IP网络中传输的数据包都有一个IP首标，如果这个数据包同时还是一个TCP或者UCP数据包，它还会有对应于这些协议的首标。在IP首标中包含了源/目的地IP地址，TCP首标中包含了源/目的地端口编号。

如果我们希望能够限制某些IP地址对我们主机的访问或是希望限制某些IP对我们主机的某些应用程序的访问，我们可以通过对数据包中的IP首标或是TCP首标的观察来判断这些数据包是否来自于我们的限制对象，由此来决定是否接受这个数据包，从而实现了对于数据包的过滤。

在Linux2.2内核中使用**ipchains**工具设置实现数据包过滤，而在2.4内核中使用的软件工具被称为**iptable**。

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件

1、编译和安装ipchains工具软件

在这里我们假设使用的版本为ipchains-1.3.10。首先在**/usr/local/src**目录对ipchains进行解包操作：

```
# tar –xvzf ipchains-1.3.10.tar.gz
# cd ipchains-1.3.10          //进入目录ipchains-1.3.10
# make all                    //进行编译
# make install                //将执行程序安装到/sbin
# tar -xvzf ipchains-scripts-1.1.2.tar.gz  //解包ipchains脚本程序
# cp ipchains-restore ipchains-save ipchains-weapper /usr/bin
                              //复制脚本程序到/usr/bin目录
```

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件

ipchains可以管理和操控对数据包进行过滤的那些数据关规则。输入数据关是对从网络进入本系统的数据包进行检验的一组规则；转发数据关是对途径本Linux系统（而本系统又被配置为路由器时）的那些数据包进行检验的数据关；输出数据关是对从本系统向外发出的那些数据包进行检验的数据关；用户自定义数据关可以对输入、转发或者输出数据包进行检验。

ipchains工具在设置数据关时首先需要指明对数据关进行什么样的操作，所以操作是ipchains的第一个参数，下面列出的是ipchains合法有效的参数：

-A	在数据关末尾添加一个或多个规则
-D	在数据关中删除一个或多个规则
-R	在数据关中替换一个规则
-I	在给定编号选中的规则中插入一个或多个规则

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件（续）

-L	列出数据关中设置的全部规则
-F	清空数据关中设置好的规则
-Z	对ipchains中用于计数目的的所有计数器清零
-N	建立一个新的数据关
-X	删除一个数据关，删除前必须使用-F操作清空数据关
-P	把某个给定的数据关设置为缺省的检验状态

在操作之后需要向ipchains指明的是对哪个数据关进行操作，所以ipchains的第二参数应该是数据关的名字，缺省的三个数据关是：输入数据关、转发数据关和输出数据关。在数据关名称之后根据操作的不同将会采用不同的参数，下面是几种参数类型的定义：

Rulespec	规则定义

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件（续）

Rulenum　准备对之进行设置的规则的编号

Target　当某个数据包匹配了某个规则时采取的动作。合法的target有：ACCEPT（接受）允许数据包过关；DENY（拒绝）丢弃数据包，系统假装从来没有收到它；REJECT（丢弃）是一种比较温和的DENY操作，拒绝接受该数据包，同时会向发信方返回ICMP消息。

Option　可以用在每个规则里的附加可选参数

下面，我们看看各种操作需要用到的参数，如下所示：

增加一个规则：

```
# ipchains –A chain rulespec options
```

删除一个规则：

```
# ipchains –D chain rulespec options
```

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件（续）

替换一个规则：

```
# ipchains -R chain rulenum rulespec options
```

插入一个规则：

```
# ipchains -I chain rulenum rulespec options
```

列出数据关中的全部规则：

```
# ipchains -L chain options
```

清空一个被选中的数据关：

```
# ipchains -F chain options
```

清空一个数据关所有的计数器：

```
# ipchains -Z chain options
```

建立一个新数据关：

```
# ipchains -N chain options
```

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件（续）

删除一个数据关：

```
# ipchains –X chain options
```

设置一个给定的数据关对数据包的检验策略：

```
# ipchains –P chain target options
```

规则定义由下列这些命令行参数组成：

-p protocol	定义对那些类型的IP数据包起作用
-s source/mask port	定义匹配什么样的源IP地址。source是IP地址，mask是子网掩码，port是端口号
--source-port port	在没有指定源IP地址的情况下指定端口
-d destination/mask port	定义匹配什么样的目的IP地址
--destination-port port	在没有指定目的IP地址的情况下指定端口
--icmp-type typename	对类型名为typename的ICMP数据包设置一个规则
-j target	定义规则匹配时要采取的行动为target

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件（续）

-i interface　　定义此规则将对哪个接口设备起作用

-f　　设置此规则作用于任何数据包分段

用在规则里的附加可选参数有：

-b　设置本规则双向起作用

-v　操作解释输出

-n　给出数字方式的输出

-l　让操作系统内核记录下所有匹配规则的数据包

-x　使用-L操作动作打印规则设置情况时给出准确的数字

-y　只匹配那些SYN位被置位的数据包，ACK或FIN位被置位的数据包不在考虑范围内

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **ipchains**工具软件（续）

2、使用ipchains工具软件（续）

为了更好的理解ipchains的各种操作动作核与之对应的参数之间的关系，我们看看下面的几个例子：

- 设置这样一个规则：来自192.168.1.8的连接将被丢弃，但是我们允许从本地对它们建立一个连接，使用命令如下所示：

```
# ipchains –A input –p tcp –s 192.168.1.8 –j DENY -Y
```

- 我们打算阻断任何人对8080端口的连接，并且还将在系统记录中记下所有的尝试性连接，使用的命令如下：

```
# ipchains –A input –p tcp –destination-port 8080 –j DENY –l
```

- 我们打算阻断除端口123之外来自ntp.ucsd.edu的所有数据包。

```
# ipchains –A input –p tcp –s ntp.ucsd.edu 123 –j ACCEPT
# ipchains –A input –p tcp –s ntp.ucsd.edu –j DENY
```

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **iptables**工具软件

1、ipchains与iptables之间的差别

两者之间大多数的差别是内部的差别，这使得它们之间的转换变得容易的多了。但是，它们之间最大的差别是，如何根据数据包是发送给主机还是只进行转发这些情况来运用各个规则。

首先，字母的大小写变了，3个核心数据关INPUT、FORWARD和OUTPUT都是大写。用户添加得数据关使用小写字母。

应该注意得第二个问题是，并不是所有得数据包都要经过所有得数据关。在ipchains中，被转发得数据包必须遵循与送往主机得本地数据包所使用得所有相同得输入和输出规则。而对于iptables来说，如果Linux系统作为路由器来运行，则不必遵循这个规定，可以根据数据包是发给主机还是转发分别配置规则。这样可以允许人们更加自由得访问你的网络中的某个主机，而使其他任何人无法访问路由器本身。

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **iptables**工具软件（续）

2、下载和安装iptables

首先从**http://netfilter.filewatcher.org**将iptables-1.1.2.tar.bz2下载到机器上，然后进行解包：

```
# tar –user-compress-program bzip2 –xzf iptables-1.1.2.tar.bz2
# cd iptables-1.1.2
# make KERNEL_DIR=/usr/src/linux
# make install
```

Linux的高级网络功能（续）

- 数据包过滤（续）
  - **iptables**工具软件（续）

3、使用iptables

iptables和ipchains的选项在文字上基本相同，主要差异有：

输入、输出关使用大写字母：INPUT和OUTPUT

-I 标记是指用于INPUT和FORWARD数据关的接口；

-o 标记用于OUTPUT数据关的接口；

源端口和目的端口应该分别使用--source-port和--destination-port进行设定，而不是在源IP地址的后面使用（-s）和在目的IP地址后面使用（-d）来设定；

IP地址和端口的设置标志必须放在-p选项的后面；

-y 标记现在改为-syn；

DENY的目标现在称为DROP

Linux的高级网络功能（续）

- IP地址欺诈技术

IP地址欺诈技术可以允许Linux系统同时具备两种功能：作为路由器，同时进行网络地址转换（NAT）工作。

在局域网中有一台计算机ford通过调制解调器连接到因特网，我们把这台主机ford设置为一个路由器，让本地局域网的各个主机能够通过它把数据包发送到因特网上，而这些数据包看起来就象是从ford主机上发送出去的一样；而当有数据包响应返回的时候，ford主机能够识别出它实际上是发送给局域网中哪台主机的，并把它正确的转发给那台主机。这时ford主机所采用的这种技术我们称为IP地址欺诈技术。

IP地址欺诈技术并不是仅仅只适用于拨号调制解调器，它可以是任何形式的网络接口。

Linux的高级网络功能（续）

- IP地址欺诈技术（续）
  - IP欺诈的三语句解决方案

1、加入/etc/rc.d/rc.local脚本程序中能实现ipchains的IP地址欺诈服务的三条语句是：

```
ipchains –P forward DENY
ipchains –A –i ppp0 –j MASQ
echo    1 > /proc/sys/net/ipv4/ip_forward
```

2、加入/etc/rc.d/rc.local脚本程序中能实现iptables的IP地址欺诈服务的三条语句是：

```
modprobe iptable_net
iptables –t nat –A POSTROUTING –o ppp0 –j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip-forward
```

Linux的高级网络功能（续）

- IP地址欺诈技术（续）
  - 地址欺诈代理

并不是所有的协议都能很好的实现地址欺诈功能，为了在许多行为古怪的协议里也能够使用上IP地址欺诈技术，必须在适当的地方安排一个特殊的代理模块。Linux操作系统带有几个比较常见的代理模块。

ip_masq_ftp	FTP服务
ip_masq_irc	IRC中继聊天地址欺诈
ip_masq_quake	允许通过使用地址欺诈功能的网络玩quake游戏
ip_masq_raudio	正确转发Real Audio实时广播数据包
ip_masq_vdolive	正确转发VDO Live实时影象数据包

这些模块的使用方法很简单,只需要在系统开机引导的时候使用**inmod**命令调入相应的模块就可以了。

第五部分 Linux操作系统的高级网络功能

- 系统管理员使用的TCP/IP
- 网络配置
- Linux的高级网络功能
- **/proc文件系统**

/proc文件系统

- **/proc文件系统**

  /proc文件系统是Linux操作系统为系统管理员提供的，一种研究操作系统以及在必要时设置操作系统参数的机制。

  - /proc文件系统里有些什么

    /proc文件系统保存的是内核信息的一个总结摘要，在它各个子目录里的各个文件分别对应着操作系统内核的某个功能或者一组内核变量。

    通过/proc文件系统我们可以和内核进行双向交流：内核可以为我们生成报告，我们也可以容易的向内核传递信息。比如在/proc/sys/net/ipv4目录里的文件代表着TCP/IP堆栈中的参数，我们可以使用echo命令动态的对他们进行调整。注意：调整内核参数要十分小心！

/proc文件系统（续）

- /proc文件系统（续）
  - 部分有用的/proc数据项

    更多的信息清看内核源代码的Documentation子目录中的proc.txt文件

/proc/cpuinfo	cpu的信息
/proc/interrupts	IRQ的使用情况
/proc/mdstat	RAID配置情况
/proc/meminfo	内存使用情况
/proc/modules	产生与lsmod命令的输出结果相同的信息
/proc/pci	系统中全部已知PCI设备的旧式报告
/proc/rtc	实时时钟的状态
/proc/sound	声卡和系统对它支持资源的信息
/proc/swaps	swap分区的状态
/proc/version	目前使用内核的信息
/proc/ide/*	全体IDE硬盘的信息

/proc文件系统（续）

- /proc文件系统（续）
  - 部分有用的/proc数据项（续）

/proc/scsi/*	全体SCSI硬盘的信息
/proc/net/arp	ARP表
/proc/net/dev	每个网络设备的信息
/proc/net/snmp	关于每一种协议的SNMP统计信息
/proc/net/sockstat	对网络套接字功能的统计信息
/proc/sys/dev/cdrom/info	已安装的CD-ROM光驱信息
/proc/sys/fs/*	内核用于文件系统功能的设置值
/proc/sys/net/core/	未处理网络数据包队列最大长度 netdev_max_backlog
/proc/sys/net/ipv4/	缺省值0,内核响应ICMP的echo_reply消息，icmp_echo_ignore_all设置为1,告诉内核停止对那些消息的回应

/proc文件系统（续）

- /proc文件系统（续）
  - 通过/proc实现的常见报告和设置

1、对SYN FLOOD攻击现象的防护

SYN FLOOD攻击：一个源主机向一个目标主机发送了数量巨大的SYN数据包，却并不准备对SYNACK做出响应。这将引起目的主机上SYN数据表的溢出，使操作系统变的不稳定。

在引导脚本程序**/etc/rc.d/rc.local**文件的末尾加上如下所示的一行语句：

echo “1” > /proc/sys/net/ipv4/tcp_syncookies

打开对**syncookie**的支持特性。syncookie会检测SYN数据包到达的速度，如果这个速度超过一个阀值，它就会从SYN数据表里开始剔除在一个合理时间间隔内没有转移到“已建立”连接数据表取得那些数据项；另外，如果数据表受到一个可能会引起自己溢出的SYN请求，它会忽略那个请求。

/proc文件系统（续）

- /proc文件系统（续）
  - 通过/proc实现的常见报告和设置（续）

2、大容量服务器的问题

/proc文件系统中的下列数据项允许对大负载情况进行处理：

/proc/sys/fs/file-max

定义了Linux操作系统支持同时打开文件的最大个数，缺省为**4096**，在有许多网络连接的繁忙站点上，8192或16384可能会有所帮助；

/proc/sys/net/ipv4/ip_local_port_range

定义了系统能够使用的端口个数的上下限值,缺省是**1024**和**4096**,在超载的系统上,最好改为31000和61000;

/proc/sys/net/ipv4/tcp_max_syn_bachlog

定义了等待连接队列的长度,缺省是**256**个,周期性拒绝连接时可考虑加大。

/proc文件系统（续）

- /proc文件系统（续）
  - 通过/proc实现的常见报告和设置（续）

3、调试硬件冲突

/proc/pci	给出系统上所有PCI设备的详细情况
/proc/ioports	描述了设备和I/O端口的对应关系，以及有什 么冲突
/proc/interrupts	用于显示中断号与硬件设备之间的关系

第一部分　安装Linux操作系统作为服务器软件
第二部分　单主机系统的管理
第三部分　Internet网络服务
第四部分　内部网(Intranet)网络服务
第五部分　Linux操作系统的高级网络功能

Question & Answer

lenovo 联想

谢 谢 ！

lenovo 联想