www.linuxidc.com

Linux 系统基础课程教案

内蒙古工业大学信息工程学院 李维忠 2004.12

第一章 绪论

一、Unix 家族中的一员

①UNIX，1971 年由 AT&T 发布，73 年重写，

70 年代末，AT&T 成立了 Unix 系统实验室（Unix System Lab，USL），宣布对 Unix 产品拥有所有权，1983 年发布了 SYSTEM V。在 UNIX7 之前，UNIX 是免费或收取少量费用开放源码的。UNIX 的研发诞生了 C 语言。

（AT&T，美国电报电话公司。1974 受到美国反托拉斯司起诉，被控利用垄断美国本土电话服务的优势，排挤长途电话竞争者。案件在 1982 年得到和解。美国电报电话公司同意将本土电话业务拆分为 8 个小公司。根据美国反垄断法规定 AT&T 不能进入计算机市场领域，所以 AT&T 无法将 UNIX 作为商品出售。只能自己使用和发放到大学的科研机构中供研究使用。）

②BSD（Berkeley Software Distribution）

伯克利的加利佛尼亚大学计算机科学研究组以 UNIX 7.0 为基础开发了 UNIX 的 BSD 版本。从 1977 年起，它免费发布了数千份 1BSD 和 2BSD（PDP—11 使用），以及 4BSD（VAX 使用）。到 90 年代开发到 4.4 版，以及 FreeBSD。 BSD Unix

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。



Linux公社（LinuxIDC.com）于2006年9月25日注册并开通网站，Linux现在已经成为一种广受关注和支持的一种操作系统，IDC是互联网数据中心，LinuxIDC就是关于Linux的数据中心。

LinuxIDC.com提供包括Ubuntu，Fedora，SUSE技术，以及最新IT资讯等Linux专业类网站。

并被收录到Google 网页目录-计算机 > 软件 > 操作系统 > Linux 目录下。

Linux公社（LinuxIDC.com）设置了有一定影响力的Linux专题栏目。

包括:

Ubuntu专题

Fedora专题

RedHat专题

SUSE专题

红旗Linux专题

Android专题

本站（LinuxIDC）所刊载文章不代表同意其说法或描述，仅为提供更多信息，也不构成任何建议。
本站带宽由[6688.CC]友情提供
Copyright © 2006-2010 Linux公社 All rights reserved

www.linuxidc.com

中最先实现了 TCP/IP，是最具影响的 UNIX。

但 90 年代 BSD 遭遇到 AT&T 的指控，后来 BSD 与 93 年已经卖给 Novell 公司的 Unix 系统实验室(USL)达成和解，但 BSD Unix 系统中必须去除原来来自 AT&T 的不到 10%源码。因此直到 1995 年 1 月他们才发布了完全 4.4BSD Lite 的系统的 FreeBSD 2.0，为此耽误了时机，成就了 Linux 等竞争对手（伯克利计算机系统研究组（CSRG）在发布了 4.4BSD-lite2 之后就解散了，小组的科研人员有些进入了 Unix 商业公司，有些继续进行其他计算机领域的研究。FreeBSD 是 CSRG 在 1992 年终止开发之前发行的最后一个 BSD 版本——4.4BSD 的一个著名分支。BSD 的著名后继有 OpenBSD、NetBSD、FreeBSD 和 BSDI 等）。

③MINIX

1987 年荷兰阿姆斯特丹Vrije（自由？）大学教授Andrew Tanenbaum以UNIX7 为蓝本开发出一个运行于IntelX86 平台上的简化类UNIX系统MINIX（mini-UNIX的意思）来用于教学。MINIX开放源码，因完全独立开发故不受AT&T许可协议的约束。最新的版本是MINIX 2.0.4 [Nov-9-2003]，可以去官方网站免费下载。

④Linux

1991 年，芬兰学生 Linus Torvalds 开始使用 MINIX 时，对 MINIX 提供的功能不满意。于是他自己写了一个类 UNIX 操作系统（尽管还是用 MINIX 编译的）并放到网上让人们

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

自由下载，取名叫 Linux。Linux 刚出现的时候是被 Tanenbaum 所鄙视的，因为它是集成化内核，不是 MINIX 采用的代表先进思想的微内核；但由于遵循 GPL 协议，Linux 仍得到了蓬勃发展。1994 年，Linux 的第一个商业发行版 Slackware 问世。1996 年，NIST 的计算机系统实验室确认 Linux 1.2.13 版符合 POSIX 标准。Linux 的核心的最新稳定版是 Linux 2.6.1 [Jan-9-2004]，可去官方网站下载。而它的发行版比比皆是，知名的有 Red Hat，Mandrake，Lycoris 等等，国内也有红旗等，相应官方网站可以找到这些发行版的下载。

⑤Xenix 与 SCO Unix

Santa Cruz Operation（SCO）公司的 UNIX 主要有两套。一方面，1980 年的时候，Microsoft 开始研制 UNIX 的 PC 版本 XENIX，后来卖给了 SCO。在这个基础上，SCO 又不断引入 AT&T 的技术，后来发展成为了 SCO Open Server 系列。目前我国银行，邮政等系统很多都用的这个操作系统，最新的版本是 SCO Open Server 5.0.7 [Feb-21-2003]。另一方面，1993 年，AT&T 把他们关于 UNIX 的整个队伍、技术包括版权卖给了 Nowell 公司，有了 UnixWare。1995 年，Nowell 又把这一套东东卖给了 SCO，这样 SCO 就成了 UNIX 的嫡传正宗。SCO 整合了自己的技术，发布了 UnixWare 7。2001 年，Linux 厂商 Caldera 并购了 SCO，从 SCO 的手中获得了 Unix 的知识产权，2002 年 8 月 26 日，

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

Caldera 宣布重新更名为 SCO 集团，以便反应出其营业收入主要来自于 SCO 业务而非 Linux 产品的状况。UnixWare 最新版是 UnixWare 7.1.3 [Nov-21-2002]。2003 年，SCO 指控 Linux 代码侵权，目前 IT 界仍在沸沸扬扬。

⑥Solaris

Sun Microsystems公司早期的操作系统版本Sun OS是基于BSD的。在 1993 年，他们与AT&T合作，转向了UNIX System V，并发布了称作Solaris.System V release 4 的系统，这是一个UNIX System V和BSD的整合体。Solaris系统主要是用在Sun公司自己的SPARC机器上，但他们也发布了相应的x86版本，并且免费。最新的测试版本Software Express for Solaris(12/03) [Dec-3-2003],在官方网站有下载。

⑦可移植操作系统界面标准 POSIX

为了克服 UNIX 版本差异对可移植性的损害，1990 年 ISO 制定了 ISO/IEC 9945-1-1990 标准，这是整个可移植操作系统界面（POSIX）的第一部分；系统应用程序界面即 POSIX.1。这就是常说的 UNIX 操作系统的界面标准。

Novell 后来把自己的 Unix 改名为 Unixware， 而将 Unix 商标赠送给一个由众多 Unix 厂家组成的联盟 X/Open，这样这个联盟内的所有成员均可使用 Unix 商标。从此之后，Unix 不再是专有产品了。95 年 Novell 又将 Unixware 卖给 SCO 公司。此时，严格意义上的 Unix System V 和 BSD Unix 都不复存在了，存在的只是他们的各种后续版本。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

二、Linux 与 GNU

GNU 是 FSF（Free Soft Foundation 自由软件基金会）的一项开放源码工程计划，按照其官方网站的说法，GNU 是“GNU's Not UNIX”的递归缩写(GNU is a recursive acronym for “GNU's Not UNIX”; it is pronounced “guh-noo.”)。

GUN 由 Richard Stallman 于 1984 年发起，目标是开发一个完全免费的可移植的类 UNIX 系统及其应用程序。GNU 开发的类 UNIX 系统名为 Hurd，但未完成，被 Linux 取代。GNU 最早的重大贡献是 1987 年 6 月完成的 11 万行源代码开放的"编译器"（GNU gcc）。

GUN 制定了一项被称之为 Copyleft 版权模式的 GPL（General Public License）条款。GPL 允许任何人免费获取、共享、修改、散发自由软件的源代码，前提是修改散发时不得添加非 GPL 限制。在遵循 GPL 规则的情况下不反对销售自由软件。

Linux 93 年加入 GNU，奉行 GPL 版权。

三、Linux 简史：

1990 年，Linus Torvalds 首次接触 MINIX

1991 年，Linus Torvalds 开始在 MINIX 上编写各种驱动程序等操作系统内核组件

1991 年，Linus Torvalds 在 Usenet 新闻组发布公告，公开了 Linux 内核

1993 年，Linux 1.0 版发行，Linux 转向 GPL 版权协议

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

1994 年，Linux 的第一个商业发行版 Slackware 问世

1996 年，美国国家标准技术局的计算机系统实验室确认 Linux 版本 1.2.13（由 Open Linux 公司打包）符合 POSIX 标准

1999 年，Linux 的简体中文发行版相继问世

2001 年，Linux2.4 版内核发布

2003 年，Linux2.6 版内核发布

Linux 内核由 Linu 负责维护，各种发行版均是内核同一定数量的应用程序的集成。这些应用程序很多基于 GNU。

四、Linux 的版本

Linux 版本分内核(Kernel)版本和发行(Distribution)版本。

内核的版本号由三部分数字组成：

major.minor.patchlevel　即：主版本号.次版本号.修订序号

如：　Kernel 2.4.20

根据约定，次版本号为奇数表示最新版，未必稳定，相当于测试版。次版本号为偶数表示稳定版。

发行版本号一般由各发行商自己规定。

五、Linux 的 GUI

Linux 的图形界面主要有 gnome 和 KDE

gnome (GNU's Network Object Environment) GNU 网络对象模型环境。（Gnome [nəum]土地神、格言，箴言，侏儒）

KDE 是 K Desktop Environment 的缩写形式。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

无论 Gnome 还是 KDE，其基础都是 X-Window

与 Windows 不同，所有 Unix 系统都是都是采用了内核与图形界面分离的做法。Unix 界通用的图形环境是 X-Window，是一种网络化的图形环境。

X-Window 服务器负责响应客户程序的图形输出请求将其显示在显示器上，并负责将输入设备的事件传送给客户程序。

但掌握 Linux 还无法避开字符界面下的命令行操作知识。

（X-Window，84 年诞生于麻省理工学院(MIT)，由来源于斯坦福大学的名为 W 的实验性图形界面系统发展而成。87 年发布了较成熟的第 11 版 X11，并交给非赢利组织 X 协会(The X Consortium)来管理。

六、关于本次 Linux 课程

开设本次课程的初衷

①目前在服务器和嵌入系统中 Linux 应用较多

②国产 CPU 目前只支持 Linux

③作为一种极具代表性的开源操作系统，专业人士理应熟悉

④ＡＯＳ项目的基础开发与运行平台。

七、推荐本次课程的主要参考教材

《新概念电脑教程提高版 操作系统 Linux》

本书编委会 电子工业出版社 2000.12 ￥35.00

(建议：其它操作系统的状况可参见X86 平台操作系统概览)

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第二章　Linux 的安装

一、Linux 分区

Linux 应该使用多少个分区的问题一直存在争论。曾经流行的一种观点是使用 9 个分区，而 Redhat Linux8.0 开始将分区数目减少为三个：

\　　根分区

\boot　　引导分区，100M 即可。存放启动 Linux 时使用的一些核心文件。

\swap　　交换分区，用来支持虚拟内存，容量应为内存的两倍。

二、Linux 对于磁盘设备的命名规则

Linux 通过字母与数字的组合来标识磁盘及分区，例 hda1。

•前两个字母表示磁盘类型：

hd　IDE 硬盘

sd　SCSI 硬盘

fd　软盘

•第三个字母表示是哪个磁盘。如：

hda　第一个 IDE 硬盘

sdb　第二个 SCSI 硬盘

（注：PC 机有两个 IDE 接口，每个 IDE 接口可接两个硬盘或光驱；
SCSI 需要加装专门的接口卡，每个 SCSI 接口卡可接 7 个设备）

•接下来的数字表示分区。

1-4　主分区或扩展分区的总标识（注意最多只有一个扩展

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

分区）。5 以上表示逻辑分区。

注：Linux 采用一个目录树方式管理文件，上述的磁盘标识一般仅用在设备配置及挂载场合。

x

三、Linux 的多重系统引导装载器

①引导器的作用：

多种操作系统共存；多版本内核共存。

②目前 Linux 上流行两种引导装载器：

LILO (the LInux LOader)。比较流行。

Grub (GRand Unified Bootloader)。**Redhat7.2** 以上默认。

(lilo 气垫（商标名称）;grub 幼虫, <俚>食物，挖，掘)

③引导装载器的安装位置：有两种：

- 安装在 **MBR**（**Main Boot Record**）
- 安装在 **Linux** 分区的引导扇区中

通常的情况是选 **MBR**（第一硬盘的 **0** 号磁头 **0** 柱面 **0** 扇区）。若选第二种（活动分区的第一个扇区）必须保证该 **Linux** 分区是可引导的主分区，否则以后就只能以软盘启动。安装完毕后应马上制作一张这样的系统启动软盘。以防不测。

编辑 /boot/grub/grub.conf可改变grub的默认值。

编辑 /etc/lilo.conf 可改变 lilo 的默认值

(grub 与 **LILO** 相比有以下特点：支持大硬盘、支持开机画面、可在选择开机前动态改变开机时的参数、菜单式系统选择、分区位置改变后不必重新设置、**md5** 算法加密口令。**)**

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

四、Linux 的四种安装类型

①个人桌面：个人计算机或便携电脑的理想选择。请选择该安装类型来为家庭或桌面使用安装图形化桌面环境和系统。

②工作站： 该选项安装带有软件开发和系统管理工具的图化桌面环境。

③服务器：如果你想设置文件共享、打印共享、以及万维网服务，请选择该安装类型。你还可以启用其它服务，或选择是否要安装图形化环境。

④定制：选择该安装类型来获得对安装进程的完全控制。包括软件包的选择和验证首选项。

五、Linux 的安装方法

①光盘安装

②硬盘安装

③通过网络安装（HTTP、FTP、NFS）

根据情况决定方法：

•应首选光盘安装，图形安装界面，操作方便。

•没有 Linux 安装光盘但网络稳定快速（如 LAN）且有站点提供安装服务，可以通过网络安装；

•有 Linux 安装盘的 ISO 映像文件且计算机硬盘容量较大，或有两块硬盘，可以采用通过硬盘的安装方法。

②与③方法基本类似，只是在安装过程中的选项稍有不同而已

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

开始非光盘安装 Linux 有两种启动方法：

①DOS 启动：利用 WinISO 将 CD1 解压到一个临时文件夹，重新启动进入 DOS，进入该文件夹 DOSUTILS 目录，执行其中的 AUTOBOOT.BAT。

②软盘启动：制作安装启动盘需要用到三个文件：rawrite.exe、bootdisk.img、drvnet.img。分别在光盘的 DOSUTILS 和 IMAGES 目录中。若你已经解压了整个 CD1，可进入 DOSUTILS 目录中执行 RAWRITE.EXE，依提示输入 ..\images\bootdisk.img 和 a，制作启动盘；再执行 RAWRITE.EXE，依提示输入 ..\images\drvnet.img 和 a，制作网卡驱动盘；否则可以只从上面给出的 FTP 站点下载这三个文件到一个硬盘目录来制作。用启动盘启机即可开始安装。

在可能的情况下，应尽量采用方法②。

安装过程可选择：定制、自动分区。

安装 Linux 前应保证硬盘有足够的未分配空间（如果完整安装 Linux 的话，至少需要 5G 以上空间）。调整分区空间可利用分区工具 PartitionMagic Pro7.0 简体中文注册版。

六、文档的安装方法：

以 root 身份登录 Linux 后下载文档 CD 到/opt 目录，在/mnt 目录下建一个目录 iso，再执行命令：

mount -t iso9660 -o loop /opt/RedhatLinux9.0 文档 CD.ISO /mnt/iso

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

就可以在/mnt/iso 中访问文档 CD 中的文件。双击其中的 autorun，即可开始文档的安装。

七、安装演示

系统版本：Redhat Linux 9.0

安装方法：CD、硬盘、FTP 或 HTTP

切换 gnome 与 KDE 界面的两种常用方法：

①可在图形界面登录时通过点击“会话”项来更改选择；

②在字符界面下执行命令：

switchdesk GNONE 或 switchdesk KDE

(在图形界面执行 switchdesk 会弹出选择窗口。)

八、显示开机设备信息

Linux 启动时会显示大量的设备信息，这些信息有助于了解设备状况，尽管初学者一时可能有些不适应，甚至感到手足无措。熟悉之后就会感觉这些信息是很有用的。系统启动后，可执行 dmsg 命令再现开机时的显示信息。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第三章 Linux 初步

一、 基本操作

①登录：

Linux 是多用户系统，必须用正确的用户名和口令登录后才能进入 Linux Shell 提示符状态。

默认的文本界面 Shell 提示符有两种：

- root 用户登录后的提示符： #
- 普通用户登录后的提示符： $

登录后的当前目录是登录用户的主目录。在 X-Window 下桌面上将出现该目录的文件夹图标。在文本终端下，假设张三以 z3 的用户名登录， Shell 将显示：

[z3@localhost z3]$

第二个 z3 表示的是当前目录名。

通常，用户的主目录往往按默认取值取与用户名一致。

（注意：⑴与 Windows 不同，Linux 区分字母大小写；

⑵Linux 系统在输入口令期间，屏幕光标不作反映。）

②注销

用户应该以注销的方式来结束使用。注销命令：

logout 或 exit 或 ctrl+d

③root 用户在终端界面下的关机或重启命令（常用于远程维护）：

init 0 关机

init 6 重启

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

但是，更提倡使用能够向其他用户发送关机信息的命令：

shutdown –h now　立即关闭系统

shutdown –c　在规定时间内取消关闭系统操作

shutdown +10　系统在 10 分钟后关闭

shutdown –r +10　系统在 10 分钟后重启

shutdown –r now 或 **reboot**　系统立即重启

(去掉/etc/inittab 文件中的“ca::ctrlaltdel:/sbin/shutdown –t3 –r now”可屏蔽 ctrl+alt+del，以避免误操作。)

④多个用户登录

Linux 提供了六个虚拟终端(TTY1-TTY6)和一个 X-window 图形终端供不同的或相同的用户名登录，用 ctrl+alt+Fn 切换：

ctrl+alt+F1 至 ctrl+alt+F6　：TTY1 到 TTY6

ctrl+alt+F7　：X-window 图形终端

(TTY：TeleTYpe，电传打字机。)

⑤用于登录用户之间交谈的命令

write [用户名称][终端机编号]

通过 **write** 指令可传递信息给另一位登入系统的用户，当输入完毕后，键入 **EOF**（**ctrl+D** 组合键）表示信息结束，**write** 指令就会将信息传给对方。如果接收信息的用户不只登入本地主机一次，你可以指定接收信息的终端机编号。（在 **Redhat Linux9.0** 上的实验表明，需要回车后再 **ctrl+D**）。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

wall [公告信息]

wall 是 write all 的意思。通过 wall 指令可将信息发送给每位同意接收公众信息的终端机用户，若不给予其信息内容，则 wall 指令会从标准输入设备读取数据，然后再把所得到的数据传送给所有终端机用户。

（还有个 talk 命令，与 write 作用相同，但必须启动 talkd 守护程序）

mesg [ny]

mesg n　　拒绝接受 write 信息。

mesg y　　允许接受 write 信息并在屏幕上显示。

用 write 命令向一个已经 mesg n 的用户 l4 发送信息，会得到提示：write: l4 messages disabled。

⑥who 命令

显示当前登录到系统的所有用户名、所用终端名和登录时刻。

例如：　**$who**

```
z3      tty1      Dec 11 14:47
root    :0        Dec 11 14:45
l4      pts/0     Dec 11 15:08 (202.207.30.131)
root    pts/1     Dec 11 15:08 (:0.0)
```

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

表示 zs 于 Dec 11 14:47 登录终端 1；root 于 Dec 11 14:45 登录 X 终端 0 端口；l4 于 Dec 11 15:08 由网络(202.207.30.131)登录到 pts/0; root 于 Dec 11 15:08 由 X-Window 桌面创建的终端窗口(:0.0)登录到 pts/1。

（tty 为 teletype 的缩写）

（Linux 的用户管理内容在第 5 章，建议同学此时先利用图形界面操作来创建 z3、l4 两个用户，以便练习上述命令。）

二、 几个常用的基本命令

① **mkdir** 创建目录

例：**mkdir m5**　　在当前目录下创建 m5 子目录

mkdir /home/t1　　在/home 目录下创建名为 t1 的子目录

注意：Unix 的文件路径分隔符是“/”而不是“\”。

② **cd** 进入指定目录

例：**cd /home/t1**　　进入/home/t1

cd myc　　进入当前目录下的 myc 目录

cd ..　　进入父目录（注意..前必须有空格）

cd /　　进入根目录（注意/前必须有空格）

cd 或 **cd ~**　　进入操作者的主目录

cd -　　返回到前一个操作目录

（多次执行可在两个目录间来回切换）

③ **pwd** 显示当前工作目录（**print working directory**）

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

④ **cp** 拷贝文件

例：**cp /etc/passwd a.txt**　将/etc 下的 passwd 文件拷贝到当前目录并改名为 a.txt

cp /etc/passwd .　将/etc 下的 passwd 文件拷贝到当前目录

cp –r /home/z3 /　将/home/z3 目录拷贝到根目录下

⑤ **rm** 删除文件或目录

例：**rm a.txt**　删除当前目录中的 a.txt 文件

rm -r m5　删除当前目录下的 M5 目录及其中文件

注：使用-f 参数可避免确认。参数可以连写，如：

rm –rf m5

⑥ **touch** 创建空文件，或改变指定文件的时间标记。

⑦ **ls** 文件列表（**list directroy content**）

ls 命令有几个常用的可选参数：

-a　列出全部文件名，包括以“.”字符开始的项目

-l　以长格式列出文件详细信息

-i　列出文件的索引号（inode）

-d　只显示目录名称，不显示其中内容

-R　递归(Recursive)列出所有子目录层

例：**ls –l**

总用量 11　(块数，不同的版本这行有所差异）

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

drwxr-xr-x 5 root root 1024 sep 13 03:27 Desktop

-rwxr-xr-x 2 root root 737 oct 7 18:42 a.txt

每列分别表示：属性、连接个数、所有者、从属组、文件长度、文件的最后更改时间、文件名。

其中文件属性的第一个字符标识文件的类型：

d 目录	**s** 套接字
- 普通文件	**p** 命名管道
b 块设备	**l** 符号连接
c 字符设备	

（**file** 是文件类型识别命令，可以文字形式显示文件属性）

类型的其它 **9** 位字符分别表示所有者、从属组、其他用户对此文件的读、写、执行权限（-表示无相应权限）。

通配符和模板可用来在 **ls** 命令中限定列表条件：

通配符	含义	举例
*	与任何字符匹配	**ls hd*** 显示以 **hd** 开头的文件名
?	与一个字符匹配	**ls t?** 显示 **t** 开头的所有 2 字符文件名
[…]	与括号中任一个字符匹配	**ls a[136]** 只与 **a1**、**a3**、**a6** 匹配
[.-.]	与括号中的字符范围匹配	**ls a[b-e]** 只与 **ab**、**ac**、**ad**、**ae** 匹配
[^…]	对括号中的字符集或范围取反匹配	**ls hda[^b-d]** 只与非 **hdab**、**hdac**、**hdad** 匹配
{…, …}	只与括号中的字符串匹配	**ls a{a1,b2,c3}** 只与非 **aa1,ab2,ac3** 匹配

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

ls 命令显示颜色的意义：

绿 色——可执行文件

蓝 色——目录

红 色——压缩文件

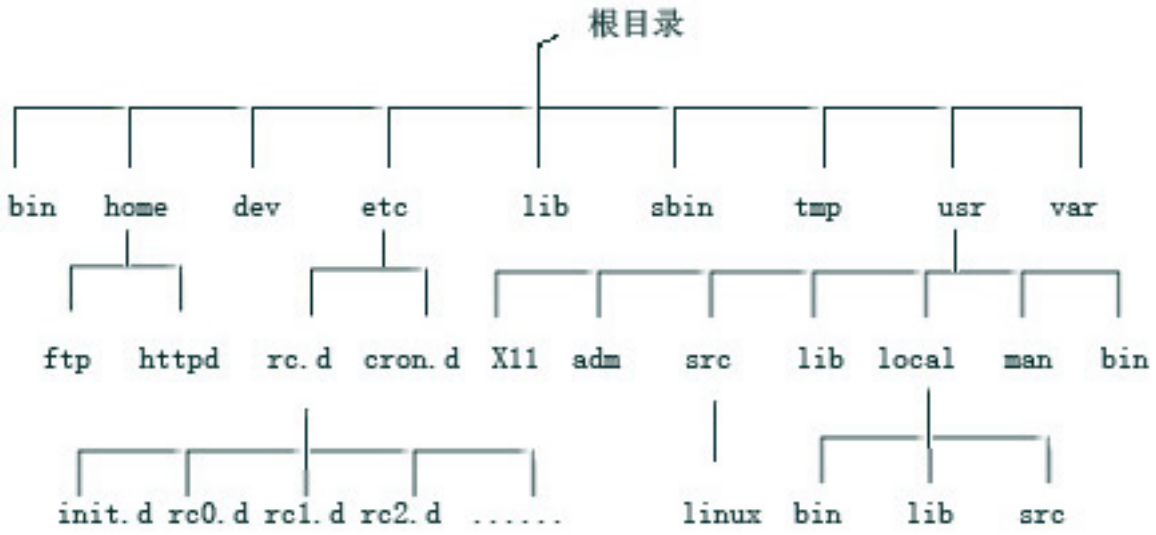
浅蓝色——链接文件

灰 色——一般文件（未定义的文件）

注：不同版本间有异，修改**/etc/DIR_COLORS** 可定义颜色。

练习与提示：当前目录下有一个 **kkk** 子目录，**ls –l kkk** 或 **ls –l k*** 只能查看到 **kkk** 下的文件名及属性。若要查看 **kkk** 自身的目录属性而不是其下的文件，需要使用**-d** 参数：**ls –ld kkk**

三、 Linux 系统目录结构



/bin — 用来贮存用户命令。目录 **/usr/bin** 也被用来贮存用户命令。

/sbin — 许多系统命令（例如 **shutdown**）的贮存位置。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

目录 /usr/sbin 中也包括了许多系统命令。

/root — 根用户（超级用户）的主目录。

/mnt — 该目录中通常包括系统引导后被挂载的文件系统的挂载点。譬如，默认的光盘挂载点是 /mnt/cdrom/.

/boot — 包括内核和其它系统启动期间使用的文件。

/lost+found — 被 fsck 用来放置零散文件（没有名称的文件）。

/lib — 包含许多被 /bin/ 和 /sbin/ 中的程序使用的库文件。目录 /usr/lib/ 中含有更多用于用户程序的库文件。

/dev — 贮存设备文件（注意，设备文件不是驱动程序。过去，在添加新磁盘或设备后，往往需要手动增加设备文件。现在通常我们不需要手动增加设备文件，运行一下 service kudzu start ，系统就会自动配置相应的设备）。

/etc — 包含许多系统配置文件和目录。

/var — 用于贮存 variable（或不断改变的）文件，例如日志文件和打印机假脱机文件。

/usr — 包括与系统用户直接有关的文件和目录，例如应用程序及支持它们的库文件。

/proc — 一个虚拟的文件系统（不是实际贮存在磁盘上的），它包括被某些程序使用的系统信息。

/etc/rc.d — 启动的配置文件和脚本。

/tmp/ — 用户和程序的临时目录。所有系统用户均可读写。

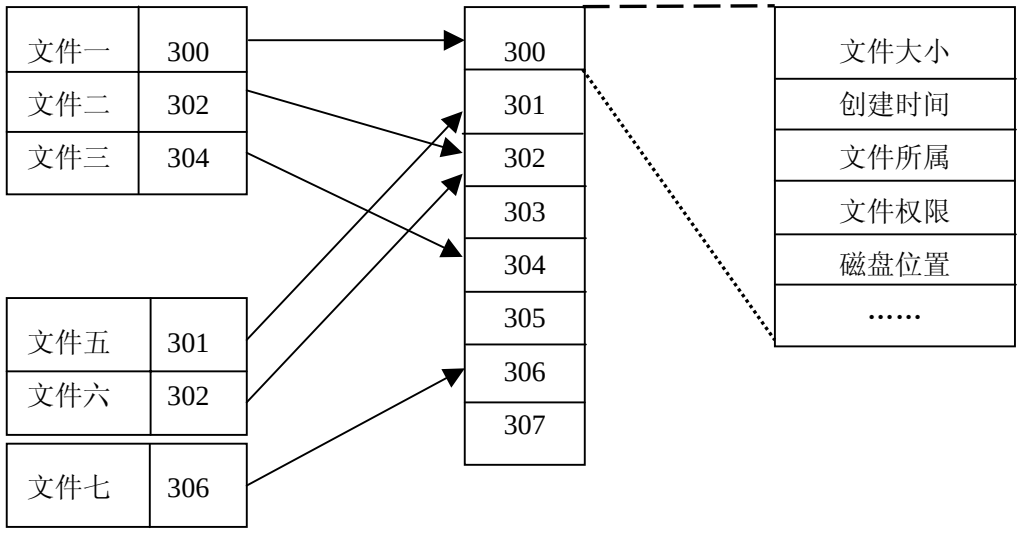
/home — 用户主目录的默认位置。

/opt — 可选的（optional），可用来存放准备安装的文件。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

四、 文件的连接



Linux 采用索引节点 inode 记录文件信息，每个文件或目录都唯一地对应 inode 数组中的一个元素。但一个元素可以与多个文件或目录名对应。文件名与索引节点号的对应称之为连接。

ln 命令用来对一个已经存在的文件建立连接。

例：cd /root

cp /etc/passwd .

ln /root/passwd /root/passwd.2

执行命令 ls –l 可在第二列看到连接数变为 2。

执行命令 ls –i 可以看到 passwd 和 passwd.2 具有相同的 inode。

这种通过相同 inode 共享同一个物理文件的方法称之为硬连接。

优点：可在不同目录通过不同路径访问同一个文件，节省磁盘空间又便于数据一致性；有一定防止误删除功能，只有同一个 inode 属于唯一一个连接时，数据才被物理删除。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

缺点：不能连接目录，不能跨越不同文件系统（如分区、驱动器）。

加参数-s 执行 ln 命令可建立软连接，也称符号连接，如：

ln –s passwd passwd.2

执行 **ls –l** ，可看出 **passwd.2** 是 **passwd** 的符号连接。

passwd.2 和 **passwd** 是两个不同的文件，前者的文件数据是指向后者的路径。即便在源文件不存在的情况下也可建立符号连接。

五、 mv 命令

Linux 中没必要配备专门的文件更名命令，使用移动命令 **mv** 即可完成文件的更名操作。**mv** 实际上是在创建一个新的硬连接后删除旧的硬连接，以此实现文件的搬移或更名。如

mv passwd.2 passwd.b

执行 **ls –i** 命令可发现文件名变了但 **inode** 没变。

六、 获得帮助

①使用 **man topic** 命令查阅手册页，如欲了解 **ls** 命令，可：

man ls

man 命令只显示查找到的第一项。若同一主题的内容可在多个部分找到（如 **kill** 既是 **shell** 命令又是系统调用），则需：

man –t 2 kill

⑵使用参数 **- -help** 可显示命令的帮助信息，如：

ls - -help

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第四章 磁盘设备

一、关于挂装的基本常识

- 与 DOS/Windows 采用驱动器标识符（A:、B:、C:）使用磁盘设备的方法不同。Linux 采用单根目录树管理全部文件系统。磁盘设备必须挂载到系统目录树上才能使用。

  (Linux 启动过程已完成对/、/ boot 和/swap 三个分区的挂装)
- 所谓挂装，就是将该设备的文件系统作为一个分枝嫁接到主文件系统的过程。嫁接的位置称为挂装点。
- 挂装点必须是个已经存在的目录。该目录可以不空，但用作挂装点后原内容将不可见，直到卸载为止。
- Linux 为用户的后续挂装专门提供了/mnt 目录，/mnt 下的目录用作挂装点。若无特别原因，应尽量遵循这一约定。
- 只有 root 有权执行挂装与卸载操作。
- 修改/etc/fstab 文件可实现启动时的自动挂装。
- gnome 和 KDE 环境支持光盘和软盘插入后的自动挂装。但在字符界面以及更多场合下，仍需用户手工挂装。
- 挂装命令 mount，常用的语法格式是：

  mount [-t 文件系统类型] [被挂装的设备文件] [挂装点]

常用的文件系统类型：

ext2	: Linux 的 ext2 文件系统
msdos	: MS-DOS 的 FAT
vfat	: Win95/98 的 VFAT
nfs	: 网络文件系统
iso9660	: CD-ROM 光盘的标准文件系统
ntfs	: WindowsNT 文件系统
auto	: 自动检测文件系统的类型

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

•卸载命令 umount，常用的语法格式是：

umount　挂装点或被挂装的设备文件

（如卸载光驱既可以 umount /mnt/cdrom 也可以 umount /dev/cdrom）

二、应用实例：

①使用 CD-ROM：

挂装：mount　/dev/cdrom　/mnt/cdrom

卸载：umount　/mnt/cdrom

(挂装时也可使用-t iso9660 或-t auto 参数)

②使用软盘：

挂装：mount　-t auto /dev/fd0　/mnt/floppy

或 mount　/dev/fd0　/mnt/floppy

或 mount　/mnt/floppy

卸载：umount　/mnt/floppy

③使用 USB 盘：

挂装：mount　-t vfat /dev/sda1　/mnt/usb

卸载：umount　/mnt/usb

说明：⑴Linux 将 USB 盘作为 SCSI 盘处理；

⑵挂装前先要在/mnt 下创建一个 usb 目录；

⑶必要是可先 fdisk –l /dev/sd? 查看 U 盘的设备名。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

④访问硬盘上的 DOS/Windows 分区

先执行： fdisk –l /dev/hd?

了解 DOS/Windows 主分区和逻辑分区所在的位置，如果 Windows 有主分区 C 和逻辑分区 D 两个分区，一般是在 hda1 和 hda5。以访问逻辑分区 D 为例:，

挂装：mount -t auto /dev/hda5 /mnt/d

卸载：umount /mnt/d

当然/mnt/d 目录要事先存在。

如下命令可避免 dos 盘上的汉字文件名出现乱码：

mount -t auto –o iocharset=gb2312 /dev/hda5 /mnt/d

若要使所有用户都对所挂装的/mnt/d 有读写权限，则需：

mount –t vfat –o user,rw,exec,,umask-000 /dev/hda5 /mnt/d

若仍无法写 DOS 分区，则可能需要设置挂装点权限：

chmod a=rwx /mnt/d

⑤使用 iso9660 映像文件：

在/mnt 目录下建一个目录 iso;

假设映像文件名为 a.iso，在/opt 目录下，则：

挂装：mount -t iso9660 -o loop /opt/a.iso /mnt/iso

卸载：umount /mnt/iso

挂装期间，在/mnt/iso 目录中可对映像包括的文件进行操作。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

三、软盘操作

①格式化

先低格：**fdformat /dev/fd0H1440**

再 **mformat a:** 在低格过的软盘上建立 **DOS** 文件系统。

或 **mkfs –t ext2 /dev/fd0 1440** 建立 **ext2** 格式。

② 读写软盘

挂装后才能使用，取出软盘前一定先要卸载以保数据完整。可用 **Linux** 命令对**/mnt/floppy** 进行软盘文件操作。另外 **Linux** 还提供了一套称之为 **mtools** 的访问 **dos** 软盘的工具，用法类似 **dos**。主要有 **mformat**、**mdir**、**mmd**、**mcd**、**mrd**、**mcopy**、**mcomp**、**mdel**、**mdeltree**、**mattrib**、**mren**、**mtype**、**mlabel** 等，键入 **mtools** 回车可列出其命令清单。

③复制软盘

先存源盘映像 **dd if=/dev/fd0 of =temp.img**

再写到目标盘 **dd if=temp.img of =/dev/fd0**

（在一些资料中其举例常加参数 **bs=1440k**，实验不加也可）

复制软盘不必事先挂装。

四、制作 ISO 光盘映像文件

将源文件用归入一个目录，如 **myfiles**；

mkisofs -L -l -v -r -o /tmp/outfile.iso myfiles

/tmp/outfile.iso 即是生成的 **ISO** 映像文件。

(-L 允许文件名的第一个字符为句号；**-l** 可使用 **iso9660 32**

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

字符长度的文件名，可能在 DOS 下无法正常读取； -v 执行时显示详细信息； -r 使用 Rock Ridge Extensions，开放所有文件的读权限；-o 指定映像文件名称。)

五、其它磁盘命令

① **fdisk** 磁盘分区命令。-l 参数用于列出指定的外设分区表。

② **df** 检查磁盘空间及利用状况

③ **du** 显示磁盘中目录或文件的大小

例：**du –b /boot**

显示/boot 目录的大小。-b 参数用以指定单位为字节。

④ **dd** 介质拷贝命令

用指定大小的块拷贝文件并进行指定的格式转换。

bs=<字节数>	将输入 ibs 输出 obs 设定为指定的字节数
cbs=<字节数>	每次只转换指定的字节数
conv=<关键字>	指定文件的转换格式
count=<扇区数>	仅读取指定的扇区数
ibs=<字节数>	指定每次读取的字节数
if=<文件>	输入文件名，省略则从标准设备输入
obs=<字节数>	每次输出的字节数
of=<文件>	输出文件名，省略则输出到屏幕
seek=<扇区数>	输出数据跳过指定的扇区数。(扇长=obs)
skip=<扇区数>	读取跳过指定的扇区数。(扇长=ibs)

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第五章 用户管理

一、增加新的用户帐号

以新增一个 **m5** 账号为例：

useradd z3 添加新用户 **z3**

passwd z3 赋予新用户 **z3** 口令

说明：①新用户只有被赋予口令后才能使用；

②**Redhat Linux** 中，**adduser** 是 **useradd** 的符号连接。

③**useradd** 命令常用的参数：

-u 设置用户的 **UID**

-g 设置用户的 **GID**

-d 设置用户的主目录

-G 使该用户成为其他组的成员

-s 用户的登录 **shell**，默认为 **/bin/bash**

-c **comment** 用户注释

-p 同时设置口令，注意无空格。**(Redhat** 版支持**)**

添加新用户的操作实际上是修改了**/etc/passwd**、**/etc/shadow** 文件，在**/home** 下为创建主目录，并将**/etc/skel** 目录内容拷入。

二、/etc/passwd 文件

/etc/passwd 文件记录用户账号信息，**(**权限：**root.root 644)**

该文件每行由 **7** 个域组成，之间由冒号分隔：

用户名：口令：**UID**：**GID**：注释：用户主目录：**shell**

例如：

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

root:x:0:0:root:/root:/bin/bash

ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin

z3:x:500:500:ZhangSan:/home/z3:/bin/bash

l4:x:501:501:LiSi:/home/l4:/bin/bash

w5:x:502:502:WangWu:/home/w5:/bin/bash

m6:x:503:503:MaLiu:/home/m6:/bin/bash

说明：

UID：用户标识；　　**GID**：组标识；　　**x** :口令被加密

用户名应该唯一，如不唯一则系统只认前面的一个。

UID 不必唯一。相同的 **UID** 具有完全相同的权限。

普通用户的 **UID≥500** 只不过是个约定。但等于 **0** 则具有特别意义，即拥有 **root** 权限。其实获得 **UID=0** 的特权就是网络攻击者的目标。

三、/etc/shadow 文件　　(权限：root.root 400)

/etc/shadow 文件主要用于记录加密加密后的口令，其格式：

username: password: last:may: must: warn: expire: disable: reserved

last：上次更改口令日期，从 **1970-1-1** 起计算的天数；

从 **may** 至 **nust** 的期间，用户才可以更改口令；

warn：在 **must** 前多少天提示用户更改口令；

expire：在 **must** 后多少天使用户账号失效；

disable：只用来记录账号失效的天数；

reserved：保留，尚未使用。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

例：

root:1RE9bMMe8$4G3nQB7tiyYYHKLJMdY.m/:12770::99999::::

ftp:*:12734:0:99999:7:::

z3:1vqHltSd2$/Zv6HBjDXknqkY8MTiW.20:12763::99999::::

l4:$1$2eRrDN.g$fCtWeiIvonMyVW5NklRfI0:12763::99999::::

w5:1sTaNpD10$ho3C12iBGIaVD1kxfOaCj/:12763::99999::::

m6:$1$4zO8RY6k$cOTmCThcLfWL7plnWlBFt0:12772:0:0:0:0::

四、用户组

groupadd 命令用于创建用户组

将一个已经存在的用户加入组可使用 **usermod –G** 命令

例：

groupadd g1	创建用户组 **g1**
usermod -G g1 z3	收 **z3** 为 **g1** 组成员

关于组的操作实际上是在修改**/etc/group** 文件。

字符界面下尚未发现有可用的命令用于对组成员的剔除，

此时，对于组管理的最有效手段是手工修改**/etc/group** 文件。

五、/etc/group 文件　　(权限：root.root 644)

/etc/group 文件记录组信息，每行一组，格式：

组名：口令：**GID**：成员列表（成员之间用逗号隔开）

若需组口令可执行 **gpasswd**。加密的组口令存于**/etc/gshadow**。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

六、用户的私有组与登录组

useradd 在创建用户的同时也创建了一个同名组。

新用户默认属于自己的组。这个组称为用户私有组。

其他用户也可以成为这个私有组的成员并享受该组成员权利（仅 **root** 有权操作）。

/etc/passwd 文件中的 **GID** 表示的是用户的默认组。

usermod –g 命令可以改变用户的默认组，也称登录组：

例：**usermod –g z3 l4**　修改 **l4** 的默认组属于 **z3**

此后用户创建的所有目录与文件的组属性属于新的默认组。

七、删除用户或组

userdel 删除用户。

例 **userdel –r w5** 删除 **w5(-r** 表示同时删除其主目录)

groupdel 删除组　例：**groupdel g2**

（注意：一些资料上说该命令不能删除非空组，但实验证实在 **redhat Linux** 中可以删除非空组，而且该命令没有确认机制，请慎用！）

八、改变用户设置

usermod 命令用于改变用户的账号设置，如 **UID**、**GID**、注释、主目录、**shell**。其参数分别是 **-u**、**-g**、**-c**、**-d**、**-s**。

chage 命令用于修改口令的有效期。

具体用法请参考系统帮助。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

九、禁止用户账号

修改用户的 **shell** 为 **nologin** 或 **false** 可阻止其登录系统。

例：**usermod –s /sbin/nologin** 或 **usermod –s false**

十、用户的身份转换

su 命令可改变用户身份，不仅省却了注销和重新登录过程，而且在使用 **root** 身份无法登录的 **Telnet** 网络登录的情况下更为有用。（ **su** 是替代用户**(Substitute user)**的意思，但常被理解为 **super user**）。

例：**su root** 在依提示输入 **root** 口令后转为 **root** 身份

但这种转换并没有相应地改换系统运行环境，提示符类似：

[root@localhost z3]#

用户已是 **root**，但仍运行于 **z3** 的运行环境下。这会引起一些程序的运行结果不正常，与真实的 **root** 环境是大有区别的（比如就无法执行关机复位等操作）。要想同时获得新用户的环境，应在 **su** 与用户名间加-号并空格，

即：**su - root**

这时系统提示符显示：**[root@localhost root]#**

十一、 其它

① 执行“**id** 用户名”命令可列出用户的 **UID**、**GID** 和所有组

② 与 **W2K** 不同，**Linux** 的组不能吸纳其它组为成员。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第六章 Linux 文件权限

一、用户分类与文件访问权

- **Linux 将用户划分为四类；**

  root ； 或所有 UID=0 的用户，具有至高权限；

  owner ： 文件的所有者，默认即文件的创建者；

  group ； 所属组，除 owner 外同组其他成员有同等权限；

  world ； 除上面三类外的其他普通用户。

  自然，Linux 的权限管理只针对 owner、group、world。

- **访问权限有三种；读、写、执行，分别用 r、w、x 表示**

  ls –l 命令显示的结果中第 1 列的 2-10 分别表示三类用户对此文件（或目录）的三种访问权限，“-”表示无此权限，例：

  -rwxr-x- - - 2 z3 g1 737 oct 7 18:42 a.sh

  表示 a.sh 文件为 z3 所有，属于 g1 组，z3 具有全部权限，g1 组内其他成员具有读和执行权限，其他用户无任何权限。

- **为便于记忆与交流，专业人员习惯以八进制描述权限，比如上例文件的访问权限可表示为 750。**

二、文件权限与目录权限

文件与目录的访问权限都分为读、写和执行，但作用不同：

	读权限	写权限	执行权限
文件	读取文件内容	修改该文件	执行
目录	列出其中的文件名	在其中创建或删除文件	进入目录

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

注意：

⑴在 **Linux** 中一个文件是否是可执行文件与其扩展名无关。其可否执行取决于两个条件：用户是否具有执行权限；该文件是否是可执行文件。 普通的文本文件（非脚本文件）即便被赋予执行权限也无法执行。**File** 命令可查看文件类型。

⑵**Redhat Linux 9.0** 上的实验表明：若要对目录中的文件名列表或在其中创建与删除文件都必须要对该目录拥有执行权限。这点与很多书籍上的说法存在出入。

三、改变文件的所有者和所在组

① chown 命令：改变文件的所有者

例：**chown l4 a.txt** 改 **a.txt** 文件的所有者为 **l4**

chown l4 mydir 改 **mydir** 目录的所有者为 **l4**

chown –R l4 mydir 将 **mydir** 目录全部归 **l4** 所有

说明：⑴ **-R** 参数表示递归操作，即把 **mydir** 及其下的所有目录与文件的所有者都改为 **l4**。

⑵ 只有 **root** 用户有劝执行 **chown** 命令。

② chgrp 命令：改变文件的所属组

例：**chgrp g1 a.txt** 改 **a.txt** 文件的所属组为 **g1**

chgrp –R l4 mydir 递归改变 **mydir** 目录所属组

说明：普通用户也可以执行 **chgrp** 命令，但前提条件是该用户必须同时具备两组的成员身份。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

③同时改变所有者和所属组

chown newuser.newgroup filename_or_directory

即用点连接新所有者和新组，执行 **chown** 命令。

例：**chown l4.g1 mydir**

四、改变文件的访问权限

- **chmod** 命令用于修改文件或目录的访问权限。命令格式：

  chmod 权限 文件或目录名

例：**chmod 755 /home/l4** 修改 **l4** 的主目录访问权限为 **755**

（注：如果需要，当然也可以使用递归操作参数**-R**）

- 还可以使用字符串取代八进制的权限表示：

u、g、o、a 分别表示 **owner、group、world、all**

r、w、x 分别表示 **read、write、exec**

+、- 分别表示增加、去除权限

例：**chmod a+r g+x u+xw /home/l4**

使**/home/l4** 对任何人可读、组用户可执行、拥有者可读可执行

又例：**chmod +x install.sh**

使当前目录下的 **install.sh** 文件对任何人可执行。这种用法常用在改变下载文件的权限属性以便其可执行。

注意：

⑴对硬连接的访问权限修改会同时影响到所有相关的硬连接

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

权限属性。因为它们是同一个文件。

⑵对软连接的访问权限修改只改变了被连接文件的权限属性，连接文件自身的访问权限并没有被修改，仍是 777。

五、用户的默认访问权限和 umask

用户在创建新文件时，系统通常按默认值赋予其权限：

	目录	一般文件
root	**755**	**644**
普通用户	**775**	**664**

影响用户的权限默认值的是个与用户相关的系统值 **umask**。

Root 的 umask=022，普通用户的 umask=002

目录的默认权限等于：**777** 与 **umask** 反码；（**777** 减去 **umask**）

文件的默认权限等于：**666** 与 **umask** 反码。（**666** 减去 **umask**）

不带参数地执行 **umask** 将显示当前用户的 **umask** 值。

umask 新值 可改变当前用户的 **umask** 值，

例：**umask 222** 令当前用户 **umask=222**，任何人都不可写

说明：

⑴**umask** 只与用户有关，与当前在哪个目录无关。

⑵只能改变自己的 **umask**，并且只对自己的本次登录有效。下次登录 **umask** 还是原值。所以修改的意义似乎不大。

⑶注意：“减去 **umask**”的说法只是一些书籍上为不熟悉布尔代数的初学者提供的一种权宜的助记方法，该说法其实很有问题，造成很多现象无法解释。笔者认为准确的

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

说法应该是：“默认的访问权限等于权限基值与上‘umask 的反码’，对于新建目录和新建文件来说，其权限基值分别是 777 和 666；但对于由拷贝操作产生的新目录和新文件来说，其权限基值就是源目录（或源文件）的原有值”。也就是将权限基值上的那些与 umask 中置 1 位相对应的权限位清零便可得到新文件的默认权限值。

六、理解 SUID 和 SGID

先分析一个现象：

记录用户账号信息的/etc/passwd 文件其权限是：root.root 644

记录加密口令的/etc/shadow 文件其权限是：root.root 400

对于普通用户 passwd 只能读而不能改写；shadow 连读都不允许。可为什么我们可以通过 passwd 命令改写自己的口令？

查看一下命令程序 usr/bin/passwd 文件的属性：

/usr/bin/passwd 文件其权限是：root.root -r-s- -x- -x

这个文件是允许普通用户执行的。root 的执行位是 s，就是这个 s 使普通用户执行 passwd 时具有了 root 身份从而获得了对/etc/passwd 和/etc/shadow 的完全访问权(root 什么权没有？)。

实际上表示 Linux 文件权限的是 12bit，前面的 3bit 是：

SUID：若文件可以执行，则按所有者身份运行；

SGID：若文件可以执行，则按组成员身份运行；

sticky：粘接位，结束后仍驻留内存。目前并不常用。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

在 **ls –l** 命令显示中的表示方法：

SUID 置位：所有者的执行位的 **x** 被 **s** 取代；

SGID 置位：所属组的执行位的 **x** 被 **s** 取代；

sticky 置位：其他人的执行位的 **x** 被 **t** 取代。

所以，**-r-s- -x- -x** 用值表示就是 **4511**。

注意，使用 **SUID** 和 **SGID** 位要特别小心。

七、

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第七章 Bash 操作

一、何谓 Bash

Bash（Bourne Again shell）,是目前 Linux 上最流行的 Shell。

Shell，命令语言解释器（command-language interpreter）

Bash 相当于 DOS 系统的 Command.exe。

Linux 上还有一些较流行的 Shell，如 tcsh 、pdksh 等。

echo $BASH_VERSION 可检查当前所用的 Shell 版本。

若返回出错信息则表明当前的 Shell 不是 Bash，可执行：

exec /bin/bash　　令系统使用 Bash

（/etc/shells 文件注册了当前系统所能使用的 shell 清单。）

二、在一行上顺序执行多个命令

可用分号将几个命令串联，如：

mkdir mydir; cd mydir

注意：⑴用分号串联的命令个数没有固定限制；

⑵各命令依顺序执行，并非同时运行；

⑶无论前一个命令是否成功，都将执行下一个命令。

三、引号和转义

•Linux 系统中需要转义的例子：

⑴$用来表示系统环境变量，可我们需要$的本来文字意义；

⑵显示出含*、？的文件名，比如 a?*，不希望按通配符解释。

这是就要使用转义符“\”，来取特殊字符的文字意义，例：

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

mkdir a\?*;　ls –l a\?*　　创建 a?*目录并显示其中内容

（思考：怎样显示 a?*目录本身属性?）

•单引号括住的字符或字符串只具文字意义；

双引号括住的字符或字符串允许特殊字符保持其特殊意义。

例：

echo　$PATH;　　将显示出当前搜索路径

echo ″$PATH″　　同样将显示出当前搜索路径

**echo　‘$PATH’**　　将显示出“$PATH”

echo　\$PATH　　同样将显示出“$PATH”

（何时需使用单引号或双引号？如，包括空格的字符串！）

四、设备转向（或称输入/输出重定向）

Linux 文件系统的标准设备有标准输入、标准输出、标准错误。

设备名称	缩写	序号	用途	默认连接
标准输入	**stdin**	**0**	读数据	键盘
标准输出	**stdout**	**1**	显示程序输出	显示器
标准错误	**stderr**	**2**	异常时显示错误和警告	显示器

设备的转向就是改变其输入/输出的设备指向，例如将键盘输入转向为文件输入、将输出显示由显示器转向为文件等。

①输出转向

输出转向操作符是>和>>，分别用于新建和追加文件。

例：**ls –l /dev/hd*>result.txt**

将原本输出到屏幕的结果转存入新文件 **result.txt**。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

ls –l kkk>>result.txt

将 **kkk** 目录下的文件列表追加到 **result.txt** 中。

可是，若 **kkk** 目录不存在，出错信息因未转向而仍出现在屏幕上。标准错误的重定向操作符是“**2>**”或“**2>>**”（其实标准输出重定向操作符“**>**”是“**1>**”的省略形式）若要将输出转到新文件 **result.txt**，并且当有错误发生时记录到 **error.txt** 文件中，应该：

ls –l kkk>result.txt 2>error.txt

若要将标准输出和标准错误都转向到同一个文件，应该：

ls –l kkk>result.txt 2>&1

2>&1 句法告诉 **Bash** 那些本来输出到文件描述符 **2**（标准错误）的内容传送到文件描述符 **1**（标准输出），由于文件描述符 **1** 已被转向到文件，所以标准错误也被送入到同一文件中。鉴于该操作极其普遍以至于 **Bash** 有一专门简化用法：

ls –l kkk &>result.txt

②输入转向

输入转向的操作符是“**<**”。

输入转向可将原本应由键盘输入的数据改由文件或其它输入设备输入。例如以前学过的广播命令 **wall**，其格式是：

wall [广播信息**]**

若只键入 **wall** 就回车，程序将等待键盘输入，直到回车再按 **EOF**（**ctrl+d**）。但为简化经常性的信息发布，我们可以

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

建立一个包含信息的的文本文件 **mes.txt**，执行：

wall<mes.txt

③说明

⑴其它教科书上一般都用 **cat**、**wc** 等命令来解释标准输入转向，这些命令在无参数执行时确实是从标准输入设备输入（这可使其作为过滤器），但有文件名作参数时就直接从该文件输入，而不必非要加转向符。即 **cat a.txt** 和 **cat<a.txt** 效果一样。所以笔者不认为这类讲解是合适的，因为其不能体现输入转向的实际应用价值。

⑵ **DOS** 中标准输入和输出设备是 **CON**，标准打印输出是 **PRN**，执行 **copy con a.txt** 将接受其后的键盘输入，当按 **F6** 或 **ctrl+z** 时生成 **a.txt** 文件内容。同 **DOS** 用法类似，**Linux** 也可以使用 **cp /dev/stdin a.txt** 命令去创建文本文件，或者使用 **cp a.txt /dev/stdout** 命令去显示文本文件内容。所不同的是 **Linux** 系统的文件结束符是 **ctrl+d** 而不是 **DOS** 中的 **crtl+z**。

五、命令管道

使用管道操作符“|”将一个以上的命令或程序连接起来，前一个的输出作为后一个的输入，如此形成一个管道。

例：**ls –l /dev|more**　　将目录列表分页显示（**more** 命令后述）

（注：管道中的各个命令或程序是同时运行的。）

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

六、历史表文件.bash_history

•每个用户的主目录下都有一个名为.bash_history 文本文件，该文件记载了用户操作的历史记录。

•Bash 为每个用户维护着一个历史表。

•用户登录时该文件内容加载到历史表，注销时再更新该文件

• history 命令可以列出用户的操作记录，即历史表。

历史表的每一行称之为一个事件，行首数字表示事件号。

•用“!”加事件号作命令可以简便地重复该历史事件，如：

!385

• 特殊操作符!! 用于重复最近一个命令。

• 更常用的技巧：在 bash 下可直接用上光标键回溯操作历史。

七、shell 函数

bash 允许用户自定义命令函数，其格式为：

name() { command_list;}　即：函数名() { 命令列表;}

注意：⑴左花括号后要有一个空格，规定如此，未必有道理；

⑵命令列表可以是多个命令，每个命令后别忘有分号；

⑶函数中可使用$*提取函数的参数。

例：dir() { ls –ld $*;}

以后就可以用命令 dir 来列出指定目录中的文件名和目录名，

如：dir /etc　显示/etc 目录中的文件和目录列表

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

八、命令补全功能

tab 键可用来对命令补全，是 Linux 操作中最常用的技巧之一。

例：在安装一章我们曾接触过一条用于文档安装的命令：

mount -t iso9660 -o loop /opt/RedhatLinux9.0 文档 CD.ISO /mnt/iso

这样长的命令且文件名夹杂汉字，输起来十分不易。

当然可先用 mv 命令将文件更名为一个简单的文件名，但利用补全 bash 的功能更为简便：

当输到“mount -t iso9660 -o loop /opt/R”时按 tab 键，系统会自动依据查找结果补全文件名，简化了不少操作。

若系统没有给予完全补全，说明存在近似的文件需要选择，例：假设/opt 目录中还有一个 Redflag.iso，则光标会停在 Red 后，此时需要按 h 或 f 进行下一步选择。

九、作业控制

进程可以简单地理解为处于调度中的程序，

作业可能有多个进程（如管道）。

①ps 查看进程状态命令

执行 ps 命令可以报告系统中进程的运行状况。如：

PID	TTY	TIME	CMD
2565	tty1	00:00:00	bash
2597	tty1	00:00:00	ps

即系统正在运行两个进程，bash 和 ps 本身。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

常用的 ps 选项：

a	显示所有用户的进程
e	命令执行后显示进程环境变量
f	输出全部信息
j	显示进程的父进程 PID
l	以长格式输出
u	显示用户名和进程的开启时间
w	以宽格式输出，将一行显示不下的其余部分显示出来
t	显示与 tty 设备相关的进程
x	显示没有控制终端的进程

ps 输出域的含义：

UID	进程所有者的名字
PID	进程 ID
%CPU	进程的 CPU 利用率，计算原因有可能超过 100%
%MEM	进程使用内存的百分比
SIZE	进程所使用的虚拟内存量，以 K 为单位
RSS	进程使用的驻留内存量或实际内存量
TTY	与进程相关的终端。有时缩写，如 p6=/dev/tty6。
STAT	进程的状态：R：正在运行或准备运行；S：休眠状态；I：空闲状态；Z：僵尸进程；D：磁盘等待；P：分野等待；W：交换到磁盘；T：终止；N：由 nice 降低优先级；<：由 root 提高执行优先级。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

START	进程启动的时间或日期
TIMR	进程使用总的 CPU 时间
CMD	正在执行的命令行
NI	Nice 优先级
PRI	进程的优先级值
PPID	父进程的 PID
WCHAN	进程因何内核函数而休眠，该函数名从 /boot/System.map 文件中检索
FLAGS	与进程相关的数字标志

②**jobs** 显示当前尚未完成的作业

显示如下：

[2]+Done find /-name fox.jpg

[1]-Stopped man ls&

[]中是作业号；加号表示是当前默认作业；**Running**、**Done**、**Stopped** 分别表示正在运行、完成、挂起。

③在后台执行命令的方法

在命令行的末尾加上一个**&**。

将运行时间较长的程序放在后台运行可以腾出前台继续进行其它操作。如：查找 **fox.jpg** 文件是个较长的过程，可以使用命令：

find / -name fox.jpg &

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

使其在后台运行，系统将显示：[1] 2551

[1]是任务的后台作业号，2551 是进程标识 PID。

（注：宜改用第 13 章的示例程序 beep1 来演示后台操作）。

④作业的前后台切换

fg　将作业切换到前台

bg　将作业切换到后台

ctrl+z　挂起当前作业

fg、bg 默认针对当前作业。可使用参数指定其它作业，如 bg %1 将 1 号作业放于后台。

后台的作业无法接收键盘的输入，若需要接收键盘的输入则将被挂起。

实际上并不能直接使用 fg 命令将当前前台作业转为后台运行，因为这时的键盘输入被视为对作业的输入而不是对 shell 的输入。解决方法是：

先 ctrl+z 挂起当前作业，再执行 bg 命令。

⑤ sleep 使进程暂停一段时间

格式：sleep　以秒为单位的时间值

此命令大多用于 shell 编程中，使两条命令执行之间停顿指定的时间。

例：sleep 100; who |grep　‘z3’

使进程先停 100 秒，再查看 z3 是否正在使用系统。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

⑥ **kill** 向进程发信号

•信号概念：信号是一种机制，进程通过信号知道系统中正在出现的事件，进程在任何时候都能接收到信号。信号通常用来要求进程中断常规运行而做某种操作，大多数进程在收到信号时的默认操作都是结束接收信号的进程，或其它操作。

Linux 的可用信号：（兰色表示常用）

信号名	值	说明
SIGHUP（HUP）	1	挂起。常使用该信号指示进程重载配置文件。
SIGINT（INT）	2	中断。
SIGQUIT（QUIT）	3	退出。
SIGILL（ILL）	4	非法指令。
SIGTRAP（TRAP）	5	跟踪陷阱。
SIGIOT（IOT）	6	IOT 指令。(input / output TRAP)指示一个实现定义的硬件故障。
SIGBUS（BUS）	7	系统总线错。
SIGFPE（FPE）	8	浮点运算错。
SIGKILL（KILL）	9	KILL。该信号不能被捕获（即由进程处理）、阻塞、忽略。
SIGUSR1（USR1）	10	用户自定义信号 1。
SIGSEGV（SEGV）	11	段错误。
SIGUSR2（USR2）	12	用户自定义信号 2。
SIGPIPE（PIPE）	13	写入一个管道，但该管道没有被其它进程读取。
SIGALRM（ALRM）	14	闹钟。
SIGTERM（TERM）	15	软终止信号。常在发 KILL 信号前发此信号给进程，进程在收到此信号后准备退出。
SIGSTKFLT	16	协处理器栈错误。
SIGCHLD（CHLD）	17	子进程状态已改变。
SIGCONT（CONT）	18	收到 STOP 信号后继续。该信号不能被阻塞。
SIGSTOP（STOP）	19	停止。该信号不能被捕获（需要由进程自己处理）、阻塞、忽略
SIGTSTP（TSTP）	20	从键盘产生的 STOP 信号。
SIGTTIN	21	尝试从后台读取控制终端。
SIGTTOU	22	尝试从后台写入控制终端。
SIGURG	23	套接字上出现紧急情况。
SIGXCPU	24	超过 CPU 时间限制。
SIGXFSZ	25	超过文件大小限制。
SIGVTALRM	26	虚拟定时器报警。
SIGPROF	27	Profiling 定时器报警。
SIGWINCH	28	窗口大小改变。
SIGIO	29	在某个文件描述符上可以进行 I/O 操作。
SIGPWR	30	电源失效。
UNUSED	31	未使用。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

•**kill** 命令用作向进程发信号，常用来终止进程。

终止前台进程可按 **ctrl+c**，但终止后台进程需用 **kill**。

kill 的命令格式：

kill [信号名或信号值] 进程 PID

如：**kill 999** 发**-15** 信号终止 **PID=999** 的进程

若不成功，可：**kill –9 999** 杀死 **PID=999** 的进程。

•使用 **killall** 命令代替 **kill** 可直接指定程序名：

例：**killall –9 find**

杀死所有与 **find** 命令有关的进程实例，尽管这些实例有不同的 **PID**。

• **kill 0** 可撤消所有后台作业

十、监视系统负荷

top 命令用来动态显示系统的运行状况。

按 **q** 结束监视退回 **shell**。

十一、

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第八章 文件处理命令

一、文本编辑器 vi

- vi 是 Unix 类操作系统中最为流行的文本编辑器。尽管目前已有 gedit 等一些工作在图形界面下使用起来也更为方便的文本编辑器，但在很多情况下，vi 这种专为字符界面操作而设计的编辑器恐怕还是要充当首选，比如通过 telnet 网络登录来使用系统时。
- 要使用 vi 编辑一个文本文件，键入：

  vi 文件名
- 在无法使用光标键的情况下，可用 k、j、h、l 上下左右移动光标到指定位置，按 i 键进入插入状态，进行编辑；
- 按 esc 键退到编辑状态，按“:wq”存盘并退回到 shell；或按”:q!”放弃对文件的修改而退回到 shell。
- 有关 vi 的更具体的操作，请详见附录。

二、有关查看文件内容的操作命令

① cat 连接多个文件并输出到标准输出设备

例：cat a1.txt a2.txt >a.txt 将两文件的内容合并写到 a.txt

更经常地是被用来显示文本文件的内容：

例： cat a.txt 将 a.txt 内容显示到终端

说明：⑴若不指定文件名，cat 将从标准输入读取数据；

⑵可用 ctrl+s、ctrl+q 来停止滚屏和恢复滚屏；

⑶参数-b、-n、-s 可对输出加上行号和合并相邻空行。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

② **more**　分屏显示

用法：**more** 文件名

翻屏操作：空格键：下翻一屏　　**b:** 前翻一屏

回车键：下翻一行　　**q:** 退出

③ **less**　分屏显示（类同 **more**）

用法：**less** 文件名

翻屏操作：空格键：下翻一屏　　**b:** 前翻一屏

回车键：下翻一行　　**q:** 退出

（实际上，**less** 与 **more** 有些细微区别。可参见具体说明）

④ **head**　显示指定文件的开头若干行

用法：**head [-n]** 文件名　　其中 **n**=行数，默认值为 **10**

例：**less –5 a.txt**　　显示 **a.txt** 的开头 **5** 行

⑤ **tail**　显示指定文件的末尾若干行

用法：**tail [-n]** 文件名　　其中 **n**=行数，默认值为 **10**

例：**tail –5 a.txt**　　显示 **a.txt** 的末尾 **5** 行

⑥ **od**　以八进制显示文件内容

用法：**od [选项]** 文件名

常用的选项：**-tc**　以 **ASC** 码和转义序列显示文件内容

-tx　以 **16** 进制显示文件内容

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

⑦ **sort** 以排序形式显示文件内容

用法：**sort** [选项] 文件名

三、用于查找、比较、统计、甄别文件的操作命令

①**grep**、**egrep**、**fgrep**　在指定文件中查找符合模式的内容

用法：**grep** [选项] [查找模式] [文件 1，文件 2，…]

常用选项：**-E** 将查找模式看成是扩展的正则表达式

-F 将查找模式看成是单纯的字符串

-i 匹配比较时不区分大小写

-r 以递归方式查询目录下所有子目录中的文件

-n 加上行号

例：**grep** ‘z3’ **/etc/passwd**　查找 z3 的账号信息

grep –Fn z3 /etc/passwd　查找 z3 的账号信息及所在行

ps -aux|grep z3　找出所有与 z3 有关的进程

注：**grep** 一次只能搜索一个指定的模式；**egrep** 可以使用扩展的字符串模式进行搜索；**fgrep** 命令检索固定字符串，它不识别正则表达式，是快速搜索命令。这三个命令在指定文件名时都可使用通配符。

② **find** 搜索文件

用法：**find** [路径] [表达式]

说明：⑴未指定路径时为当前目录。

⑵常用**-name** 参数指定字符串作为表达式

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

例：**find / –name vnc*** 从根开始找出所有冠以 **vnc** 的文件

(注：**locate** 命令也可用来定位查找文件，执行速度要快地多。但有时需要执行 **updatedb** 更新其数据库**)**

③ **whereis** 查找命令相关文件所在的位置

例：**whereis ls** 朝找出与 **ls** 命令相关文件的位置

④ **which** 在环境变量**$PATH** 设定的目录中查找文件

例：**which vncserver** 依**$PATH** 找出 **vncserver** 所在目录

注：似乎不能使用通配符。

⑤ **diff** 逐行找出两个文件的不同之处，完全相同则无输出

用法：**diff** [选项] 文件一 文件二

常用选项：**-b** 忽略行尾空格

-c 带上下文的三行输出格式

-r 若比较的是目录，递归比较其子目录

注：若用“-”取代文件名，则意味取标准输入。

⑥ **cmp** 比较两个文件是否相同

cmp 文件 **1** 文件 **2**

⑦ **wc** 统计文件的行数、字数和字节数

用法：**wc** [选项] 文件一 文件二 …

显示：行数、字数、字节数、文件名

选项**-l**、**-w**、**-c** 分别表示按行数、字数、字节数统计

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

默认的情况相当于这三个选项都选。

例：**wc a.txt b.txt**　　显示两个文件的行数字数字节数。

⑧ **file**　识别文件类型

例 1：**file　　/etc/passwd**

显示：**/etc/passwd：ASCII text**

例 2：**file　/boot/vmlinuz***

显示：**/boot/vmlinuz：symbolic link to vmlinuz-2.4.20-8**

/boot/vmlinuz-2.4.20-8：x86 boot sector

注：⑴**cat、more、less、head、tail、od、sort、grep、find、diff、wc** 等命令在未指定文件名时的默认输入都是读自标准输入，都具有过滤器特性。如 **more a.txt、more <a.txt、cat a.txt>more** 效果一样。

⑵本节只是对一些常用命令作入门性的概括介绍，有关命令的更多的一些具体选项请参见其手册信息。

四、特殊过滤命令 tee

tee 是 **T** 的发音，表示象 **T** 型管道那样一头进两头出，把从标准输入读取的数据送到标准输出的同时也送往文件。

例 1：**tee a.txt**　直接将键盘输入(**ctrl+d** 结束)生成文件 **a.txt**。（用法同 **dos** 的 **copy con**，但 **dos** 是 **crtl+z** 或 **F6** 结束。）

例 2：**cat /etc/passwd|head –5|tee –a a.txt**

显示账号文件的前 **5** 行并追加到文件 **a.txt**。**-a** 参数表示追加。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第九章 shell 编程

一、编写与执行 Shell 脚本

- shell 脚本的作用类似于 DOS 的批处理文件，但无文件名规定。
- shell 脚本是文本文件，可用 vi、gedit 或其它文本编辑去创建。
- 脚本的首行应是#!/bin/sh，指明该用什么程序来解释该脚本。
- 使用 chmod +x 命令为脚本文件增加可执行属性
- 执行：./脚本文件名（除非修改主目录下的.bash_profile 文件）

二、部分要点的实例导引

①命令的串联：||和&&

例：cd hhh||mkdir hhh; cd hhh

②.命令，表示在当前系统环境下执行后面的命令

③空设备/dev/null　　无底洞而不是废纸篓

④系统变量

⑤eval　求值，例：eval chmod 777 ~$1

$( )　将命令的结果赋给变量，例：result=$(ls –l /boot)

$(( )) 数学计算，例：echo $((2+3))

三、条件表达式和流程控制

对于熟悉编程概念的专业人士，不过是另外一套规则约定。需要时不难自学，这里暂时从略。

四、

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第十章 Linux 的启动过程

一、常见的三种 Linux 引导方式

① **lilo/grub** 引导，可实现多重系统引导，应用较普遍；

② **loadlin** 引导，可在 **dos** 下引导 **linux**；

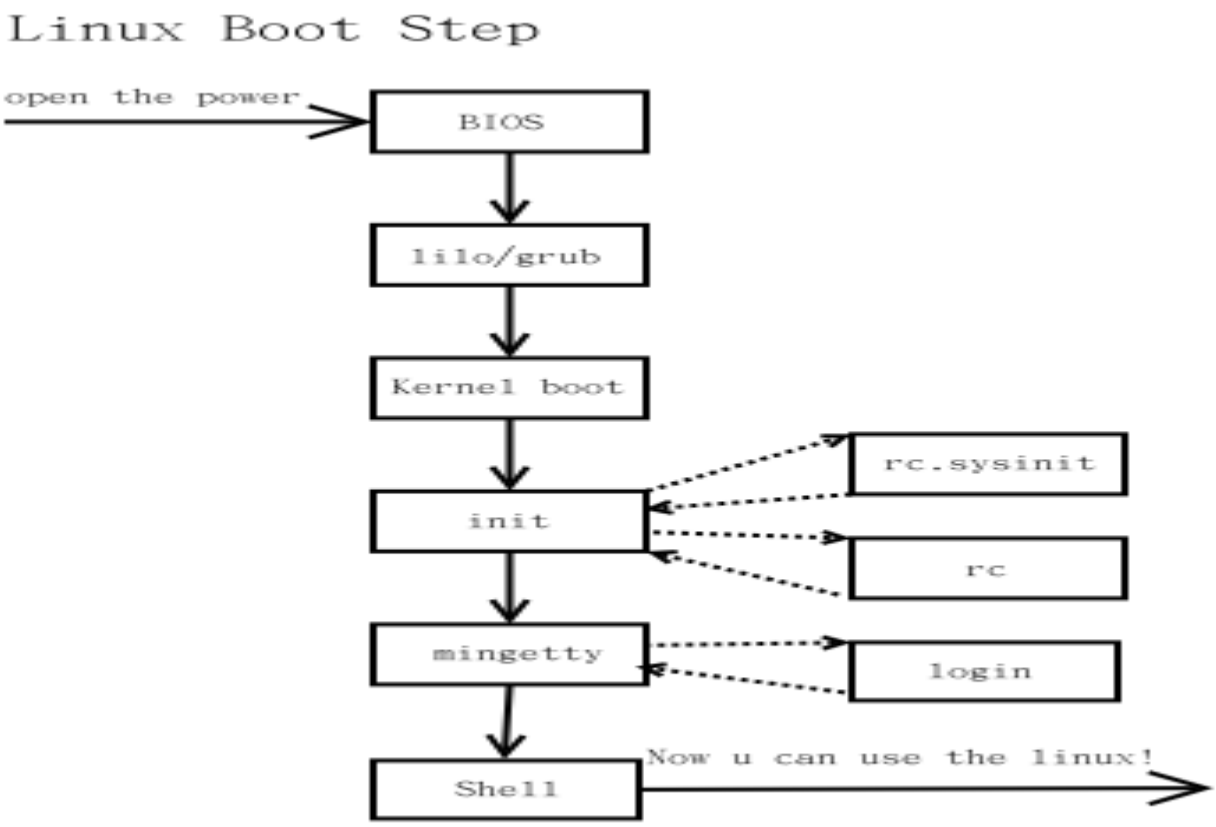
例：将 **loadlin.exe** 和 **vmlinuz** 拷贝到一张 **DOS** 启动软盘上，启动后执行：**loadlin vmlinuz root=/dev/hda8**

则可从软盘启动硬盘上根目录为/dev/hda8 的 **Linux** 系统。

(vmlinuz 是 **Linux** 的压缩内核文件，在/boot 目录下可找到)

③ **linux** 内核自带的 **bootsect-loader**，代码短小，但只能实现 **linux** 的引导，在普通的应用场合很少使用。

二、启动过程简述



Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

⑴加电后执行 BIOS 自检，读取并执行 MBR 中的引导代码；

⑵由 mbr 中 lilo/grub 或其它引导器解压内核 vmlinuz 到内存；

⑶内核检测硬件设备，以只读方式挂装根目录以检测文件系统，若无误则该用读写方式挂装文件系统；

⑷内核启动 init 进程，该进程 PID=1，是内核启动的唯一进程；

⑸init 进程读/etc/inittab 配置文件，获取默认的运行级。对于安装了 x window 的系统通常是 id:5:initdefault；否则通常是 id:3:initdefault；

⑹init 进程运行/etc/rc.d/rc.sysinit 脚本，完成激活交换分区，检查磁盘，加载硬件模块、配置网络等一些系统初始化的工作，以及其它一些需要优先执行任务；

⑺init 进程以运行级别为参数运行/etc/rc.d/rc 脚本，该脚本进入与运行级别对应的目录（如/etc/rc5.d），先以 stop 为参数顺序运行所有以 K 打头的脚本，停止相应服务，再以 start 为参数顺序运行所有以 S 打头的脚本文件，启动相应服务；

⑻运行/etc/rc.d/rc.local 脚本，该脚本缺省的情况下只有一条语句：touch /var/lock/subsys/local。即修改该文件的时间属性；

⑼init 进程为每个终端运行一个/sbin/mingetty 的程序，该程序在终端初显示提示符，等待用户登录；

⑽用户登录后，/sbin/mingetty 交控制权给 login 程序。Login 程序负责建立用户环境，并启动一个 shell，首先寻找执行/etc/profile 和用户的主目录中的.bash_profile 脚本文件；

⑾ shell 负责解释执行用户命令。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

三、自动运行程序

①开机启动时自动运行程序

修改放置在/etc/rc.d 或 /etc/rc?.d 目录下的脚本文件，可以使 init 自动启动其它程序。例如：编辑 /etc/rc.d/rc.local 文件，在文件最末加上一行“xinit”或“startx”，可以在开机启动后直接进入 X－Window。

②登录时自动运行程序

用户登录时，bash 首先自动执行系统管理员建立的全局登录 script ：/ect/profile。然后 bash 在用户起始目录下按顺序查找三个特殊文件中的一个：/.bash_profile、/.bash_login、/.profile，但只执行最先找到的一个。因此，只需根据实际需要在上述文件中加入命令就可以实现用户登录时自动运行某些程序（类似于 DOS 下的 Autoexec.bat）。

③退出登录时自动运行程序

退出登录时，bash 自动执行个人的退出登录脚本/.bash_logout。例如，在/.bash_logout 中加入命令“tar －cvzf c.source.tgz ＊.c”，则在每次退出登录时自动执行“tar” 命令备份 ＊.c 文件。

④定期自动运行程序

Linux 有一个称为 crond 的守护程序，主要功能是周期性地检查 /var/spool/cron 目录下的一组命令文件的内容，并在设定的时间执行这些文件中的命令。用户可以通过 crontab 命令来建立、修改、删除这些命令文件。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

例如，建立文件 crondFile，内容为“00 9 23 Jan ＊ HappyBirthday”，运行“crontab cronFile”命令后，每当元月 23 日上午 9:00 系统自动执行“HappyBirthday”的程序（“＊”表示不管当天是星期几）。

⑤定时自动运行程序一次

定时执行命令 at 与 crond 类似（但它只执行一次）：命令在给定的时间执行，但不自动重复。at 命令的一般格式为：at [－f file] time ，在指定的时间执行 file 文件中所给出的所有命令。也可直接从键盘输入命令：

```
$ at 12:00
at>mailto Roger －s ″ Have a lunch″ ＜plan.txt
at>Ctr－D
Job 1 at 2000－11－09 12:00
```

2000－11－09 12:00 时候自动发一标题为“Have a lunch”，内容为 plan.txt 文件内容的邮件给 Roger。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第十一章 升级与定制

一、RPM 包安装操作

RPM（Redhat Packet Manager）。

①安装 rpm –i rpm 文件名

（注：⑴常见用法：-ivh 参数显示安装过程和 hash 符#；

⑵覆盖安装：使用- -force 选项。）

②卸载 rpm –e rpm 文件名

③更新 rpm –U rpm 文件名

④查看 rpm 信息 rpm –q 包名称

例：rpm –q apache 查看 apache 的安装情况

若查看系统中安装的所有软件包，需要使用-a 选项。

-q 选项可以结合其它字符以获得更为详细的信息：

rpm -qc	列出某软件包上的所有配置文件
rpm -qd	列出某软件包上的所有文档文件，通常是手册页
rpm -qi	列出软件包详细信息（版本号大小安装日期等）
rpm -ql	列出软件包中所有文件
rpm -qs	列出软件包中所有文件的状态

二、定制内核

准备：执行 uname –r 了解内核版本；执行 rpm –q kernel-source 查看内核源码是否已经安装；以 root 身份进入 /usr/src/linux-2.4，此后命令均需在此目录下执行；（linux-2.4

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

是 linux-2.4.20-8 的软连接），之后按以下步骤进行：

① make mrproper，清除以前编译内核时残留的.o 文件和不必要的关联遗迹；（.config 是上次的配置文件）

②从/usr/src/linux-2.4/configs 目录下选择默认的入手配置，如

cp /configs/kernel-2.4.20-i686.config .config

③ make xconfig 在 x 界面下配置内核；

也可以选择下列之一的字符界面方式：

- make menuconfig 文本界面下的菜单方式；
- make config 互动文本方式
- make oldconfig 非互动脚本,便于按默认值设置内核

④ make dep 正确设置关联文件；

⑤ make clean 准备要构建的源码树；

⑥ make bzImage 编译内核（对于大内核，比如需要 SCSI 支持），或 make zImage（对于小内核）；编译结束后在/usr/src/linux-2.4/arch/i386/linux/boot/下产生 bzImage 文件

⑦ make modules 编译模块；

⑧ make modules_install 安装模块（即便什么也没做）；

⑨ make install 安装新内核。在 /boot 目录中安装内核文件，这个命令还执行 /sbin/new-kernel-pkg 脚本。该脚本会建构一个新的 initrd 映像，并在引导装载程序的配置文件中添加一个新项目。

三、

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第十二章 远程登录与 NFS

一、Telnet

- Telnet 是 Internet 提供的一项基本服务，用来将本地计算机作为网络中另一主机的一个远程终端使用，作用上与本地终端无异。Telnet 协议是 TCP/IP 协议族中的一个应用层协议。
- 同 Internet 上的所有服务一样，Telnet 也工作在客户机/服务器模式，这就意味着服务方必须运行着服务软件而客户方运行着客户端软件，目前的绝大多数操作系统均已以系统外部命令的方式提供了 Telnet 的客户端软件。

•Telnet 用法很简单：telnet 主机名或 IP 地址，

例：telnet 202.207.30.136

依提示输入用户名和口令后即可登入远程主机的系统。

- exit 或 logout 命令结束 telnet。

•注意：

⑴telnet 不允许使用 root 身份登录。若必须执行 root 权限的操作只能先以一个普通用户账号登录后再执行 su 操作。

⑵telnet 以明文方式传输数据，很不安全。

(注：微软系统内的 telnet 可使用 NTLM 加密方法进行身份认证，但 Linux 的 telnet 仍遵循标准的 telnet 协议。)

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

二、SSH（Secure SHell）

- SSH 以 RSA 公开密钥算法加密数据，并且进行了压缩。
- Linux 实现了 SSH 的服务端和客户端软件。
- SSH 可视作 telnet 的替代，用法类似。

  例：SSH　202.207.30.135
- 常用参数-l 指定用户名，如 SSH 202.207.30.136 –l z3
- 说明：

  ⑴SSH 是安全的，所以允许以 root 身份远程登录。

  ⑵Windows 没有 SSH 命令，需安装使用第三方软件。如：

  - SecureCRT v3.4　(支持 SSH 的字符终端)
  - Netsarang Xmanager v2.0　(SSH 字符终端/远程桌面)
  - F-Secure SSH Client for Windows 5.3　(SSH+FTP)等。

  ⑶一些常用的 FTP 软件已支持 SSH，如 WS-FTP Pro 8.1、Xftp 1.2 等。当使用这些软件往 Linux 系统主机上传时应注意设置为 SSH 认证方式（但 WS-FTP Pro 8.1 尚不支持 root 身份连接，若需安全上传 root 文件时可选用后者）。

  ⑷若先 telnet 到主机 A 再 SSH 到主机 B，则 AB 之间是密文而本机与 A 之间是明文（有文章建议在 Windows 客户机上可先 telnet 到 A 再 SSH 到 A，而实际上这毫无意义！）

  ⑸Linux 兼容 Unix 原有的远程登录命令 rlogin（该命令使用 rlogin 协议），但不太提倡使用。应尽量采用 SSH 作为 tenlnet 和 rlogin 的安全替代。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

三、scp（Secure Copy）

scp 命令可用于本地与远程计算机间安全地互相拷贝文件。

格式：scp 本地文件路径 远程用户名@主机:文件路径

或： scp 远程用户名@主机:文件路径 本地文件路径

例：scp z3@202.207.30.136:a.txt . 将 a.txt 从远程拷到本机

scp a.txt z3@202.207.30.136: 将 a.txt 从本机拷到远程

说明：⑴执行时会要求你输入远程用户的口令；

⑵同 cp 一样，可用-r 参数递归地拷贝目录；

⑶实验表明 scp 不作覆盖提示，需小心使用；

⑷scp 可作为 Unix 中 rcp 命令的安全替代。

四、sftp（Secure FTP）

- sftp 是 ftp/rftp 命令的安全替代，用法同 ftp 一样。
- sftp 可用 scp 同样的用法传输单个文件；但，
- 更常见的用法是：sftp [远程用户名@]远程主机 <回车>

进入 sftp>状态。这时可执行该状态下的命令。

如：get remote-path [local-path] 下载文件

put local-path [remote-path] 上传文件

quit 退出 sftp 状态

- 在 sftp 状态下可打？号列出该状态下可用的命令清单。 其中不少命令同 Linux bash 命令，如 mkdir、cd、pwd、ls 等，但这些命令是在对被登录机操作，对本地机操作的命令会被冠以字母 l，如 lmkdir、lcd、lpwd、lls 等(但有 rm 没有 lrm)。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

- **sftp** 为命令行操作提供了足够的本地与远程操作命令。**Linux** 的图形 **FTP** 软件(如 **gftp**)和 **Windows** 平台的一些 **FTP** 软件(如 **Xftp**)也都支持 **ssh/sftp** 协议，安全且操作方便。

五、VNC（Virtual Network Computing）

- 一款优秀且免费开源的远程桌面控制工具软件，由著名的 **AT&T** 的欧洲研究实验室开发。其使用的通信协议 **RFB**（**Remote Fram Buffers**）是一个开放协议。VNC 已被移植到 Windows 等操作系统。
- 远程图形；数据经过加密；带宽需求小，拨号上网也可使用。
- Redhat Linux 9.0 自带 VNC3.3.3 版的服务端和客户端程序。
- 服务端操作：

```
#vncserver                                    启动 vncserver,
New 'X' desktop is cstd136:1
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/cstd136.1.log
```

注意：

⑴记住首行末尾冒号和数字表示的窗口号（如这里的 :1）；

⑵root 或普通用户均可启动自己的 vncserver，vnc 默认按启动顺序赋予窗口号；

⑶若该用户是第一次执行，系统会提示创建 vnc 口令；（以后可执行 vncpasswd 命令修改自己的 vnc 口令）

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

⑷VNC 的默认分辨率是 1024x768，由/usr/bin/vncserver 配置文件中的$geometry=”1024x768”语句决定。可在启动时用-geometry 参数指定所用分辨率。

⑸关闭服务端进程的命令：vncserver －kill <X-display>

例：vncserver －kill :2

- 客户端操作：

①Linux 客户端有两个 VNCviewer 可用，在 gnome 中：

可：主菜单->附件->更多附件->VNC Viewer

或：主菜单->->互联网->远程桌面连接

在 VNCserver 或远程桌面对话中填入服务端 IP 和窗口号：

例：202.207.30.136:1

之后回答口令，即可开始远程桌面操作。

②Windows 客户端需下载并安装 x86 Win32 viewer，

（ftp://ftp.cstd.impu.edu.cn/Linux/Remote%20Desktop/VNC/index.htm）

③用浏览器作为 VNC 客户端(http 协议,端口:5800+窗口号)

例：http://202.207.30.136:5801

注：⑴VNC 的服务不会因客户端的关闭而终止。若服务端一直持续运行，客户下次连接时将看到上次离开时的现场。

⑵尽管 VNC 安全方便，但需要在使用前以注册的用户身份启动服务端的服务，因此就得先 SSH 到服务器，执行 VNC 启动命令 vncserver 后再开启 VNC Viewer。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

六、NFS（Network File System）

①NFS 的基本概念：

- NFS 通过网络将远程主机的文件系统安装到本地主机，使本地主机存取远程主机上的文件如同存取本地文件一样。
- 应用效果上类似早期基于服务器的局域网中对磁盘服务器或文件服务器的使用，但 NFS 是基于客户机/服务器模式的，
- NFS 是一种工作于 TCP/IP 的应用层软件，建立在表示层协议 XDR（External Date Representation）和会话层协议 RPC（Remote Procedure Call）机制之上。（XDR 规定了数据的表达与传输格式；RPC 提供了允许应用程序调用在远程主机上运行的例程的接口。）
- 服务器上为 NFS 客户提供的共享目录称之为“导出目录”，导出（exporting）目录可以是整个硬盘分区也可以是一个子树；将已经共享的远程目录挂装到本地文件系统的过程称之为“挂装（mounting）目录”。
- NFS 是目前 Linux 嵌入式系统开发过程中的主要平台手段。

②服务器端的配置操作：

- 启动 NFS 服务：/etc/rc.d/ini.dt/nfs start

  停止 NFS 服务：/etc/rc.d/ini.dt/nfs stop
- 配置 NFS

  ㈠gnome 图形界面操作：

  主菜单->系统设置->服务器设置->NFS 服务器->添加

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

在目录栏中指定导出目录，

主机栏中指定客户的主机名或 IP 地址。确定。

㈡文本操作：

修改/etc/exports 文件，增加如下内容

/home/z3　　202.207.30.137(ro,sync)

/home/l4　　202.207.30.138(rw,sync)

/home/w5　　*(rw.sync,all_squash,anonuid=533)

/home/h5　　*(ro,sync)

表示

/home/z3 可供 **202.207.30.137** 以只读方式安装

/home/l4 可供 **202.207.30.138** 以读写方式安装

/home/w5 可供任何主机以被指定本地 **UID** 的匿名读写方式安装（**533** 是 **z3** 的 **UID**）

/home/w5 可供任何主机以只读方式安装

（**sync** 表示同步写操作）

③客户端的挂装操作：

•以 root 身份登录，在/mnt 下建一 mynfs 目录，执行：

mount 202.207.30.136:/home/z3 /mnt/mynfs

（服务器为 202.207.30.136，导出目录/home/z3，挂装在本地/mnt/mynfs 目录，之后本机所有用户都可以通过对/mnt/mynfs 目录的访问来访问实际上在服务器上的/home/z3 目录。）

•使用完毕应对其卸载：umount /mnt/mynfs

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

注：⑴LinuxNFS 要求服务器和客户机都须是 Linux 系统。

⑵实验证实，导出目录的属性应设置为允许他人访问，否则需设置为“把所有客户用户当作匿名用户”并“为匿名用户指定本地用户（或组群）ID”的匿名访问方式。

⑶NFS 根据主机名而非用户名来限制权限，安全性欠佳。

七、总结

- telnet 是 internet 提供的标准远程登录服务；

  FTP 是 internet 提供的标准文件传输服务；

  rlogin 和 rftp 是 Unix 系统的远程登录和文件传输服务；
- 在需要安全的场合下：

  SSH 可作 telnet 的安全替代，但跨平台应用需第三方软件；

  scp、sftp 比较适宜在 Linux 系统间以字符方式应用；
- VNC 提供加密的远程图形桌面，但需注意启动服务时的安全；
- NFS 为嵌入式系统开发提供了一种平台手段。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第十三章 GCC 编程入门

一、gcc 基础

①简介

- **gcc（GNU C Compiler），是 GNU 项目的编译器套件。**
- **gcc 对源码进行预处理、编译、汇编和连接，生成可执行文件。**
- **gcc 调用 cc1 进行编译，生成汇编语言程序。但若非指定则立即调用 as 进行汇编生成具有.o 扩展名的目标文件(所以这两步也常被一起统称为编译)。中间的汇编语言输出不保存在文件中，除非使用-S 参数指定。**

  (-S 参数用于输出以. s 作扩展名的 Linux 汇编语言文件)
- gcc 常用参数：

  -c 只编译生成目标文件，不连接

  -o 指定输出文件名。若不指定默认的可执行文件名是 a. out。

  -g 产生用于 GDB 调试和排错的扩展符号表(毋与优化同用)

  -O 代码的长度和速度优化

  -O2 更好的速度优化

②编译示例

- 第一个示例程序 **hello.c**，如下：

```
#include <stdio.h>

int main(void)
{
   printf(“Hello,world!\n”);
   exit(0);
}
```

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

编译并连接：gcc –o hello hello.c

执行：./hello

显示：Hello,world!

(注：在 Linux 中执行当前目录中的文件须指定“./”，除非修改主目录下的.bash_profile 文件，在 PATH 语句末尾加上“:.”，但基于安全考虑不提倡这种做法。)

- 第二个程序示例多模块文件编译，拆 hello.c 为两个文件：

```
//helloa.c        第一个文件
exitern void disp(char *);
int main(void)
{
   disp_str(“Hello, world!/n”);
    exit(0);
}

//hellob.c        第二个文件
#include <stdio.h>
void disp_str(char *str)
{   printf(“%s”,str);  }
```

编译：gcc –c helloa.c hellob.c　　(或 gcc –c hello?.c)

　　这一步生成 helloa.o 和 hellob.o 文件

连接：gcc –o hello helloa.o hellob.o　(或 gcc –o hello?.o)

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

执行：**./hello**

③Makefile 文件

- Makefile 文件可有效地控制多模块文件的项目编译。
- 使用 make 命令启动应用了 Makefile 文件的项目编译过程。
- 若非使用-f 选项指定 make 所用的 makefile，make 将在当前目录下以 GNUmakefile、makefile、Makefile 的顺序寻找。资料中一般都建议使用 Makefile 文件名。
- Makefile 的编写规则：

  暂且理解为：自顶向下分步求精地结构化表述，句法是：

  目标：与此目标直接相关的依赖文件列表

  如何利用这些依赖文件生成目标

  例：第二个示例程序的 Makefile 文件，可以如下编写：

```
hello: helloa.o hellob.o
        gcc -o hello helloa.o hellob.o
helloa.o: helloa.c
        gcc -c helloa.c
hellob.o: hellob.c
        gcc -c hellob.c
```

  执行 **make** 命令后即可生成 **hello** 可执行文件。

  (如果使用了用户头文件，应排在与目标有关的文件列表行中。如：**helloa.o: helloa.c helloa.h)**

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

- Makefile 文件的简化：

  Makefile 有几个很有用的变量，其中常用的三个意义分别是：

  $@ --目标文件，

  $^ --所有的依赖文件，

  $< --第一个依赖文件.

  利用这三个变量可以简化我们的 Makefile 文件为：

```
hello:      helloa.o hellob.o
            gcc -o $@ $^
helloa.o:   helloa.c
            gcc -c $<
hellob.o:   hellob.c
            gcc -c $<
```

  利用 **Makefile** 的缺省规则 **.c.o: gcc -c $<**(这个规则表示所有的 **.o** 文件都是依赖与相应的**.c** 文件) 还可以进一步简化：

```
hello: helloa.o hellob.o
       gcc -o $@ $^
.c.o:  gcc -c $<
```

  (只要修改文件中目标名和目标文件名即可应用于其它项目)。

- 在 Makefile 文件中还可以增加其它目标，比如追加两行：

```
clean:
       rm -f hello *.o
```

  对编译无影响，但执行 make clean 命令可删除所有目标模块。

  (回想一下编译内核章节，是否见过这种用法？)

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

- 也可以利用 automake 和 autoconf 生成 Makefile，但笔者不认为就能很轻松。其详细用法请参见相关资料，这里从略。

二、进程创建与守护程序编程

①本次用于第七章第九节讲解前后台作业管理的道具程序：

```
//beep1.c  每 2 秒响一声，15 声后结束。
#include <stdio.h>
main(void)
{
    int i;
   for(i=0; i<15; i++) {putc(7,stdout); fflush(stdout): sleep(2);}
}
```

说明：⑴ASC 码 7 表示响铃，fflush()用于泻缓冲。

⑵编译：gcc –o beep1 beep1.c

⑶前台运行：./beep1　　后台运行：./beep1&

②子进程和守护程序的概念

- fork()函数用于创建子进程,其原型在unistd.h和stdlib.h，函数格式：int fork(void)
- 子进程是父进程的拷贝，fork()是分支点。分支前的代码只执行一次
- fork()函数的最特殊点是每调用一次返回两次：在父进程中返回所创建的子进程的 ID 值；在子进程中的返回值是 0。因

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

此可以从返回值来判断这两次返回都是在哪个进程完成的。（若出错返回-1）。

- PID(Process Identify Number)，进程标识号。
- Linux 中实现守护程序的一般做法是：父进程在成功 fork()后结束，子进程便成为守护进程(daemon)。

③daemon 化改造后的 beep 程序

```
//beep2.c   每 2 秒响一声，15 声后结束。守护程序。
#include   <stdio.h>
#include   <stdlib.h>
#include   <unistd.h>
main(void)
{
pid_t pid;  int i;
pid = fork();
if(pid<0) {printf(“Can’t fork!\n”); exit(-1);}
if(pid>0){printf(“Daemon (pid=%d) on duty!\n”, pid); exit(0);}
printf(“I am daemon beep2 (pid=%d) !\n”, getpid());

for(i=0; i<15; i++) {putc(7,stdout); fflush(stdout): sleep(2);}
exit(0);
}
```

编译：gcc –o beep2 beep2.c

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

执行：**./beep2**

验证：**jobs** 命令只能显示后台执行的作业，看不到守护进程。查看守护进程要用 **ps**。（通常的守护程序并没有类似此例结束处理，杀死常驻的守护进程要用 **kill** 命令。

三、时间编程

①获取当前的日期与时间

- **#include <time.h>**

  time_t time(time_t *tloc);

  char *ctime(const time_t *clock);

  time 函数返回从 **1970** 年 **1** 月 **1** 日 **0** 点以来的秒数，存储在 **time_t** 结构之中。**ctime** 函数可将上述秒数转化为象 **Thu Dec 7 14:58:59 2000** 这样长度固定为 **26** 的字符串。

```
//示例 now_tml.c：
#include <time.h>
#include <stdio.h>
main()
{  time_t the_time;
   time(&the_time);
   printf("The date and time is %s",ctime(&the_time));
   exit(0);
}
```

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

- 还可用到时间函数 localtime(),显示本地时间的函数:

```
//示例 now_tm2.c:
#include<time.h>
#include<stdio.h>
main()
{   struct tm *tm_ptr;    time_t the_time;

    (void) time(&the_time);
    tm_ptr=localtime(&the_time);
    printf("Raw time is %ld\n",the_time);
    printf("Mytime show:\n");
    printf("Date:%02d - %02d - %02d\n",
    tm_ptr->tm_year,tm_ptr->tm_mon+1,tm_ptr->tm_mday);
    printf("Time:%02d:%02d:%02d\n",
    tm_ptr->tm_hour,tm_ptr->tm_min,tm_ptr->tm_sec);
    exit(0);
}
```

显示结果:

```
Raw time is 1107433723
Mytime show:
Date: 105-02-03
Time: 07:28:43
```

结果的第一行显示的数字是 linux 的原始时间表示。还有一

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

个时间函数 **gmtime()** ，显示的是格林尼治时间，将例子程序的 **localtime()** 换成 **gmtime()**，会与北京时间相差 **8** 小时。

②计时

- 若需测量某个程序执行所花费的时间时，可用下面这个函数：

#include <sys/time.h>

int gettimeofday(struct timeval *tv，struct timezone *tz);

strut timeval { long tv_sec; /* 秒数 */

long tv_usec; /* 微秒数 */

};

gettimeofday 将时间保存在结构 **tv** 中。**tz** 一般可置为 **NULL**。

- 示例

```
//delta_tm.c 测试 function()函数的运行时间
#include <sys/time.h>
#include <stdio.h>
#include <math.h>

void function()
{
   unsigned int i,j; double y;
   for(i=0;i<1000;i++)
    for(j=0;j<1000;j++) y=sin((double)i);
}
```

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

```
main()
{
   struct timeval tpstart,tpend; float timeuse;

   gettimeofday(&tpstart,NULL);
   function();
   gettimeofday(&tpend,NULL);
   timeuse=1000000*(tpend.tv_sec-tpstart.tv_sec)+
            tpend.tv_usec-tpstart.tv_usec;
   timeuse /= 1000000;
   printf("Used Time:%f\n",timeuse);
   exit(0);
   }
```

编译：**gcc -o delta_tm delta_tm.c –lm**

执行：**./delta_tm**　　结果显示：**Used Time: 0.226031**

③系统计时器

- **Linux** 为每个进程提供了 **3** 个内部间隔计时器：
- ITIMER_REAL：实时时钟，到时产生 SIGALRM 信号；
- ITIMER_VIRTUAL：虚拟时钟，只在进程运行时才运行，到时产生 SIGVTALRM 信号；
- ITIMER_PROF：形象时钟，在进程运行时和在系统代表进程运行时都运行，到时产生 SIGPROF 信号。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

- 具体的操作函数是:

  #include <sys/time.h>

  int getitimer(int which, struct itimerval *value);

  int setitimer(int which, struct itimerval *newval, struct itimerval *oldval);

  struct itimerval { struct timeval it_interval;

  struct timeval it_value; }

  getitimer 函数得到间隔计时器的时间值，保存在 **value** 中。**setitimer** 函数设置间隔计时器的时间值为 **newval**，并将旧值保存在 **oldval** 中。 **which** 表示使用三个计时器中的哪一个。**itimerval** 结构中的 **it_value** 是减少的时间，当这个值为 **0** 的时候就发出相应的信号了. 然后设置为 **it_interval** 值。

- ITIMER_PROF 计时器的初始化示例：

```
void init_time()
{ struct itimerval value;
  value.it_value.tv_sec=2;
  value.it_value.tv_usec=0;
  value.it_interval=value.it_value;
  setitimer(ITIMER_PROF,&value,NULL);
}
```

四、信号编程

- 信号机制是在软件层次上对中断机制的模拟。从概念上讲，一个进程收到一个信号和一个处理器接受到一个中断请求其

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

处理过程是类似的（中断当前操作、转入信号处理程序）。

- 信号可以来自其它进程、外部事件、进程自身、或 **kill** 命令。
- 信号的最多种类数与处理器字长有关，**i386** 是 **32** 位，所以定义了 **32** 种信号。
- **POSIX** 下，每个进程有一个信号掩码**(Signal Mask)**。简单地说，信号掩码是一个“位图”，其中每一位都对应着一种信号。为 **1** 表示在执行当前信号的处理程序期间相应的信号被暂时屏蔽，以避免嵌套响应。
- sigaction 函数为 **signum** 指定的信号设置对应的处理程序：

```
#include<signal.h>
int sigaction(int signum, const struct sigaction *act,
                          struct sigaction *oldact);
```

sigaction 结构也在<signal.h>中定义：

```
struct sigaction{
        void (*sa_handler)(int);        //信号处理程序
        sigset_t sa_mask;  //处理期间应阻塞的信号集掩码
        int sa_flags;          //修正 sa_handler 行为的掩码
        void (*sa_restorer)(void);      //这行现已废弃不用
        };
```

- 所以，对 SIGPROF 信号的服务和初始化代码如下：

```
void prompt_info(int signo)     //信号服务程序
  { char *mes = "The time has lost 2 second.\n\a";
    write(STDERR_FILENO, mes, strlen(mes));
```

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

```
}

void init_sigaction(void)      //信号机制初始化
  { struct sigaction act;
    act.sa_handler=prompt_info;
    act.sa_flags=0;
    sigemptyset(&act.sa_mask);
    sigaction(SIGPROF,&act,NULL);
  }
```

- **sigemptyset/sigfillset/sigaddset/sigdelset** 用于初始化信号集。关于信号集处理可参见：《GNU/Linux 编程指南(第二版)》(张辉译 清华大学出版社2002年6月第1版￥68.00)13.5节。

- 完整的 ITIMER_PROF 计时器应用程序示例：

```
//tmr.c
#include <sys/time.h>
#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <string.h>

void prompt_info(int signo)
  { char *mes = "The time has lost 2 second.\a\n";
```

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

```
    write(STDERR_FILENO, mes, strlen(mes));
    }
void init_sigaction(void)
    { struct sigaction act;
      act.sa_handler=prompt_info;
      act.sa_flags=0;
      sigemptyset(&act.sa_mask);
      sigaction(SIGPROF,&act,NULL);
    }
void init_time()
    { struct itimerval value;
    value.it_value.tv_sec=2;
    value.it_value.tv_usec=0;
    value.it_interval=value.it_value;
    setitimer(ITIMER_PROF,&value,NULL);
    }

int main()
    { init_sigaction(); init_time();
    while(1);
    }
```

编译：gcc -o tmr tmr.c

执行：./tmr &

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

显示：[1] 2471

表示：程序 tmr 在以作业号 1、进程号 2471 运行。

结果：该程序每隔两秒钟输出一个提示。

- 实验：

注释掉主函数 **main()**中用于初始化 ITIMER_PROF 计时器的语句 init_time(); 用 kill 命令代其向进程发信号：

kill –27 2471 (27 是 SIGPROF 信号的值)

结果也会出现报警提示。 （以此加深对 **kill** 命令的理解）

五、驱动模块编程示例（示例程序 **drvmod/hello.c**）

安装模块：**insmod hello.o [--force]**

模块列表：**lsmod**

挟载模块：**rmmod hello**

六、**Linux** 用户界面编程简介

- 终端控制：<termios.h> POSIX.1 定义的标准终端接口
- 屏幕控制：ncurses 经典的 Unix 屏幕控制库
- 底层图形：SVGAlib、framebuffer、MiniGUI
- X-Window 编程：Xlib API、X toolkit API 等
- gnome 编程：GTK、GTK+/gnome、Glade
- KDE 编程： QT、QT Designer、KDevelop
- 3D 图形编程：OpenGL、Mesa

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

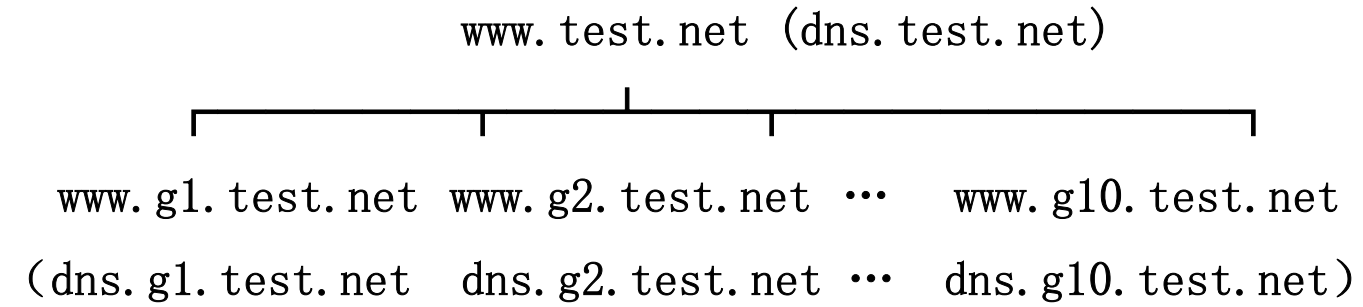
第十四章 实战：构建多级网站

一、目的

通过架设 **Intranet** 多级网站的实例，学习如何利用 **Linux** 开设 **Web** 和 **DNS** 服务器，以及相应的配置管理方法。

二、规划

- 分级结构



- 一台顶级服务器，域名：**www.test.net** 和 **dns.test.net**
- 若干台次级服务器，域名：**www.g1.test.net**、**dns.g1.test.net**
  g1 表示第一组，以此类推。
- 若干台工作站，其 dns 分别只指向顶级和各次级 **DNS** 服务器。
- 实现一个多级的 intranet。

三、实施

以演示讲解为主。具体步骤和相关知识这里暂略。

四、总结

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。

www.linuxidc.com

第十五章 国产方舟 CPU 系统开发板简介

（待续）

- Linux 的优点：①与 Unix 兼容；②自由软件源码公开；③便于定制和再开发；④全面的多任务和真正的 32 位操作系统。
- Linux 的缺点：①命令难记难用，文档晦涩，帮助信息欠佳，用户友好性差；②其多用户特性在桌面应用方面意义不大，运行上并不比 Windows 顺畅，表现尚难与 Windows 匹敌；③发行版本杂乱，缺乏统一规划，各版有所差异，设备驱动程序与内核版本关系密切，需要提供源码进行编译，难以得到设备厂商的广泛支持； ④Linux 在服务器应用上表现较好，但其权限管理机制上不如 W2K/NT 细致灵活；⑤不是一个微内核操作系统；不支持实时处理；⑥Linux 不支持软猫。

- 本教案为 2005.2.21 开课的 03、04 级研究生 Linux 课程班准备。
- 首次勘修稿，仅供本次教学。40 课时内容。欢迎提出意见。

Linux公社(LinuxIDC.com)
是包括Ubuntu,Fedora,SUSE技术，最新IT资讯等Linux专业类网站。