

Redhat Linux 教程

Contents

Linux 基础.....	2
1 Linux 系统简介	2
1.1 Linux 操作系统的特点与优点	2
1.2 Linux 发布版本	3
1.3 Linux 的发展方向	4
2 Linux 的目录和文件系统.....	5
2.1 Linux 目录.....	5
2.2 Linux 文件系统.....	8
3. Linux 系统的登录和退出	12
4 学习 shell 工具	14
5 Linux 常见命令	16
5.1 Linux 命令基础.....	16
5.2 Linux 文件目录类命令	17
6 Vi 编辑器的用法.....	28

北南南北论坛: WWW.VXWORKS6.COM

Linux 基础

1 Linux 系统简介

Linux 是一个以 Intel 系列 CPU (CYRIX, AMD 的 CPU 也可以) 为硬件平台, 完全免费的 UNIX 兼容系统, 完全适用于个人的 PC。它本身就是一个完整的 32 位的多用户多任务操作系统, 因此不需要先安装 DOS 或其他的操作系统 (MS Windows, OS2, MINIX...) 就可以进行直接的安装。Linux 的最早起源是在 1991 年 10 月 5 日由一位芬兰的大学生 Linux Torvalds (Torvalds@kruuna.helsinki.fi) 写了 Linux 核心程序的 0.02 版开始的, 但其后的发展却几乎都是由互联网上的 Linux 社团 (Linux Community) 互通交流而完成的。Linux 不属于任何一家公司或个人, 任何人都可以免费取得甚至修改它的源代码 (source code)。Linux 上的大部分软件都是由 GNU 倡导发展起来的, 所以软件通常都会在附着 GNU Public License (GPL) 的情况下被自由传播。GPL 是一种可以使你免费获得自由软件的许可证, 因此 Linux 使用者的使用活动基本不受限制 (只要你不将它用于商业目的), 而不必像使用微软产品是那样, 需要为购满许可证付出高价还要受到系统安装数量的限制。

1.1 Linux 操作系统的特点与优点

Linux 可以与 MS-DOS、OS/2、Windows 等其他操作系统共存于同一台机器上。它们均为操作系统, 具有一些共性, 但是又各有特色, 有所区别。

目前运行在 PC 机上的操作系统主要有 Microsoft 的 MS-DOS、Windows、

Windows NT、IBM 的 OS/2 等。早期的 PC 机用户普遍使用 MS-DOS，因为这种操作系统对机器的硬件配置要求不高，而随着计算机硬件技术的飞速发展，硬件设备价格越来越低，人们可以相对容易地提高计算机的硬件配置，于是开始使用 Windows、Windows NT 等具有图形界面的操作系统。Linux 是新近被人们所关注的操作系统，它正在逐渐为 PC 机的用户所接受。

Linux 的核心具有 Windows 无法比拟的稳定性和高效率，在不使用 X Windows 的情况下，它占用系统资源较少，可以使一台 Intel486 摇身一变成为高效工作站。对于想要学习 UNIX 的用户来说，使他们熟悉 UNIX 操作环境，通往“骨灰级”高手境界的一大捷径。最重要的是，Linux 上有公认的 C 语言世界最好的编译器 gcc，如果想得到一个效率更高的开发环境（并不一定是最友好的，但一定是最强大的），建议你一定要好好的熟悉一下 Linux。

1.2 Linux 发布版本

任何一个软件都有版本号，Linux 也不例外。Linux 的版本号又分为两部分：内核(kernel)与发布套件(distribution)版本。虽然 Linux 只有一个标准化的版本，但存在着好几个不同的发布版本。Linux 的发布版本就是将 Linux 核心与应用软件打成一个包，发布版本的不同主要是指不同的公司和组织在组织打包 Linux 软件的时候稍有差异，各个公司或组织通常会使用 CD-ROM 发布它们的 Linux 包。目前已有多个发布版，包括 Ubuntu、Fedora、RedHat、Mint、OpenSUSE、Mandriva、CentOS 和 Debian 等。尽管这些发布版之间有着各种各样的差别，但他们都依赖于 Linux 内核、使用程序和应用程序。

Redhat 是由 Red Hat Software 公司推出的一套优秀的 Linux 发行版本。它更多地考虑到新用户对系统配置与安装过程普遍生疏，因此在安装过程上做了改进，使 Linux 系统初级用户可以在友好界面指引下安装好 Linux 系统。同时在安装过程中，Red Hat Linux 为用户配置完成许多硬件设备，比如最让头疼的 XWindow 配置工作，对网卡、声卡及其他一些不常用设备的配置，在安装过程中要求用户回答的问题很少，避免了新用户选择时的困惑。

Ubuntu（乌班图）是一个以桌面应用为主的 Linux 操作系统，其名称来自非洲南部祖鲁语或豪萨语的“ubuntu”一词，意思是“人性”、“我的存在是因为

大家的存在”，是非洲传统的一种价值观，类似华人社会的“仁爱”思想。Ubuntu 基于 Debian 发行版和 GNOME 桌面环境，目标在于为一般用户提供一个最新的、同时又相当稳定的主要由自由软件构建而成的操作系统。Ubuntu 具有庞大的社区力量，用户可以方便地从社区获得帮助。2013 年 1 月 3 日，Ubuntu 正式发布面向智能手机的移动操作系统。

1.3 Linux 的发展方向

Linux 的出现绝不仅仅是为用户带来了一种价廉物美的产品，使他们多了一种选择，在更深层次上的意义是，将给传统的软件版权制度、软件开发模式及企业经营模式带来革命性的影响。Linux 的开放源代码使用户拥有了知情权和参与权，更符合用户的希望和需求，将成为软件业未来的发展方向。

另一方面，Linux 是中国软件业摆脱目前低水平的二次开发，使之快速、健康发展的难得机遇。从以往经验教训来看，没有自主知识产权的操作系统就要受制于人，整个软件业就好像踩在棉花团上，不可能获得良好的自主发展的空间。Linux 给我们提供了这样一个大好环境。

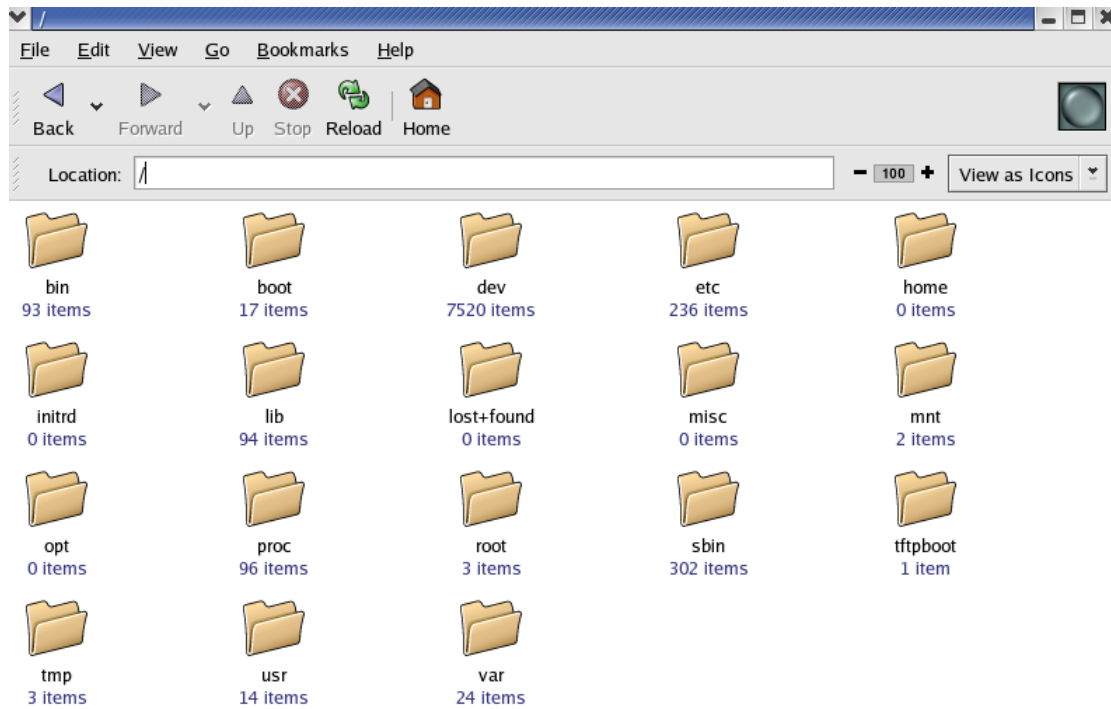
WWW.TI-MO-KO.COM

北南南北论坛: WWW.TI-MO-KO.COM

2 Linux 的目录和文件系统

2.1 Linux 目录

Linux 系统的文件目录如下图所示：



/

这就是根目录。在 Windows, DOS 或者其他类似的操作系统的分区里，每个分区都会有一个相应的根目录。但是 Linux 和其他 Unix 系统把所有的文件都放在一个目录树里面，/ 就是唯一的根目录。一般来讲，根目录下面很少保存什么文件，或者只有一个内核映像在这里。

/bin

系统启动时需要的执行文件（二进制），这些文件可以被普通用户使用。

/dev

设备文件目录。LINUX 下设备被当成文件，这样一来硬件被抽象化，便于读写、网络共享以及需要临时装载到文件系统中。正常情况下，设备会有一个独立的子目录。这些设备的内容会出现在独立的子目录下。LINUX 没有所谓的驱动符。

/home

该目录中包含系统上各个用户的主目录，子目录名称即为各用户名。

`/lib`

`lib` 是库 (library) 英文缩写。这个目录是用来存放系统动态连接共享库的。几乎所有的应用程序都会用到这个目录下的共享库。

`/opt`

某些第三方应用程序通常安装在这个目录。

`/root`

系统管理员 (也叫超级用户或根用户) 的 Home 目录。不要与根目录混淆，因为二者有相同的名字: `root directory`。

`/usr`

很多系统中，该目录是作为独立分区挂载上来的。该目录中主要存放不经常变化的数据。这是一个很庞大的目录，下面将会列举一些重要的子目录：

`/usr/X11R6`, `/usr/X11`, `/usr/Xfree86`

这里面保存着 X 系统所需要的文件，它的目录结构和 `/usr` 是一样的。

`/usr/bin`

二进制可执行文件存放的目录，这里放着绝大部分的应用程序。

`/usr/sbin`

这里放着绝大部分的系统程序。

`/usr/games`

游戏程序和相应的数据会放在这里。

`/usr/include`

这个目录保存着 C 和 C++ 的头文件。除非你是程序员，否则你可能不会感兴趣。

`/usr/lib`

启动时用不到的库文件都会被放在这里。

`/usr/info`

这里保存着 GNU Info 程序所需要的数据。

`/usr/man`

这里保存着 man 程序所需要的数据。

`/usr/src`

这里保存着源代码文件。Linux 内核的源代码都放在 `/usr/src/linux` 下了。

`/usr/doc`

这里保存着各种文档文件。这些文件可以帮助你了解 Linux，解决问题和提供一些技巧。

`/usr/local`

这里面保存着本地计算机所需要的文件。在用户进行远程访问的时候特别有意义。这个目录在有些 Linux 系统下就是一个单独的分区，存放一些这台机器所属的那个用户的文件。里面的结构和 `/usr` 是一样的。

`/usr/shared, /usr/share`

这里保存着各种共享文件，比如说某个应用程序所需要的声音文件。

`/boot`

该目录中存放系统的内核文件和引导装载程序文件。

`/etc`

这里保存着绝大部分的系统配置文件。相对来讲，单个用户的系统配置文件将会保存在这个用户自己的主目录里面。这里的文件会根据你所安装的软件而增减。

`/mnt`

这个目录下面放着一些用来 mount 其他设备的子目录，为某些设备提供的默认挂载点。

`/proc`

是一个虚拟的文件系统，该目录中的文件是内存中的映像。它们实际上并不存在磁盘上，也不占用任何空间。当查看这些文件时，实际上是在访问存在内存中的信息，这些信息用于访问系统。

`/sbin`

该目录中保存系统管理员或者 `root` 用户使用的命令文件。

`/selinux:`

`selinux` 的配置目录。Selinux 是安全增强 linux 的简称，是 linux 操作系

统中的一种强制访问控制实现机制，可以增强 linux 系统的安全性。

/tmp

用来存放一些临时文件的地方。

/var

该目录以及该目录下的子目录中通常保存经常变化的内容，如系统日志、打印队列、DNS 数据库文件等。

2.2 Linux 文件系统

文件系统是操作系统在硬盘或者分区上保存文件信息的方法和数据结构，也就是文件在硬盘或分区上的组织方式。

现在 Linux 系统使用比较多的是 ext3 文件系统，该文件系统是 ext2 的升级版本。在 Linux 系统中，存储数据的各种设备都属于块设备。对于磁盘设备，通常在 0 磁道第一个扇区上存放引导信息，称为主引导记录，该扇区不属于任何一个分区。每个分区包含许多数据块，可以认为是一系列块组的集合。

查看文件系统的命令：mount

2.2.1 Linux 与 DOS 文件系统的区别

Linux 文件系统是一个目录树的结构，它的根是根目录“/”，往下连接各个分支，例如，/bin、/usr、/sbin 等，如图 7-1 所示。

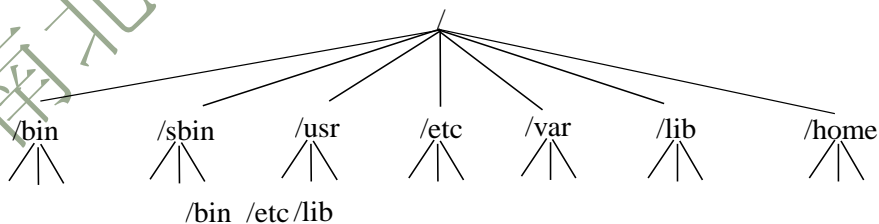


图 7-1 Linux 文件系统

DOS 也采用目录树的结构，但是与 Linux 的略有不同，如图 7-2 所示。

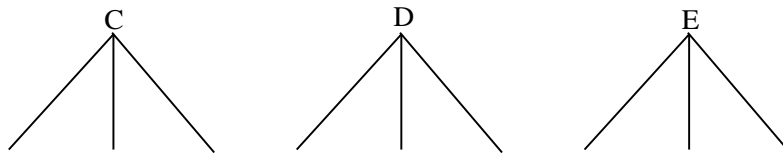


图 7-2 DOS 文件系统

DOS 以每个分区为树根，由于有多个分区，所以形成了多个树并列的情形。

2.2.2 Linux 文件系统结构

Linux 将文件存放到逐层继承排列的子目录中，这种结构的形状很像是一棵树，所以称为“树状结构” (tree structure)。这个树状结构是倒挂的，它的根被置于最顶部，从根向下延伸的是枝，每个枝向上只有一个连接，但向下可以有更多个枝。从这个角度看，也可以说有个“父—子”结构，即每个子目录都是另一个子目录的下级子目录。一个子目录可以有多个下级子目录，但它自身只能是一个父目录的子目录，如图 7-3 所示。

Linux 的文件结构从一个根目录 (即“/”目录) 开始向下分成多个子目录。根目录中的几个系统级子目录里容纳着形成 Linux 操作系统各种功能的文件和程序。标准的、原始的文件系统包括 /、/home、/usr、/var、/bin、/sbin、/etc、/dev、/lib。Linux 系统是以文件的目的为依据对文件进行分组的，即相同目的的命令都放在同一子目录中。系统子目录中的文件的作用是保证系统的正常运行。它们中的许多还有自己的下级子目录，并容纳完成 Linux 操作系统的特定功能的程序。

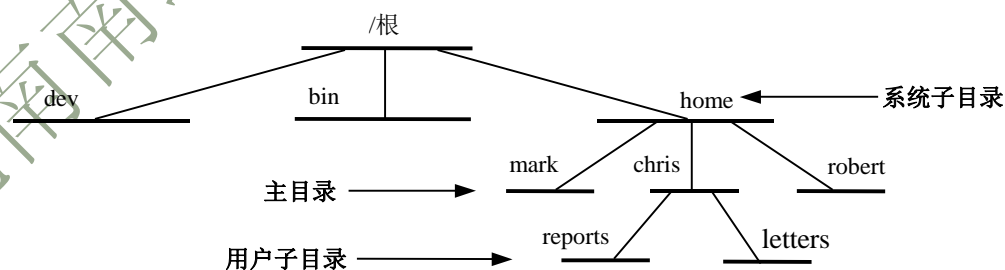


图 7-3 从根目录开始的 Linux 文件结构

根目录里有一个名为 home 的子目录，它容纳系统上全体用户的登录子目录 (home directory)，也叫做主目录、家目录等。

用户在创建子目录或者文件时分配的名字并不是它的全名。一个子目录的全名是它的路径名 (pathname)。子目录间继承性的嵌套关系构成了路径，而这些路径用来惟一地确定和引用一个子目录或者一个文件。在图 7-3 中，从根目录 “/” 到 home 子目录再到 robert 子目录是一个路径，路径名写成 /home/robert。系统标识某个子目录时实际用的名字总是从根目录开始，并由堆叠在该子目录上面的各级子目录组成。

路径名可以是绝对的也可以是相对的。

Linux 中各种文件都有相同的物理格式——即字节流 (byte stream)。“字节流”就是一个字节序列。这使 Linux 操作系统能够把文件的概念应用到系统中的每个数据元。子目录也被归入文件类，设备也是一样。将各种事情都作为文件使 Linux 操作系统能够更容易地组织和交换数据。文件中的数据可以直接发送给显示器等设备，因为设备与操作系统之间与普通文件一样使用的都是同一种字节流文件格式。

这个同样的文件格式也被用在这个操作系统的其他成分里。显示器和键盘这类设备的接口被设计为一个文件；子目录等其他成分就是字节流文件，只不过它们有一个特殊的内部结构罢了。子目录文件保存着关于子目录的信息，按照特殊的子目录格式组织起来。因为这些不同的组成都被视为文件，所以说它们构成了不同的文件类型。根据用户具体实现的 Linux 操作系统的不同，文件数量也会发生变化。但最常见的文件类型有 4 种：普通文件、目录文件、连接和设备文件。而 DOS 下常见的文件类型有两种：普通文件和目录文件。

2.2.3 Linux 文件权限

由于 Linux 是一个多用户的系统，它的文件当然不可避免地要根据用户来划分，并对文件进行保护。

用户分组名指出组权限分配给哪一个用户分组。如图 7-4 所示，mydata 文件的类型为普通文件；只有一个链接，表明该文件没有其他名字和其他链接；属主名为 chris，与用户的登录名相同；用户分组名为 weather，这个 weather 组里

可能还会有其他用户；文件的长度为 207 字节；上次的修改日期是 6 月 20 日上午 11 点 55 分；文件名为 mydata。

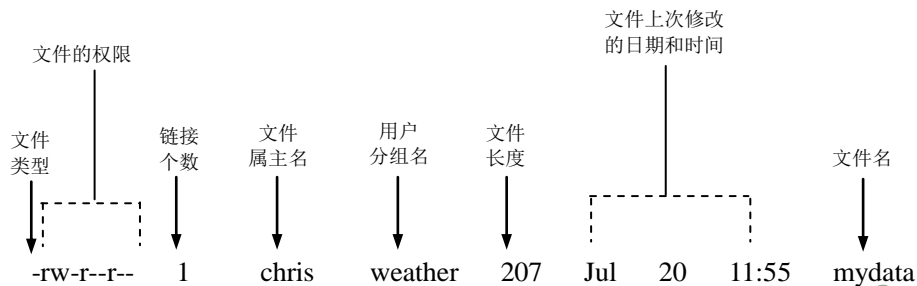
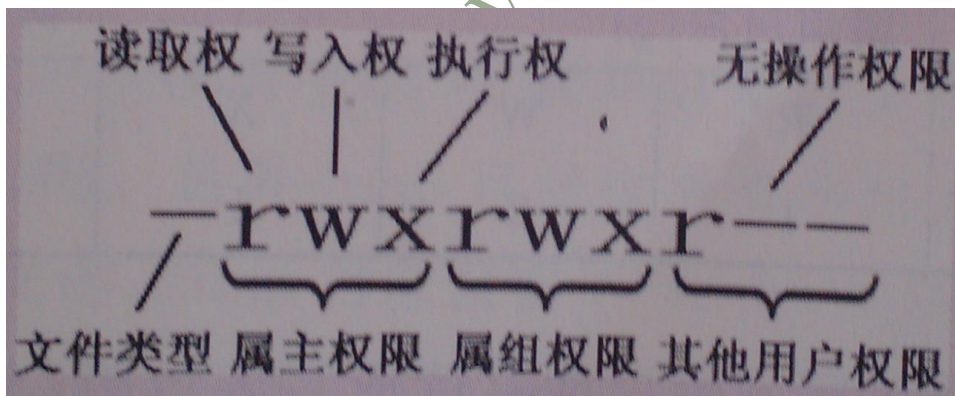


图 7-4 使用带-l 参数的 ls 命令列出的文件信息

如图 7-4 所示，左面第一组字符是文件 mydata 的访问权限清单。空白权限使用短划线字符“-”来表示的；读权限用字母“r”代表，写权限用字母“w”代表，而执行权限则用字母“x”代表。总共有 10 个位置，第一个字符指出文件的类型。由于子目录也可以被看成是一种文件类型，如果第一个字符是一个短划线字符“-”，就表示列出的是一个文件；如果是一个字母“d”，则表示显示的是关于某个子目录的信息。



随后的 9 个字符是根据不同的用户分类来排列的。第一组 3 个字符是文件属主对该文件的权限集；第二组 3 个字符是用户所在分组对该文件的权限集；最后一组 3 个字符是其他用户对该文件的权限集。在图 7-4 中，文件 mydata 的属主类用户被分配给读写权限 rw-；同组成员类用户被分配给读权限 r--；其他用户类用户也只被分配给读权限 r--。这意味着虽然同组成员和系统上的其他用户能够读取这个文件的内容，但只有文件的属主才能修改它。

使用“ls -l”命令目录详细信息，可以看到当前目录下所有文件的权限。使用“ls -l filename”命令显示文件的详细信息。

3. Linux 系统的登录和退出

Linux 系统的登录，在命令行方式下，登录界面如下：

```
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686

localhost login: root
Password: _
```

输入用户名，这里以 root 为用户名，回车，再输入登录密码，回车，即可进入 Linux 系统。

root 用户是系统的超级用户，可以执行所有的管理任务，可以不受限制地执行任何操作。基于安全的考虑，建议超级用户也为自己建立一个用来处理一般事物的普通帐户，只在必要的时候才使用 root 身份操作。

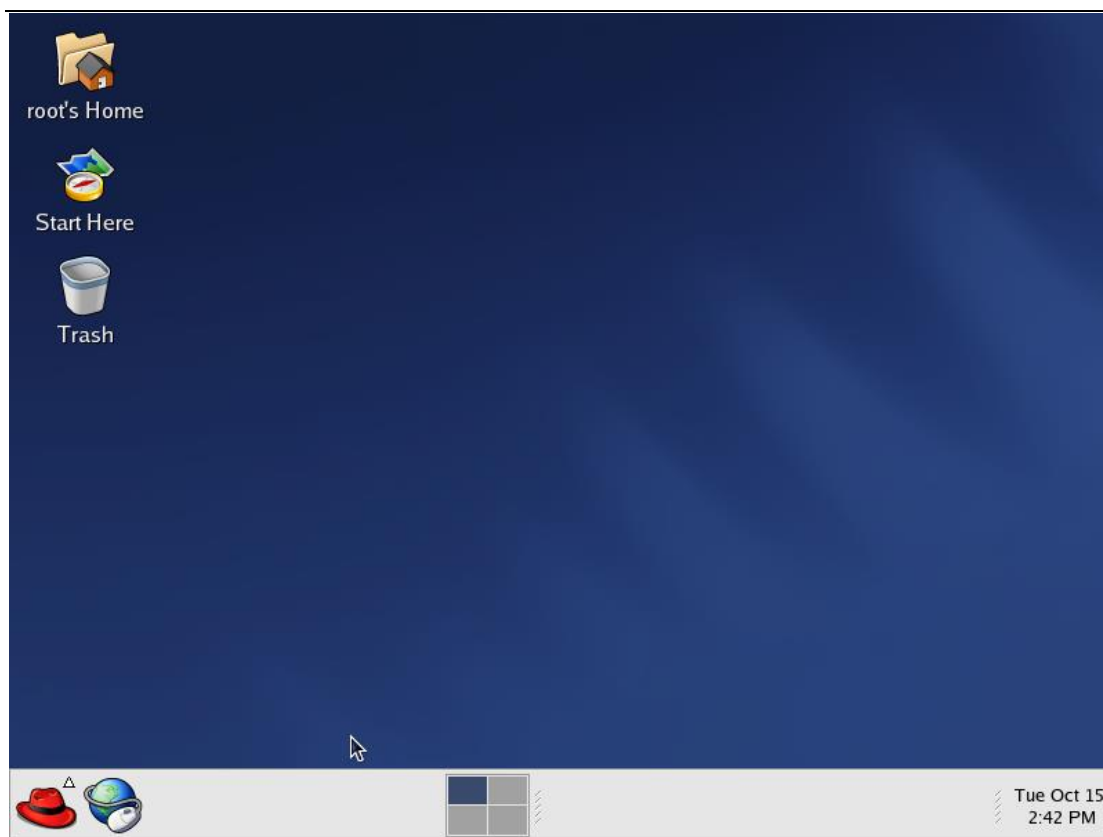
进入系统后的界面如下：

```
localhost login: root
Password:
Last login: Tue Oct 15 14:19:38 on tty1
[root@localhost root]# _
```

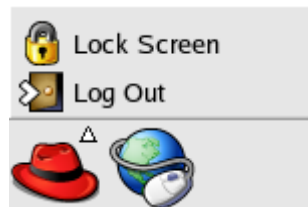
有时候进入 Linux 系统后默认的界面为命令行方式，如上图所示，如果想切换到图形化界面，在命令行输入命令 startx，回车即可。

```
localhost login: root
Password:
Last login: Tue Oct 15 14:19:38 on tty1
[root@localhost root]# startx_
```

进入 Linux 系统后的图形化界面如下：



Linux 系统的退出，可以点击左下角的  图标，弹出的菜单，



点击  Log Out，即可退出 Linux 系统。

4 学习 shell 工具

Shell 是 Linux 系统中的一个重要组成部分，是用户和系统交互作用的界面。Shell 是用户和 Linux 内核之间的接口程序，如果把 Linux 内核想象成一个球体的中心，Shell 就是围绕内核的外层，如图 所示。



Shell 的位置

Shell 还是一个命令语言解释器，拥有自己内建的 Shell 命令集，用户在提示符下输入的命令都由 Shell 先解释然后传给 Linux 核心。

在各种版本的 Linux 系统中，往往都提供多种 Shell 程序，用户可以根据自己的习惯进行选择。执行“chsh -l”命令会列出所有可用的 Shell 程序，如：

```
[admin@localhost ~]$ chsh -l
/bin/sh
/bin/bash
/sbin/nologin
/bin/ash
/bin/bsh
/bin/ksh
/usr/bin/ksh
/usr/bin/pdksh
/bin/tcsh
/bin/csh
```

用户登录到系统时，已经启动了一个默认的 Shell 程序。要想使用其他 Shell 的话，只需要命令行中输入要进入的 Shell 名称就可以。用户可以嵌套进入多个 Shell，连续使用“exit”命令会逐个推出各个 Shell，直至最后注销出来。在 Red Hat Enterprise Linux 4 种默认使用的 Shell 程序是 bash。

4.1 命令的基本用法

用户登录后使用命令时，实际上进入了某个 Shell 程序，它遵循一定的语法将输入的命令加以解释并传给系统。命令行中输入的第一个字必须是一个命令的名字，第二个字是命令的选项或参数，命令行中的每个字必须由空格隔开，格式如下：

\$ Command Option Arguments

选项 (Option) 是包括一个或多个字母的代码, 它前面有一个 “-” 符号, 该符号是必要的, Linux 用它来区别选项和参数。选项可用于改变命令执行的动作的类型。例如没有选项的 `ls` 命令, 可列出当前目录中所有文件但只列出各个文件的名称, 而不显示其他更多的信息。而加了 “-l” 选项的 `ls` 命令可以列出包含文件大小、权限、修改日期等更多信息的文件和目录列表。

当需要使用一个命令的多个选项时, 可以简化输入。例如要使用命令 “`ls -l -a`”, 可以简化写为 “`ls -la`”。如果某个选项由多个字符组成, 则在应用该选项时必须使用 “-” 符号, 以便与间歇的多个选项相区别。例如要查看 `ls` 命令的简单帮助信息, 可以运行如下命令:

```
$ ls -help
```

大多数命令都可以接纳参数, 参数通常是命令地操作对象。例如: “`ls -l /tmp`” 将显示 `/tmp` 目录下的所有文件及其信息。其结果与先进入 `/tmp` 目录再执行 “`ls -l`” 的结果一样。

有些命令, 如本例中的 `ls` 可以不带参数, 而有一些命令可能需要一些最小数目的参数。例如复制文件时, `cp` 命令至少需要两个参数, 即源文件和复制后的目标文件。例如:

```
$ cp oldfile newfile
```

4.2 命令历史

用户执行的命令会保存在一个命令缓存区中, 成为命令历史。默认情况下, Red Hat Enterprise Linux 4 可以存储 1 0 0 0 个历史命令。用户可以查看自己的命令历史, 并且可以根据需要重新调用历史命令。下面这些案件可以帮助用户查看和修改命令行。

“↑”: 显示上一个命令;

“↓”: 显示下一个命令;

“←”: 光标左移;

“→” 光标右移;

在 Shell 执行命令时, 如果一条命令花费了很长的时间来运行, 或者在屏幕上产生了大量的输出, 或者用户进入一个陌生的交互模式, 则可以在键盘上按 `Ctrl+C` 发出终端信号来终止当前程序的运行。

4.3 命令自动补齐

如果用户输入了文件名、命令或路径名的一部分, 然后按 `Tab` 键, `bash` 要么会把文件或路径名的剩余部分补全, 要么会给用户一个蜂鸣声。这时, 只需要再按一次 `Tab` 键就可以获取与已输入的部分匹配的所有文件或路径名的列表。

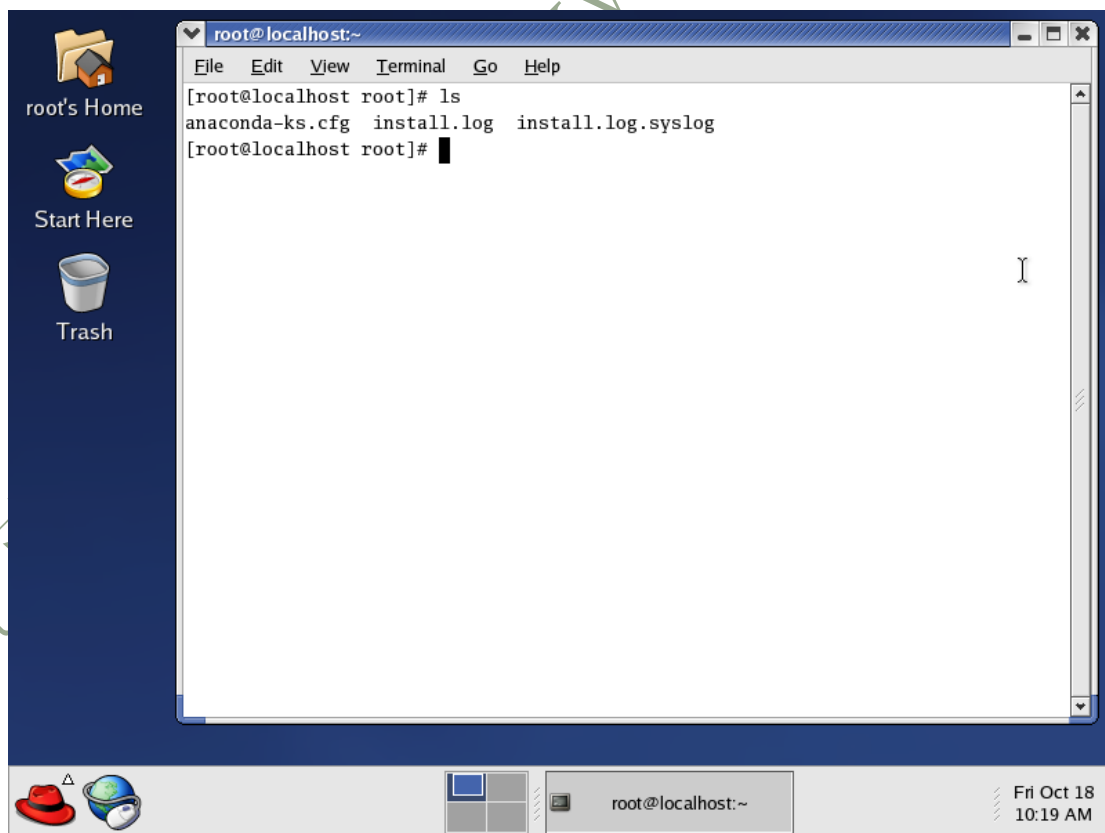
5 Linux 常见命令

5.1 Linux 命令基础

Linux 终端窗口：

- 字符终端为用户提供了一个标准的命令行接口，在字符终端窗口中，会显示一个 Shell 提示符，通常为\$。
- 用户可以在提示符后输入带有选项和参数的字符命令，并能够在终端窗口中看到命令的运行结果，此后，将会出现一个新的提示符，标志着新命令行的开始。
- 字符终端窗口中出现的 Shell 提示符因用户不同而有所差异，普通用户的命令提示符为“\$”，超级管理员用户的命令提示符为“#”。
- 在 Linux 系统中命令区分大小写！

Linux 终端窗口如下图所示：

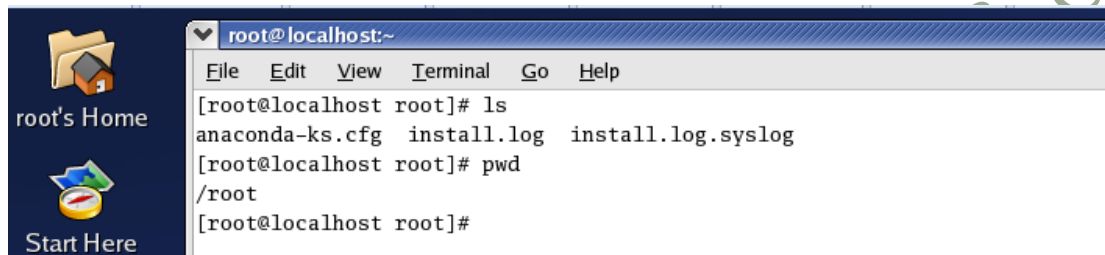


5.2 Linux 文件目录类命令

5.2.1 浏览目录类命令

pwd 命令:

用于显示用户当前所在的目录。如果用户不知道自己当前所处的目录,就可以使用这个命令获得当前所在目录。



```
root@localhost:~
File Edit View Terminal Go Help
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost root]# pwd
/root
[root@localhost root]#
```

ls 命令:

ls 命令跟 dos 下的 dir 命令一样,用于显示当前目录的内容。

如果想取得详细的信息,可用 `ls -l` 命令,这样就可以显示目录内容的详细信息。

如果目录下的文件太多,用一屏显示不了,可以用 `ls -l | more` 分屏显示。

```
[root@localhost opt]# ls
[root@localhost opt]# ls / -l
total 201
drwxr-xr-x  2 root  root    4096 Oct  1  2005 bin
drwxr-xr-x  4 root  root    1024 Oct  1  2005 boot
drwxr-xr-x 20 root  root  118784 Oct 18 10:03 dev
drwxr-xr-x 76 root  root    8192 Oct 18 10:03 etc
drwxr-xr-x  2 root  root    4096 Jan 25  2003 home
drwxr-xr-x  2 root  root    4096 Jan 25  2003 initrd
drwxr-xr-x  9 root  root    4096 Oct  1  2005 lib
drwx----- 2 root  root   16384 Oct  1  2005 lost+found
drwxr-xr-x  2 root  root    4096 Jan 28  2003 misc
drwxr-xr-x  4 root  root    4096 Oct 15 14:16 mnt
drwxr-xr-x  2 root  root    4096 Jan 25  2003 opt
```

cd 命令:

用户在登录系统后,会处于用户的家目录中,该目录一般以/home 开始,后跟用户名,这个目录就是用户的初始登录目录(root 用户的家目录为/root)。

cd 命令用来在不同的目录中进行切换。如果用户想切换到其他的目录中,就可以使用 cd 命令,后跟想要切换的目录名。

在 Linux 系统中，用 “.” 代表当前目录；用 “..” 代表当前目录的父目录；用 “~” 代表用户的个人家目录。

```
[root@localhost home]# ls /
bin  dev  home  lib      misc  opt   root  tftpboot  usr
boot etc  initrd lost+found mnt   proc  sbin  tmp       var
[root@localhost home]# cd /opt
[root@localhost opt]# ls
[root@localhost opt]#
```

su 命令：

su 命令是最基本的命令之一，常用于不同用户间切换。

su 命令的常见用法是变成根用户或超级用户。如果发出不带用户名的 su 命令，则系统提示输入根口令，输入之后则可切换为根用户。如果登录为根用户，则可以用 su 命令成为系统上任何用户而不需要口令。

```
[root@localhost root]# su test
[test@localhost root]$
```

5.2.2 目录操作类命令

mkdir 命令：

mkdir 命令用于创建一个目录。

该命令的语法为：`mkdir [参数] 目录名`

常用参数 `-p`：如果父目录不存在，则同时创建该目录及该目录的父目录。

```
root@localhost:~
File Edit View Terminal Go Help
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost root]# mkdir test
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog  test
```

rmdir 命令：

rmdir 命令用于删除空目录。

该命令的语法为：`rmdir [参数] 目录名`

常用参数-p: 在删除目录时, 一起删除父目录, 但父母录中必须没有其他目录及文件。

```
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog  test
[root@localhost root]# rmdir test
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost root]#
```

5.2.3 文件操作类命令

cp 命令:

cp 命令主要用于文件或目录的复制。

该命令的语法为: cp [参数] 源文件 目标文件

cp 命令的常用参数选项有:

-f: 如果目标文件或目录存在, 先删除它们再进行复制 (即覆盖), 并且不提示用户。

-i: 如果目标文件或目录存在, 提示是否覆盖已有的文件。

-R: 递归复制目录, 即包含目录下的各级子目录。

```
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost root]# cp /boot/boot.b /root
[root@localhost root]# ls
anaconda-ks.cfg  boot.b  install.log  install.log.syslog
[root@localhost root]#
```

mv 命令:

mv 命令主要用于文件或目录的移动或改名。

该命令的语法为:

mv [参数] 源文件或目录 目标文件或目录

mv 命令的常用参数选项有:

-i: 如果目标文件或目录存在时, 提示是否覆盖目标文件或目录。

-f: 无论目标文件或目录是否存在, 直接覆盖目标文件或目录, 不提示。

```
[root@localhost root]# ls
anaconda-ks.cfg  boot.b  install.log  install.log.syslog
[root@localhost root]# mv boot.b /home
[root@localhost root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@localhost root]# ls /home
boot.b  music
```

rm 命令:

rm 命令主要用于文件或目录的删除。

命令的语法为: rm [参数] 文件名或目录名

rm 命令的常用参数选项有:

- i: 删除文件或目录时提示用户。
- f: 删除文件或目录时不提示用户。
- R: 递归删除目录, 即包含目录下的文件和各级子目录。

```
[root@localhost home]# ls
boot.b  music
[root@localhost home]# rm boot.b
rm: remove regular file `boot.b'? y
[root@localhost home]# ls
music
```

diff 命令:

diff 命令用于比较两个文件内容的不同。

该命令的语法为: diff [参数] 源文件 目标文件。

```
[root@localhost root]# cat b.txt
aaa
bbb
ccc
[root@localhost root]# cat c.txt
aaa
ddd
eee
[root@localhost root]# diff b.txt c.txt
2,3c2,3
< bbb
< ccc
---
> ddd
> eee
[root@localhost root]#
```

tar 命令:

用于文件打包的命令，tar 命令可以把一系列的文件归档到一个大文件中，也可以把档案文件解开以恢复数据。

命令语法为：tar [参数] 档案文件 文件列表

tar 命令的常用参数选项有：

- c: 生成档案文件。
- v: 列出归档解档的详细过程。
- f: 指定档案文件名称。
- r: 将文件追加到档案文件末尾。
- z: 以 gzip 格式压缩或解压缩文件。
- j: 以 bzip2 格式压缩或解压缩文件。
- d: 比较档案与当前目录中的文件。
- x: 解开档案文件。

```
[root@localhost root]# tar cvf root.tar.gz *.*
anaconda-ks.cfg
a.txt
b.txt
c.txt
install.log
install.log.syslog
[root@localhost root]# ls
anaconda-ks.cfg  b.txt  install.log          root.tar.gz
a.txt           c.txt  install.log.syslog
[root@localhost root]#
```

grep 命令:

grep 命令用于查找文件中包含有指定字符串的行。

该命令的语法为:

grep [参数] 要查找的字符串 文件名

grep 命令的常用参数选项有:

- v: 列出不匹配的行。
- c: 对匹配的行计数。
- l: 只显示包含匹配模式的文件名。
- h: 抑制包含匹配模式的文件名的显示。
- n: 每个匹配行只按照相对的行号显示。
- i: 对匹配模式不区分大小写。

```
[root@localhost root]# cat c.txt
aaa
ddd
eee
[root@localhost root]# grep dd /root/c.txt
ddd
[root@localhost root]#
```

5.2.4 文本文件显示命令

cat 命令:

cat 命令主要用于滚屏显示文件内容或是将多个文件合并成一个文件;

滚屏显示文件的语法为: cat [参数] 文件名;

合并文件的语法为：cat file1 file2 >file3 或者 cat file1 file2>>file3;

```
[root@localhost root]# cat a.txt
aaa
bbb
ccc
```

more 命令：

使用 cat 命令时，如果文件太长，用户只能看到文件的最后一部分。这时可以使用 more 命令，一页一页的分屏显示文件的内容。

按【Enter】键可以向下移动一行，按【space】键可以向下移动一页；按【q】键可以退出 more 命令。

该命令的语法为：more [参数] 文件名。

```
[root@localhost root]# cat install.log |more
Installing 725 packages

Installing glibc-common-2.3.2-11.9.
Installing hwdata-0.75-1.
Installing redhat-logos-1.1.12-1.
```

less 命令：

less 命令是 more 命令的改进版，比 more 命令的功能强大。more 命令只能向下翻页，而 less 命令可以向下、向上翻页，甚至可以前后左右的移动。

执行 less 命令后，进入了 less 状态，按【Enter】键可以向下移动一行，按【space】键可以向下移动一页；按【b】键可以向上移动一页；也可以用光标键向前、后、左、右移动；按【q】键可以退出 less 命令。

head 命令：

head 命令用于显示文件的开头部分，默认情况下只显示文件的前 10 行内容。

该命令的语法为：head [参数] 文件名。

head 命令的常用参数选项有：

-n num: 显示指定文件的前 num 行。

-c num: 显示指定文件的前 num 个字符。

tail 命令:

tail 命令用于显示文件的末尾部分, 默认情况下只显示文件的末尾 10 行内容。

命令的语法为: tail [参数] 文件名

tail 命令的常用参数选项有:

- n num: 显示指定文件的末尾 num 行。
- c num: 显示指定文件的末尾 num 个字符。
- +num: 从第 num 行开始显示指定文件的内容。

5.2.5 添加新用户和修改密码命令

useradd 命令:

useradd 命令用于添加新的用户账号。

命令的语法如下: useradd 选项 用户名

其中各选项含义如下:

- c comment 指定一段注释性描述。
- d 目录 指定用户主目录, 如果此目录不存在, 则同时使用-m 选项, 可以创建主目录。
- g 用户组 指定用户所属的用户组。
- G 用户组, 用户组 指定用户所属的附加组。
- s Shell/文件 指定用户的登录 Shell。
- u 用户号 指定用户的用户号, 如果同时有-o 选项, 则可以重复使用其他用户的标识号。

```
[root@localhost root]# useradd test01
[root@localhost root]# su test01
[test01@localhost root]$
```

userdel 命令

删除用户账户。

命令语法如下: userdel 选项 用户名

例如 userdel user2, 用于删除用户 user2。

删除用户 user3, 同时删除他的工作目录

```
userdel -r user3
```

passwd 命令:

passwd 命令用于修改登录密码。如果是以 root 身份登录, 修改 root 密码。

只要输入 passwd 就会出现:

```
New password:
```

```
Retype new password:
```

按提示输入密码确认即可。

如果想更改其他用户密码, 或者给用户设置密码, 只要输入 passwd username 即可。如: passwd lsyer New password:

```
Retype new password:
```

可以输入

```
passwd -help
```

来查看该命令的使用方法。

5.2.6 退出、关闭系统命令

shutdown 命令:

用于在指定时间关闭系统。

```
[root@RHEL4 ~]# shutdown
Usage:  shutdown [-akrhfncl] [-t secs] time [warning message]
-a:    use /etc/shutdown.allow
-k:    don't really shutdown, only warn.
-r:    reboot after shutdown.
-h:    halt after shutdown.
-f:    do a 'fast' reboot (skip fsck).
-F:    Force fsck on reboot.
-n:    do not go through "init" but go down real fast.
-c:    cancel a running shutdown.
-t secs: delay between warning and kill signal.
** the "time" argument is mandatory! (try "now") **

[root@RHEL4 ~]# _
```

halt 命令:

表示立即停止系统，但该命令不自动关闭电源，需要人工关闭电源。

reboot 命令：

用于重新启动系统，相当于“shutdown -r now”。

poweroff 命令：

用于立即停止系统，并关闭电源，相当于“shutdown -h now”。

date 命令：

可以用来查看系统当前的日期和时间。

```
[root@localhost root]# date
Fri Oct 18 11:48:27 CST 2013
```

5.2.7 挂载、卸载U盘命令

Linux 下挂载U盘：

首先在/mnt 目录下建立一个挂装 USB 存储器的目录：

```
mkdir /mnt/udisk
```

然后再运行装载设备命令，将 USB 设备挂装到/mnt/usb 目录下：文件系统为 FAT32

```
mount -t vfat /dev/sda1 /mnt/udisk
```

或者如果为其它文件格式，如 FAT16

```
mount -t msdos /dev/sda1 /mnt/udisk
```

运行如下命令即可查看 USB 存储器中的文件信息：

```
ls -l /mnt/udisk
```

Linux 下卸载U盘：

卸载U盘必须在不同的目录下进行卸载，也就是说不能在/mnt/udisk 目录下进行卸载。。

卸载命令如下

RT Embedded <http://www.kontron.com>

```
cd /root
```

```
umount /mnt/udisk
```

如果显示 device is busy, 可以用如下命令进行卸载:

```
cd /root
```

```
fuser -km /mnt/udisk umount /mnt/udisk
```

北南南北论坛: www.vxworks6.com

用 vi 打开一个文件就直接进入指令模式了（这时默认的模式）。在这个模式中，你可以使用上下左右键来移动光标，可以删除字符、字或删除行，也可以移动复制、粘贴你的文件数据。

6 Vi 编辑器的用法

vi 编辑器是任何 Unix 及 Linux 系统下标准的编辑器，是一款很强大的文本编辑器，这里只是简单地介绍一下他的用法和一小部分指令。由于对 Unix 及 Linux 系统的任何版本，vi 编辑器是完全相同的，因此您能够在其他任何介绍 vi 的地方进一步了解他。Vi 也是 Linux 中最基本的文本编辑器，学会他后，您将在 Linux 的世界里畅行无阻。

6.1 vi 的基本概念

基本上 vi 能够分为三种状态，分别是指令模式（command mode）、插入模式（Insert mode）和末行模式（last line mode），各模式的功能区分如下：

- 1) 指令模式（也称为一般模式，命令模式）
- 2) 输入模式（也称为编辑模式）

在指令模式中可以进行删除、复制、粘贴等的操作，但是却无法编辑文件内容的。要等你按下“i, I, o, O, a, A”等任何一个字母之后才会进入编辑模式。通常在 Linux 中，按下这些按钮时，在节目的左下方会出现 INSERT 或 R E P L A C E 的字压根，此时才可以进行编辑。而如果要回到指令模式，则必须要按下 ESC 这个按键才可退出编辑模式。

- 3) 末行模式（也称为底行模式）

在指令模式下，输入“:”，就可以将光标移动到最下面那一行。在这个模式不可以将文档保存或退出 vi，也能够配置编辑环境，如寻找字符串、列出行号……等。

但是一般我们在使用时把 vi 简化成两个模式，就是将末行模式（last line mode）也算入命令行模式 command mode）。

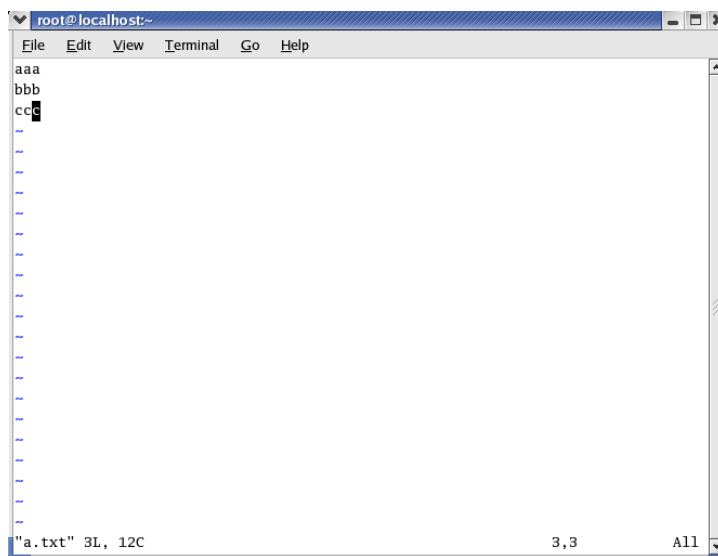
6.2 vi 的基本操作

a) 进入 vi

在系统提示符号输入 vi 及文档名称后，就进入 vi 全屏幕编辑画面：

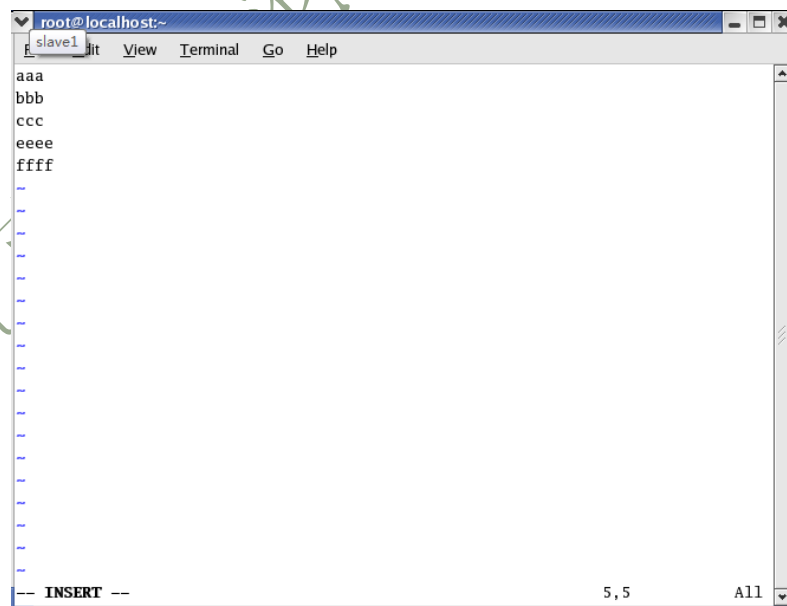
```
$ vi myfile
```

```
[root@localhost root]# vi a.txt  
[root@localhost root]#
```



b) 转换至插入模式 (Insert mode) 编辑文档

在「指令模式 (command mode)」下按一下字母「i」就能够进入「输入模式 (Insert mode)」，这时候您就能够开始输入文字了。



6.3 常用的编辑命令

Vi 提供许多移动光标的方式，下图显示了常用的光标移动命令。

指令	说明
h	向左移动光标
j	向下移动光标
k	向上移动光标
l	向右移动光标
0	移动光标到所在行的最前面(行首)
\$	移动光标到所在行的最后面(行尾)
Ctrl+f	向下翻一页
Ctrl+b	向上翻一页
Ctrl+d	向下翻半页
Ctrl+u	向上翻半页
H	移动到当前窗口的第一行
M	移动到当前窗口的中间行
L	移动到当前窗口的最末行
B	移动到上一个单词的第一个字符
W	移动到下一个单词的第一个字符
E	移动到下一个单词的最后一个字符
~	移动到光标所在行的第一个非空白字符
n-	减号移动到上一行的第一个非空白字符,前面加上数字可以指定向上移动 n 行
n+	加号移动到下一行的第一个非空白字符,前面加上数字可以指定向下移动 n 行
nG	数字 n 加上大写 G,移动到第 n 行

行编辑指令非常灵活，基本上是由指令与操作范围构成。例如指令 d 与范围 w 所组成的 dw 命令表示从光标所在位置开始删除，直到下个行的第一个字母。

指令列表如下：

- d: 删除(delete)。
- y: 复制(yank)。
- p: 放置(put)。
- c: 修改(change)。

操作范围可以是下列几个。

- e: 光标所在位置到该词的最后一个字符。
- w: 光标所在位置到下个词的第一个字符。
- b: 光标所在位置到上个词的第一个字符。
- \$: 光标所在位置到该行的最后一个字符。
- O: 光标所在位置到该行的第一个字母。
-): 光标所在位置到下个句子的第一个字母。
- (: 光标所在位置到该句子的第一个字母。
- }: 光标所在位置到该段落的最后一个字母。
- {: 光标所在位置到该段落的第一个字母。

6.4 多文件操作

在 vi 编辑器中，可以很方便地操作多个文本文件。例如要在某个文件中复制另一个文件的所有内容到当前光标处，可以使用末行命令“r filename”。

也可以使用 vi 编辑器同时打开多个文件，例如：

```
$vi file1 file2
```

这时两个文件都已经被 vi 编辑器打开，只不过在当前窗口上只能显示一个文件的内容。可以使用末行模式命令“:next”和“:previous”在两个文件之间切换。