

# 第3部分

## 文件管理与常用命令

### (续1)

# Linux 目录结构

Linux和Windows操作系统中的文件系统不同。先对Windows和Linux上面文件系统的一些概念进行区分，介绍一些Linux文件系统相关的原理，再较为详细地介绍了Linux系统的目录结构。

## 1、Window上面的文件系统

Windows系统中一切东西都是存放在硬盘上的。启动系统后，先**硬盘**，再**硬盘上的分区**和**每个分区所对应的文件系统**，最后是存放在某个分区特定的文件系统中的**文件**。

也就是说，Windows是通过“**某个硬盘-硬盘上的某个分区-分区上的特定文件系统-特定文件系统中的文件**”这样的顺序来访问到一个文件的。

## 2、Linux上面的文件系统

在Linux系统中，一切东西都是存放在一个唯一的“**虚拟文件系统**”中的，这个“虚拟文件系统”是树状的结构以一个**根目录**开始。

启动系统后，先有这个虚拟文件系统，再硬盘，再把某个硬盘的某个分区做为这个虚拟文件系统的一部分（**用某个子目录表示这个分区**），再通过这个虚拟文件系统中对应硬盘分区的那个子目录确定硬盘分区中的特定文件系统，最后的文件就存放在这个特定的文件系统中，也就是说，Linux系统是通过“**虚拟文件系统-硬盘-硬盘上的分区-分区上的特定文件系统-特定文件系统中的文件**”这样的顺序来访问一个文件的。

- 这里的“虚拟文件系统”，实质就是一棵**目录树**，最开始的目录叫做**根目录**，根目录中又有每一级子目录，或者文件，子目录又有子子目录和文件，其中**每个子目录都特定的功能**。
- 如果想要使用哪个硬盘的某个分区，就把那个分区“挂载”到某个子目录之下，这样硬盘中的分区，文件系统，目录等内容就呈现到了那个子目录里面。也就是说，在Linux中，使用硬盘中的数据，实际是先把硬盘“挂载”到某个子目录下，然后**通过那个子目录来访问硬盘**的。硬盘中原的数据组织结构都反映到了子目录上，成为了整个“虚拟文件系统”根目录下的一棵子树。

### 3、Linux文件系统在操作系统中的位置

在Linux系统中，主要两个方面：“**虚拟文件系统**”中的**文件**和Linux操作系统**内核本身**。逻辑上可以认为前者属于上层，后者在下层，前者基于后者，后者依赖前者而存在。

Linux把除了它本身（Linux操作系统内核）以外的一切事物都看作是在“虚拟文件系统”中的**文件**了。无论是键盘，鼠标，数据，程序，CPU，内存，网卡……无论是硬件、软件、数据还是内存中的东西，都可以在虚拟文件系统中的**相应子目录**对他们进行访问和操作，操作统一。

而实现这些管理的幕后就是Linux操作系统内核本身：  
启动Linux系统的时候，首先把操作系统内核加载到内存中，内核本身提供了文件管理，设备管理，内存管理，CPU进程调度管理，网络管理等功能，等内核运行起来之后，就在内存中建立起相应的“虚拟文件系统”，最后就是内核利用它提供的那些功能，来管理虚拟文件系统中的硬件软件等各种资源了。

- 我们一般都是使用**路径**来访问一个文件的。表示文件的路径由“**文件所在的目录+各级目录的分隔符+文件名**”各部分组成，这无论在Windows还是Linux上面，都是一样的，所不同的是，Windows下面目录分隔符是“\”，Linux下面是“/”。

- 例如：

- Window系统上的文件：

**D:\Program Files\PPStream\PPStream.exe**

- Linux系统上的文件：

**/usr/bin/screen**



- Linux系统的目录结构是**分层的树形结构**，都是挂载在根文件系统“/”下。
- 在Linux文件系统中的每一个子目录都有特定的**目的和用途**。
- 一般都是根据FHS标准（目录配置标准）定义一个正式的文件系统结构的，这个标准重点在于规范**每个特定的子目录下应该要放置什么类型的数据**。

### 3.3.2 Linux 目录结构

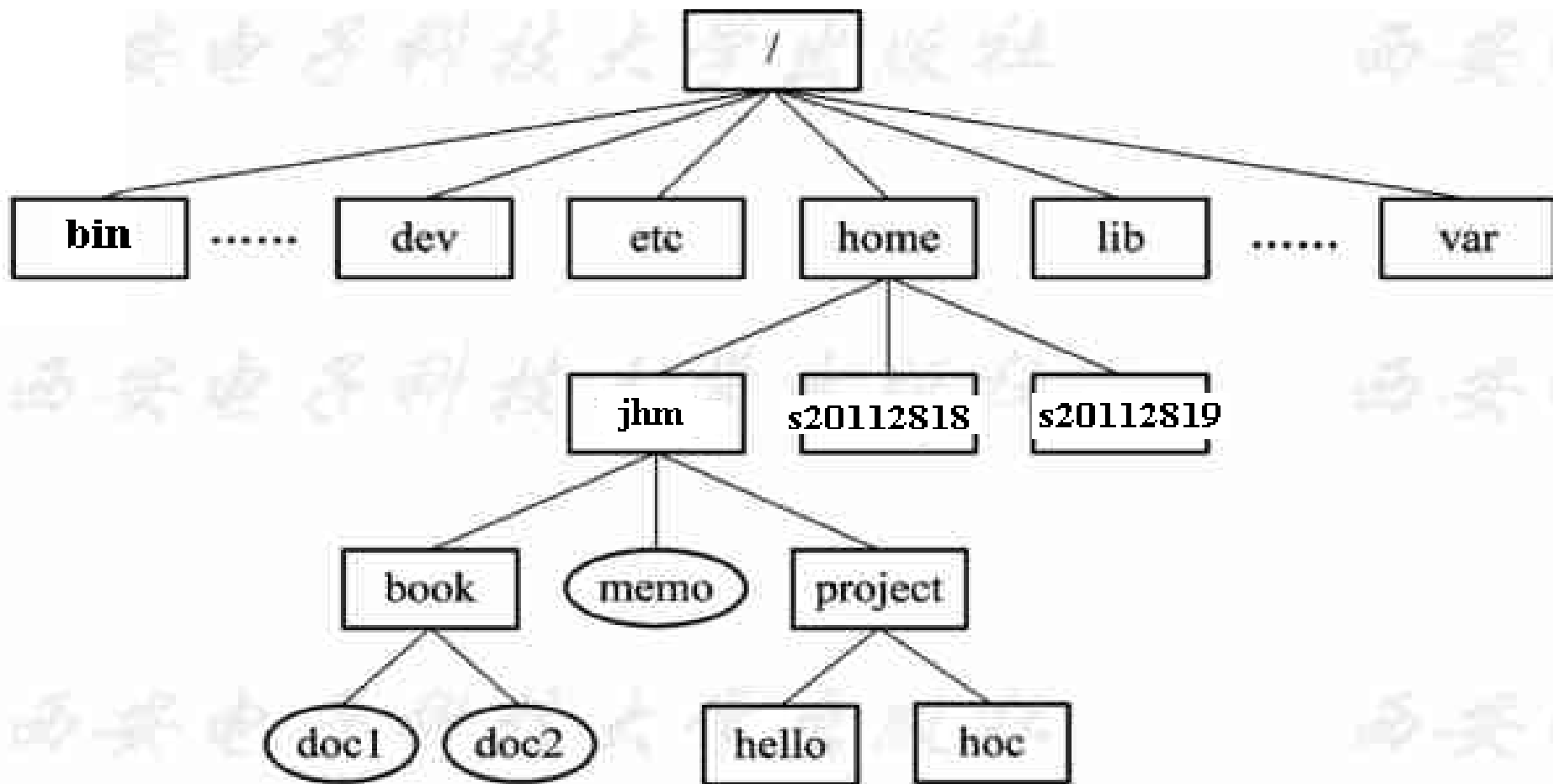


图3-1

# 1. Linux 目录结构

## (1) /

**/目录**也称为**根目录**，位于linux文件系统目录结构的顶层。是很多系统中的唯一分区。如果还有其他分区，必须挂在到**/目录**下某个位置。整个目录结构呈树型结构，因此也称为**目录树**。

Linux文件系统的入口，也是处于最高一级的目录；

2) bin bin目录为**命令文件**目录，也称为**二进制目录**。包含了供系统管理员及普通用户使用的重要的**linux命令**和**二进制（可执行）文件**，包含shell解释器等，**该目录不能包含子目录**。

**3) boot目录** 该目录中存放系统的**内核**文件和**引导装载程序**文件。

这个目录，包括了在**引导过程**中所必需的文件。在最开始的启动阶段，通过**引导程序**将**内核**加载到**内存**，完成内核的启动。

然后内核自己创建好虚拟文件系统，并且从虚拟文件系统的其他子目录中（例如 **/sbin** 和 **/etc**）加载**需要在开机启动**的其他程序或者服务或者特定的动作。

**4) dev目录**，也称设备（**device**）文件目录，存放连接到计算机上的**设备**（终端、磁盘驱动器、光驱及网卡等）的**对应文件**，包括字符设备和块设备等。

在Linux中设备都是以文件形式出现，这里的设备可以是硬盘，键盘，鼠标，网卡，终端等设备，通过访问这些文件可以访问到相应的设备。一般来说，想要Linux系统支持某个设备，只要有：**相应的硬件设备**，**支持硬件的驱动模块**，以及**相应的设备文件**。

**5) etc目录**存放系统的大部分**配置**文件和**子目录**。**X Window**系统的文件保存在**/etc/X11**子目录中，与网络有关的配置文件保存在**/etc/sysconfig**子目录中。

系统和程序一般都可以通过**修改相应的配置文件**，来进行配置。例如，要配置系统开机的时候启动那些程序，配置某个程序启动的时候显示什么样的风格等等。

该目录下的文件由系统管理员来使用，普通用户对大部分文件有只读权限。

**6)home目录**，普通用户的**家目录**。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**。

该目录中保存了绝大多数的用户文件(用户自己的配置文件，定制文件，文档，数据等)，**root**用户除外。

由于这个目录包含了用户实际的数据，通常系统管理员为这个目录单独挂载一个独立的磁盘分区，这样这个目录的文件系统格式就可能和其他目录不一样了



**7) lib目录**是根目录下的所程序的**共享库目录**。

目录下存放了各种编程语言库。典型的**linux**系统包含了**C、C++和FORTRAN**语言的库文件。

目录**/lib/modules**包含了可加载的内核模块。

**/lib**目录存放了所有重要的库文件，其他的库文件则大部分存放在**/usr/lib**目录下。

**8) lost+found 目录**，在EXT2或EXT3文件系统中，当系统意外崩溃或机器意外关机，产生的一些文件碎片放在这个目录。在系统启动的过程中**fsck**工具会检查这里，并修复已经损坏的文件系统。

有时系统发生问题，有很多的文件被移到这个目录中，可能会用手工的方法来修复，或者移动文件到原来的位置上。

**9) opt目录** **opt**表示的是可选择的意思，有些软件包也会被安装在这里，某些第三方应用程序通常安装在这个目录。比如在**Fedora Core 5.0**中，**OpenOffice**就是安装在这里。一些我们自己编译的软件包，就可以安装在这个目录中；通过源码包安装的软件，可以把它们的安装路径设置成**/opt**这样来安装。

## 10) **root**目录为系统管理员的主目录。

系统管理员(就是**root**用户或超级用户)的主目录比较特殊，不存放在**/home**中，而是直接放在**/root**目录下了。

**11) usr目录**是linux文件系统中最大的目录之一，很多系统中，该目录是为独立分区挂载的。该目录中主要存放**不经常变化的数据**，以及**系统下**安装的应用程序目录。

**12) mnt目录**主要用来临时挂载文件系统，为某些设备提供默认挂载点，如**floppy**，**cdrom**。这样当挂载了一个设备如光驱时，就可以通过访问目录**/mnt/cdrom**下的文件来访问相应的光驱上的文件了。

**13) proc目录**是一个虚拟的文件系统，该目录中的文件是内存中的映像。可以通过查看该目录中的文件获取有关系统硬件运行的详细信息，

例如使用**more**或者**less**命令查看 **/proc/interrupts** 文件以获取硬件中断（**IRQ**）信息，查看 **/proc/cpuinfo** 文件以获取**CPU**的型号、主频等信息。

**14) sbin目录**下保存系统管理、**root**用户的命令文件。

**15) tmp 目录**存放了临时文件，一些命令和应用程序会用的到这个目录。该目录下的所有文件会被定时删除，以避免临时文件占满整个磁盘。

**16) media 目录**挂载的媒体设备目录。

挂载的媒体设备目录，一般外部设备挂载到这里。比如插入一个U盘，**Linux**自动在这个目录下建立一个**disk**目录，然后把U盘挂载到这个**disk**目录上，通过访问这个**disk**来访问U盘。

## 2. 根目录与当前目录

根目录是一个特殊目录，用“/”表示。它是整个文件系统的唯一的根，系统中的所有文件都在它及其下属的子目录中。

用户在系统中工作时，始终处在某个目录之中，此目录称作当前目录。用户可以通过改变当前目录来变换自己在文件系统中的位置。当前目录用“.”表示。当前目录的父目录用“..”表示。每个目录(包括空目录)中都至少有“..”和“.”这两个隐含文件。但根目录中的“..”和“.”都是指其自身。



### 3. 路径

路径是**文件在目录树中所处的位置**。在对文件进行操作时，除需指明文件名外，还须指明该文件的路径。路径的表达方法是将若干个目录名连结起来，中间用斜杠“/”分开。

根据起点的不同，路径分为**绝对路径**和**相对路径**两种。

绝对路径是从根目录沿目录树到文件所在目录的路径。

绝对路径都是以“/”开头的，并且是唯一的。例如：

**/home/zherry/project。**

相对路径是从当前目录沿目录树到文件所在目录的路径。

在访问当前目录附近的文件时，使用相对路径可以简化路径的描述，尤其是当目录的层次较深的情况下。

- 相对路径是以“.”或“..”开始的。
- “.”表示用户当前操作所处的位置，而“..”表示上级目录；
- 要把“.”和“..”当做目录来看。

## 4. 用户主目录

用户刚登录到系统中时所处的目录称为主目录(home directory)，系统默认的用户主目录是 **/home/user-id**。其中，**user-id** 是用户的登录名。例如：用户 **jhm** 的主目录是 **/home/jhm** (**root** 例外，它的主目录是 **/root**)。主目录就是用户登录后的初始工作目录，用“**~**”表示。用户对自己的主目录拥有全部权限，可以在其下任意组织自己的文件。

### 3.3.3 常用的目录操作命令

Linux系统提供了一些专门针对目录进行操作的命令，常用的是建立、删除、查看和改变目录，如表3-6所示。此外，由于**目录也是文件**，所以许多文件操作命令，如复制、移动、删除、更改属性等，也适用于对目录进行操作。

表 3-6 常用的目录操作命令

功能分类	命 令
建立、删除目录	<code>mkdir</code> 、 <code>rmdir</code>
显示、改变当前目录	<code>pwd</code> 、 <code>cd</code>
显示目录内容	<code>ls</code>

## 1. 显示与改变当前目录

访问**当前目录**中的文件时是不需要加路径名的，因此，当需要集中对某个目录中的文件进行操作时，先进入这个目录，使其成为当前目录，就可大大简化命令的输入。

要了解自己当前处在哪个目录下，可用**pwd(present working directory)**命令；

要改变当前目录，可用**cd(change directory)**命令。

## pwd命令

【功能】显示当前工作目录的**绝对路径**，无参数。

【格式】pwd

例3.7 pwd命令用法示例：

```
$ pwd
```

```
/home/zhou
```

## cd命令

**【功能】** 可以让用户在不同的目录间切换，但该用户必须有足够的权限进入目的目录

**【格式】** `cd [目录]`

**【说明】** 不指定目录参数时，进入用户登录的主目录。

```
$ cd
```

```
$ pwd
```

```
/home/zhou
```



- 使用cd进入指定目录

```
cd /home/student
```

```
pwd
```

```
/home/student
```

- “..”代表上一级目录

```
cd ..
```

```
pwd
```

```
/home
```

- 进入home目录下的子目录user

`cd user`

`pwd`

`/home/user`

- 回到用户登陆后的初始目录

`cd`

`pwd`

`/home/zhou`

## cd命令用法示例：相对路径

```
$ pwd
```

```
/home/jhm
```

```
$ cd ./project/hello
```

```
$ pwd
```

注意空格

```
/home/jhm/project/hello
```

```
$ cd ../../book
```

```
$ pwd
```

```
/home/jhm/book
```

往用户主目录去

## 特殊用法

**\$ cd ~**

直接进入**用户主目录**，与不加“~”相同

**\$ cd /**

直接进入根目录

- 更改目录位置至用户zhou的主目录
- 更改目录位置为当前用户主目录

```
[root@localhost root]# cd ~zhou
[root@localhost zhou]# pwd
/home/zhou
[root@localhost zhou]# cd ~
[root@localhost root]# pwd
/root
[root@localhost root]#
```

## 2. 显示目录内容

显示目录内容就是列出目录中所包含的文件以及文件的各种相关信息，子目录也作为一个文件列出，用于显示目录中的文件列表的命令是**ls(list)命令**。通常在进行文件操作前，应先用ls命令了解现有文件的状况。

## ls命令

**【功能】** 显示指定文件或指定目录中的所有文件的信息。

**【格式】** ls [选项] [文件或目录]

**【选项】**

**-a** 显示所有文件及目录，包括隐藏文件、现行目录“.”及上层目录“..”。

**-A** 显示所有文件及目录，包括隐藏文件。

**-R**      **递归**显示下层子目录。

**-F**      显示文件类型描述符：**目录** 名后标记/，**可执行的普通文件**后标记\*，**符号链接**后面标记@，**管道**后面标记|，**socket文件**后标记=

**-d**      如果参数是目录，只显示目录的名称而不显示此目录下的个各文件。

**-u**      显示文件的最近访问时间，与-l连用。



**-c**      显示文件的最近变更时间，与-l连用。

**-t**      按文件修改时间排序显示。

**-l**      按长格式显示文件详细信息。第一列：类型及访问权限；第二列：文件的硬链接数或者目录子目录数；第三列：文件的所有者；第四列：文件的所有组；第五列：文件所占的字节数；第六-八列：文件上一次的修改时间；第九列：文件名。

## 例3.9 ls命令用法示例:

\$ **cd ~**

\$ **ls**    #显示当前目录下的文件名

\$ **ls /home**   #显示目录/home下的文件名

\$ **ls -a**        #显示当前目录下的所有文件名

**\$ ls -F**      #显示当前目录下的文件名及类型

**\$ ls -l**      #显示当前目录下的文件的详细信息

- 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：

**ls -la**

**total 16**

```
drwxr-xr-x  4 root  root  4096 Jan  1 11:28 .  
drwxr-x--- 11 root  root  4096 Jan  1 11:27 ..  
drwxr-xr-x  2 root  root  4096 Jan  1 11:27 team01  
drwxr-xr-x  2 root  root  4096 Jan  1 11:28 team02
```

- 列出子目录中以字母t打头的全部非隐藏文件，使用下面的命令：

**ls t\***

### 3. 创建与删除目录

为了分类保存文件，用户可以建立自己的目录。建立目录用**mkdir**(make directory)命令，删除目录用**rmdir**(remove directory)命令。

#### mkdir命令

【功能】建立目录同时还可以给目录设置权限。

【格式】**mkdir** [选项] [目录名]

【选项】

**-m**  $\square$ 限            按指定的 $\square$ 限建立目 $\square$ 。

**-p**  $\square\square$ 建立目 $\square$ （后面可跟路径）。即，若目 $\square$ 的父目 $\square$ 不存在， $\square$ 一并建立其父目 $\square$ 。

【说明】未指定目录权限时，默认权限为**777**-创建掩码（**755**）

### 例3.10 mkdir命令用法示例：

```
$ ls
```

```
book  memo  project
```

```
$ mkdir -m 744 temp    #建立temp目录，权限744
```

```
$ ls
```

```
book  memo  project  temp
```

```
$ mkdir -p ./backup/version1
```

```
# 递归建立/backup/version1目录
```

# rmmdir命令

【功能】删除目录。

【格式】rmmdir [选项] [目录]

【选项】

-p 递归删除目录，即当子目录删除后其父目录为空时，也一同被删除。

【说明】若目录不空，则删除操作不能成功。

## 例3.11 rmmdir命令用法示例：

```
$ ls
```

```
backup  book  memo  project  temp
```

```
$ rmmdir temp          #删除空目录temp
```

```
$ ls
```

```
backup  book  memo  project
```

**\$ rmdir project                      #删除非空目录project**

**rmdir: `project': Directory not empty**

**\$ ls**

**backup   book   memo   project**

**\$ rmdir -p ./backup/version1**

**#递归删除目录./backup/version1**

**\$ ls**

**book   memo   project**

**\$**



# 链接文件

在Linux系统中，内核为每一个新创建的文件分配一个inode（索引节点）号，文件属性保存在索引节点里，在访问文件时，索引节点被复制到内存里，从而实现文件的快速访问。

- inode -索引节点。每个存储设备或存储设备的分区（硬盘、软盘、U盘等）被**格式化为文件系统**后，应该有两部份，一部份是inode，另一部份是Block，Block是用来存储数据用的。而inode呢，就是用来存储这些**数据的信息**，这些信息包括文件大小、属主、归属的用户组、读写权限等。
- inode为每个文件进行**信息索引**，所以就有了inode的数值。操作系统根据指令，能通过inode值最快的找到相对应的文件。

## 链接文件简介

链接是一种在**共享文件**和访问它的**用户的若干目录项**之间建立联系的方法。Linux系统中包括**两种链接**：**硬链接**（Hard Link）和**软链接**（Soft Link），软链接又称为**符号链接**。

# 1. 硬链接

硬链接是一个指针，指向文件索引节点，系统并不为它重新分配inode，可以用**ln命令**来建立硬链接。

硬链接节省空间，也是Linux系统整合文件系统的传统方式。

**硬链接文件有两个限制：**

- (1) 不允许给目录创建硬链接；**
- (2) 只有在同一文件系统中的文件之间才能创建链接。**

## 2. 软链接

软链接又称为符号链接，这个文件包含了另一个文件的路径名。

### 3. 硬链接和软链接的区别

硬链接记录的是目标的inode，软链接记录的是目标的路径。

软链接就像是快捷方式，而硬链接就像是备份。

软链接可以做跨分区的链接，而硬链接由于inode的缘故，只能在本分区中做链接。所以，软链接的使用频率要高得多。

## 链接文件的使用

使用 **ln** 命令可以创建链接文件（包括软链接文件和硬链接文件）。

命令语法：

**ln** [选项] [源文件名] [链接文件名]



## 链接文件的使用

**【例】** 创建硬链接文件，在修改删除源文件后，硬链接文件的变化。

```
[root@redhatAS6 dd]# cat a
hello
[root@redhatAS6 dd]# ln a b
[root@redhatAS6 dd]# ls -l
总用量 20
-rw-r--r--. 2 root root    6 3月  8 16:59 a
drwxrwxr-x. 2 han  han  4096 3月  8 08:49 aa
-rw-r--r--. 2 root root    6 3月  8 16:59 b
drwxrwxr-x. 2 han  han  4096 3月  8 08:57 cc
-rw-rw-r--. 1 han  han    56 9月 18 2013 h.c
[root@redhatAS6 dd]# cat b
hello
```

```
[root@redhatAS6 dd]# echo world>>a
[root@redhatAS6 dd]# cat a
hello
world
[root@redhatAS6 dd]# cat b
hello
world
[root@redhatAS6 dd]# rm -rf a
[root@redhatAS6 dd]# ls -l
总用量 16
drwxrwxr-x. 2 han  han  4096 3月  8 08:49 aa
-rw-r--r--. 1 root root   12 3月  8 17:04 b
drwxrwxr-x. 2 han  han  4096 3月  8 08:57 cc
-rw-rw-r--. 1 han  han   56 9月 18 2013 h.c
[root@redhatAS6 dd]# cat b
hello
world
```

## 链接文件的使用

**【例】** 创建软链接文件，在修改删除源文件后，软链接文件的变化。

```
[root@redhatAS6 dd]# cat c
work
[root@redhatAS6 dd]# ln -s c d
[root@redhatAS6 dd]# ls -l
总用量 20
drwxrwxr-x. 2 han han 4096 3月 8 08:49 aa
-rw-r--r--. 1 root root 12 3月 8 17:04 b
-rw-r--r--. 1 root root 5 3月 8 17:11 c
drwxrwxr-x. 2 han han 4096 3月 8 08:57 cc
lrwxrwxrwx. 1 root root 1 3月 8 17:11 d -> c
-rw-rw-r--. 1 han han 56 9月 18 2013 h.c
[root@redhatAS6 dd]# cat d
work
```

```
[root@redhatAS6 dd]# echo hard>>c
[root@redhatAS6 dd]# cat c
work
hard
[root@redhatAS6 dd]# cat d
work
hard
[root@redhatAS6 dd]# rm -rf c
[root@redhatAS6 dd]# ls -l
总用量 16
drwxrwxr-x. 2 han  han  4096 3月   8 08:49 aa
-rw-r--r--. 1 root root   12 3月   8 17:04 b
drwxrwxr-x. 2 han  han  4096 3月   8 08:57 cc
lrwxrwxrwx. 1 root root    1 3月   8 17:11 d -> c
-rw-rw-r--. 1 han  han   56 9月  18 2013 h.c
[root@redhatAS6 dd]# cat d
cat: d: 没有那个文件或目录
```

区别：

**1.硬链接是原文件 / 链接文件共用一个inode号，说明他们是同一个文件，而软链接则是原文件 / 链接文件拥有不同的inode号，表明他们是两个不同的文件**

**2.在文件属性上软链接明确写出了是链接文件，而硬链接没有写出来，因为在本质上硬链接文件和原文件是完全平等关系**

**区别：**

**3. 链接数目是不一样的，软链接的链接数目不会增加**

**4. 文件大小是不一样的，硬链接文件显示的大小是跟原文件是一样的。而这里软链接显示的大小与原文件就不同了**

**5. 软链接没有任何文件系统的限制，任何用户可以创建指向目录的符号链接**

总之，建立软链接就是建立了一个**新文件**。当访问链接文件时，系统就会发现他是个链接文件，它读取链接文件找到真正要访问的文件。

**软链接的不足之处：**因为链接文件包含有原文件的路径信息，所以当原文件从一个目录移动到其他目录中，再访问链接文件，系统就找不到了，而硬链接就没有这个缺陷，想怎么移就怎么移；软链接还需要系统分配额外的空间用于建立新的索引节点和保存原文件的路径。

## 小 结

在Linux系统下除了一般文件之外，所有的目录和设备（光驱、硬盘等）都是以**文件**的形式存在。Linux文件类型和Linux文件的文件名所代表的意义是两个不同的概念。Linux文件类型常见的有：**普通文件、目录文件、字符设备文件、块设备文件和符号链接文件**等。



## 小 结

Linux系统都有**根文件系统**，它包含系统引导和使其他文件系统得以挂载所必要的文件，Linux系统的目录结构是**分层的树形结构**，都是挂载在根文件系统“/”下。

在Linux系统下对文件和目录进行操作的命令主要有pwd, cd, ls, mkdir, rmdir等。

## 小 结

在Linux系统中，内核为每一个新创建的文件分配一个inode（索引节点）号，文件属性保存在索引节点里，在访问文件时，索引节点被复制到内存里，从而实现文件的快速访问。

## 小 结

在Linux系统中包括硬链接和软链接两种链接。硬链接是一个指针，指向文件索引节点，系统并不为它重新分配inode。硬链接节省空间，也是Linux系统整合文件系统的传统方式。软链接又叫符号链接，这个文件包含了另一个文件的路径名。可以是任意文件或目录，可以链接不同文件系统的文件。

### 3.3.4 常用的文件操作命令

Linux系统提供了丰富的**文件操作命令**，可以完成各种各样的文件操作。而且，大部分**文件操作命令也适用于目录文件**。本小节介绍几个常用的文件操作命令，见表3-7。

表 3-7 常用的文件操作命令

功能分类	命 令
文件显示	<code>cat</code> 、 <code>more</code> 、 <code>less</code>
文件复制、删除和移动	<code>cp</code> 、 <code>rm</code> 、 <code>mv</code>
文件内容的统计与排序	<code>wc</code> 、 <code>sort</code>
改变文件的存取权限	<code>chmod</code>
改变文件的时间标签或创建新文件	<code>touch</code>

## 1. 文件的显示

阅读一个文本文件的最简单的方法就是用文件显示命令将文件内容显示在屏幕上。显示文本文件的常用命令是 **cat(concatenate)**、**more** 和 **less** 命令。

### cat 命令

【功能】显示文件内容、创建新文件或合并文件内容成一个文件。

【格式】 **cat** [选项] [文件]

如果文件已经存在，**cat** 就一次显示存在文件中的全部内容。与后面的 **more** 命令的分屏显示不同，**cat** 命令是全部显示。

## 【选项】

**-A** 显示所有字符，包括换行符、制表符及其他非打印字符。

**-n** 对输出的所有行进行编号并显示行号。

**-b** 和-n相似，但对于空白行不编号。

**-s** 将连续的空白行压缩为一个空白行。

**【说明】** 指定多个文件时，依次显示各个文件；未指定文件时，读标准输入(默认为键盘)并显示。

## 例3.12 cat命令用法示例：

```
$ cat doc1.txt
```

#显示一个文件

#将文件doc1内容加行号后输入到文件doc2中

```
$ cat -n doc1.txt >doc2.txt
```

```
$ cat doc2.txt
```

#将文件doc1和文件doc2的内容加行号后输入到文件doc3中

```
$ cat -b doc1.txt doc2.txt >>doc3.txt
```

```
$ cat doc3.txt
```

```
[root@redhatAS6 han]# cat >file1.txt <<EOF
> computer
> cpu
> EOF
[root@redhatAS6 han]# cat file1.txt
computer
cpu
[root@redhatAS6 han]# cat >file2.txt <<EOF
> hello
> world
> EOF
[root@redhatAS6 han]# cat file2.txt
hello
world
[root@redhatAS6 han]# cat file1.txt
computer
cpu
[root@redhatAS6 han]# cat -n file1.txt file2.txt >>file3.txt
[root@redhatAS6 han]# cat file3.txt
  1  computer
  2  cpu
  3  hello
  4  world
```





## more命令

【功能】分屏显示文件内容。

【格式】more [选项] [文件]

【选项】

- p 不刷屏，清屏。
- s 将连续的空白行显示为一个空白行。
- +*n* 由第*n*行开始显示。
- +/*str* 由含有*str*字符串的地方开始显示。
- n 一次显示的行数。

【说明】浏览时可使用如下键进行控制，浏览到末页后自动退出。

**Enter**向下翻一行。

**Space** 向下翻页。

**b** 向上翻页。

**/string** 查找字符串string。

**q** 退出。

### 例3.13 more一个长文件

在浏览过程中，屏幕左下角会显示“**more**”，命令因此得名。当浏览到末页时命令会自动退出。

**less**命令具有更好的交互性。用户可以完全控制浏览的过程，上下翻页和滚行，到末页时也不会自动退出。

## less命令

【功能】交互式浏览文件内容。

【格式】less [选项] 文件...

【选项】与more相似。

【说明】浏览时可以用PageUp、PageDown上下翻页，用↑、↓上下滚行。其他按键与more相同。

man命令就是采用less控制浏览手册页的

# head：显示指定文件前若干行

使用head命令可以显示指定文件的前若干行  
文件内容。

命令语法：

head [选项] [文件]

-q 隐藏文件名

-v 显示文件名

-c<字节> 显示字节数

-n<行数> 显示的行数 如果没有给出具体n值，默认设置为  
10行

**【例】** 查看文件/etc/passwd的前3行内容。

administrator@NeoKylin-CNVL:~/桌面

文件(E) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

administrator@NeoKylin- CNVL 桌面]\$ head /etc/passwd

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

bin: x: 1: 1: bin: /bin: /sbin/nologin

daemon: x: 2: 2: daemon: /sbin: /sbin/nologin

adm: x: 3: 4: adm: /var/adm: /sbin/nologin

lp: x: 4: 7: lp: /var/spool/lpd: /sbin/nologin

sync: x: 5: 0: sync: /sbin: /bin/sync

shutdown: x: 6: 0: shutdown: /sbin: /sbin/shutdown

halt: x: 7: 0: halt: /sbin: /sbin/halt

mail: x: 8: 12: mail: /var/spool/mail: /sbin/nologin

operator: x: 11: 0: operator: /root: /sbin/nologin

administrator@NeoKylin- CNVL 桌面]\$ head - 3 /etc/passwd

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

bin: x: 1: 1: bin: /bin: /sbin/nologin

daemon: x: 2: 2: daemon: /sbin: /sbin/nologin

## tail：查看文件末尾数据

使用tail命令可以查看文件的末尾数据。

命令语法：

**tail [选项] [文件名]**

+num 从第num行以后开始显示文件内容

-num 从距文件末尾num行数开始显示文件内容，如果省略num值，则默认为10

-c<字节> 显示字节数

**【例】** 查看文件/etc/passwd末尾3行数据。

administrator@NeoKylin-CNVL:~/桌面

文件(E) 编辑(E) 查看(V) 搜索(S) 终端(I) 帮助(H)

[administrator@NeoKylin- CNVL 桌面]\$ tail - 3 /etc/passwd

administrator: x: 1000: 1000: administrator: /home/administrator: /bin/bash

mysql: x: 990: 988: MySQL server: /var/lib/mysql: /bin/bash

zhou: x: 1001: 1001: : /home/zhou: /bin/bash

[administrator@NeoKylin- CNVL 桌面]\$ tail - c 100 /etc/passwd

n/bash

mysql: x: 990: 988: MySQL server: /var/lib/mysql: /bin/bash

zhou: x: 1001: 1001: : /home/zhou: /bin/bash

## 2. 文件的复制、移动与删除

复制文件使用 **cp(copy)命令**，删除文件使用 **rm(remove)命令**，移动文件和重命名文件使用 **mv(move)命令**。



## cp 命令

【功能】 复制文件。

【格式】 **cp** [选项] 源文件或目录 目标文件或目录

**cp** [选项] [源文件...][ 目标目录]

【选项】

**-i** 交互模式，当目标文件存在时，提示是否覆盖。

输入 **y** 或 **Y** 覆盖，输入其他字符不覆盖。

**-r** 递归复制目录。

**-b** 为被覆盖的文件建立**备份**。备份文件的名称是原文件名后加“~”。

**-f** 强制复制。在覆盖目标文件之前不给提示信息要求用户确认。

**-p** 保持文件原有属性。

**-v** 显示操作结果。

## 例3.14 复制一个文件：

d1.txt

doc1.txt

\$ **cp doc1.txt d1.txt**    #在当前目录下复制一个文件

\$cat d1.txt

这个操作将覆盖  
**d1.txt**

\$ **cp doc2.txt d2.txt**    #在当前目录下复制一个文件

当前文件中没有**d2.txt**时，此命令相当于生成了一个文件

使用**cp**命令复制文件到一个已存在的目录下，且还可以复制多个文件到目标目录。

```
[ administrator@NeoKylin- CNVL zwf] $ cp doc1.txt /home/administrator/zwf/z  
[ administrator@NeoKylin- CNVL zwf] $ cp doc1.txt /home/administrator  
[ administrator@NeoKylin- CNVL zwf] $ cp doc1.txt /home/administrator/zwf/z/d1.txt  
[ administrator@NeoKylin- CNVL zwf] $
```

## rm命令

【功能】 删除文件。

【格式】 **rm** [选项] [文件...]

【选项】

**-f** 忽略不存在的文件，不作提示。

**-i** 删除前提示用Y确Y。

**-r** 递归删除目录。

**-v** 显示操作结果。

【说明】 若参数是目录文件，需要有**-r**选项，否则报错。

## 例3.16 用rm命令删除文件

**\$ ls**

**a.out hello hello.c hello.o makefile temp**

**\$ rm a.out** **#删除一个文件**

**\$ ls**

**hello hello.c hello.o makefile temp**

**\$ rm -v hello** **#删除一个文件，显示结果**  
**removed 'hello'**

**\$ rm -i hello.\***                    **#删除多个文件，逐个提示确认**

**rm: remove regular file 'hello.c'? **n****

**rm: remove regular file 'hello.o'? **y****

**\$ ls -F**

**hello.c   makefile   temp/**

**\$ rm -r ~/temp**

**#删除一个目录时要注意所删除目录所在路径**

**\$ rm -rf ~/temp    #连同temp目录中的内容一起删除**

## **mv命令**（能够对文件和目录**更改名称**以及**移动**文件和目录的**路径**）

**【功能】** 移动文件、重命名文件。

**【格式】** mv **【选项】** 源文件或目录 目标文件或目录

### **【选项】**

- i** 覆盖前提示用户确认。
- f** 不提示用户确认，直接覆盖。（相对危险）
- b** 为被覆盖的文件建立备份。备份文件的名称是原文件名后加“~”。
- v** 显示操作结果。



## 例3.17 用mv命令移动文件

```
$ ls
```

```
hello.c hello.o hello.save makefile temp
```

```
$ mv -v hello.save hello.o ~/temp
```

```
'hello.save' -> 'temp/hello.save'
```

```
'hello.o' -> 'temp/hello.o'
```

```
$ ls
```

```
hello.c makefile temp
```

结果将hello.save hello.o 移到 temp中。

```
$ mv -v makefile ~/temp/makefile.old
```

```
  'makefile' -> 'temp/makefile.old'
```

```
$ ls . /temp
```

```
  .:
```

```
hello.c      temp
```

```
temp:
```

```
hello.o      hello.save  makefile.old
```

```
[ administrator@NeoKylin- CNVL zwf]$ mv -v doc3.txt ~/zwf/z  
"doc3.txt" -> "/home/administrator/zwf/z/doc3.txt"  
[ administrator@NeoKylin- CNVL zwf]$ mv -v doc3.txt ~/zwf/z/d2.txt  
"doc3.txt" -> "/home/administrator/zwf/z/d2.txt"
```

### 3. 文件内容的统计与排序

Linux提供了许多用于文件内容处理的命令，比较常用的有统计文件字数的wc(word count)命令和对文件内容排序的sort命令。

#### WC命令

【功能】 显示文件的字节数、字数和行数。

【格式】 **wc** [选项] [文件]

【选项】

**-c**          只□□字□数。

**-l**          只□□行数。

**-m**          只□□字符数。

**-w**          只□□字数。

## 例3.19 统计一个文件的内容：

\$ **cat doc1**

To see a world in a grain of sand,  
And a heaven in a wild flower,  
Hold infinity in the palm of your hand,  
And eternity in an hour.

\$ **wc doc1**

4 29 131 doc1(行数, 单词数, 字节数, 文件名)

## sort命令

sort命令是个常用的文字处理命令，它将文本文件的各行按ASCII字符顺序由小到大排序，并输出排序后的结果。

【功能】对文本文件中的各行按字符顺序排序并显示。

【格式】sort [选项] [文件]...

### 【选项】

- b 忽略开始的空白。
- d 只考虑字母、数字和空格。
- f 忽略大小写。
- k 指定从第n个字段开始的内容作为排序关键字。
- r 逆序排列。

【说明】未指定文件时，读标准输入文件。

## 例3.20 排序一个文件的内容：

```
$ cat doc1.txt
```

```
$ sort -r doc1.txt
```

```
[ administrator@NeoKylin- CNVL zwf] $ cat doc1. txt
```

```
a
```

```
b
```

```
c
```

```
[ administrator@NeoKylin- CNVL zwf] $ sort - r doc1. txt
```

```
c
```

```
b
```

```
a
```

# uniq：将重复行从输出文件中删除

使用uniq命令可以将文件内的重复行数据从输出文件中删除，只留下每条记录的唯一样本。

命令语法：

uniq [选项] [文件]

-d 仅显示重复行。

-u 仅显示不重复的行。



```
[ administrator@NeoKylin- CNVL zwf]$ cat doc1.txt
```

```
a
```

```
a
```

```
b
```

```
c
```

```
[ administrator@NeoKylin- CNVL zwf]$ uniq doc1.txt
```

```
a
```

```
b
```

```
c
```

```
[ administrator@NeoKylin- CNVL zwf]$ uniq -d doc1.txt
```

```
a
```

```
[ administrator@NeoKylin- CNVL zwf]$ uniq -u doc1.txt
```

```
b
```

```
c
```

## cut命令

使用cut命令可以从文件的每行中显示出选定的字节、字符或字段。

命令语法：

```
cut { -b List [ -n ] | -c List | -  
f List [ -s ] [ -d Character ] } [ 文件 ]
```

-b : 以字节为单位进行分割。

-c : 以字符为单位进行分割。

**按字节cut：**

**注意：**一个空格算一个字节，一个汉字算三个字节

```
[root@localhost ~]# date
```

```
2011年08月11日 星期四20:44:52 EDT
```

```
[root@localhost ~]# date | cut -b 1-4      #取前四个字节
```

```
2011
```

```
[root@localhost ~]# date  
2011年 08月 11日 星期四 20:44:52 EDT
```

一个汉字算三个字节

```
[root@localhost ~]# date | cut -b 1-7  
2011年
```

多个定位之间用逗号隔开：

```
[root@localhost ~]# date | cut -b 1-7,10  
2011年8
```

## comm命令

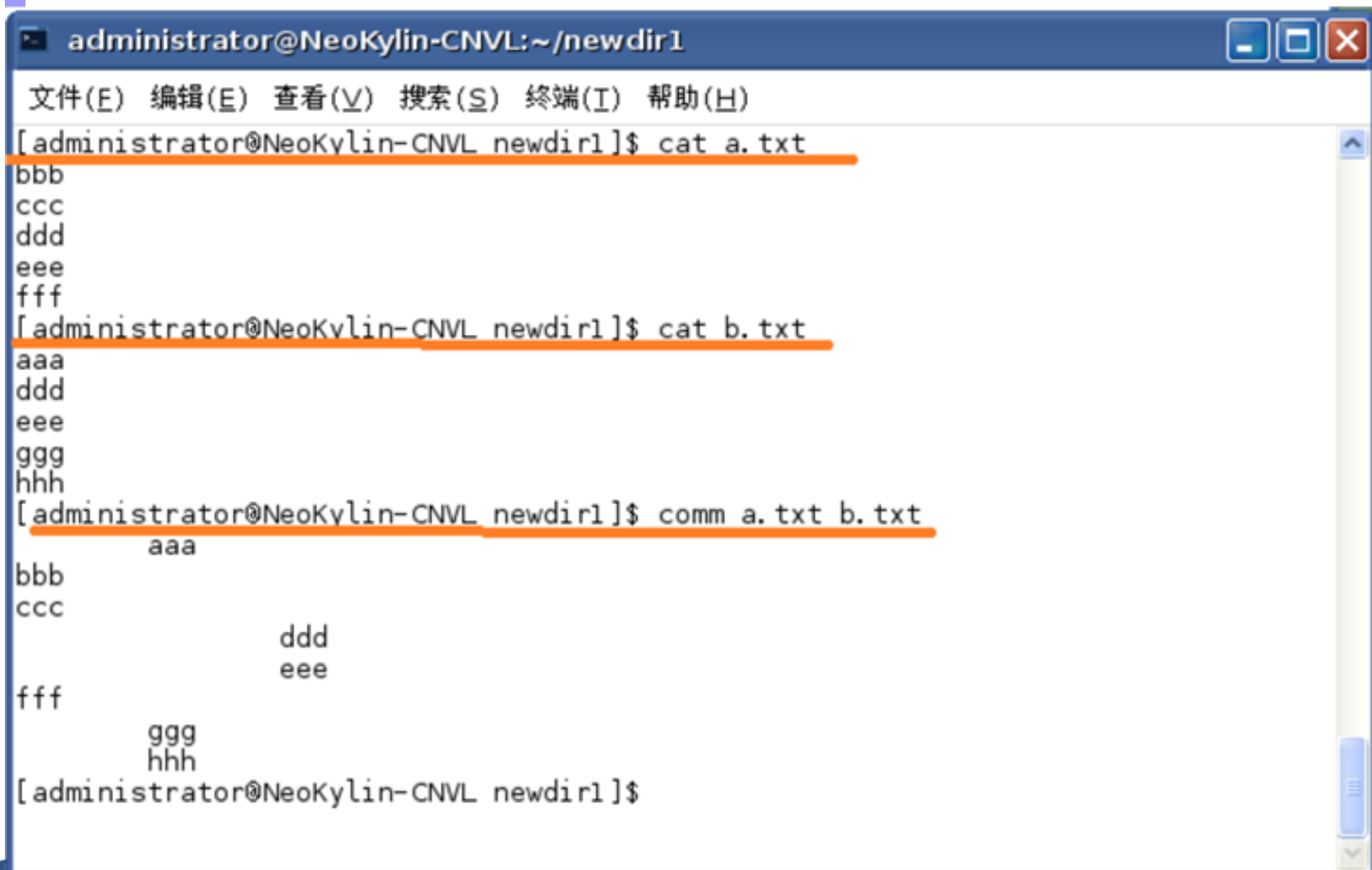
使用comm命令可以比较两个已排过序的文件，并将其结果显示出来。

命令语法：

```
comm [-123] [--help] [文件1][文件2]
```

参数：

- 1 不显示只在第1个文件里出现过的列。
- 2 不显示只在第2个文件里出现过的列。
- 3 不显示只在第1和第2个文件里出现过的列。



The image shows a terminal window titled "administrator@NeoKylin-CNVL:~/newdir1". The window has a menu bar with "文件(E)", "编辑(E)", "查看(V)", "搜索(S)", "终端(I)", and "帮助(H)". The terminal content shows the execution of three commands: "cat a.txt", "cat b.txt", and "comm a.txt b.txt". The output of "comm a.txt b.txt" shows the differences between the two files. The first column contains lines unique to a.txt (bbb, ccc, ddd, eee, fff), the second column contains lines unique to b.txt (aaa, ddd, eee, ggg, hhh), and the third column contains lines common to both (ddd, eee). The terminal window has a blue title bar and standard window controls (minimize, maximize, close) on the right.

```
administrator@NeoKylin-CNVL:~/newdir1
文件(E) 编辑(E) 查看(V) 搜索(S) 终端(I) 帮助(H)
[administrator@NeoKylin-CNVL newdir1]$ cat a.txt
bbb
ccc
ddd
eee
fff
[administrator@NeoKylin-CNVL newdir1]$ cat b.txt
aaa
ddd
eee
ggg
hhh
[administrator@NeoKylin-CNVL newdir1]$ comm a.txt b.txt
aaa
bbb
ccc
          ddd
          eee
fff
          ggg
          hhh
[administrator@NeoKylin-CNVL newdir1]$
```

administrator@NeoKylin-CNVL:~/newdir1

文件(E) 编辑(E) 查看(V) 搜索(S) 终端(I) 帮助(H)

[administrator@NeoKylin-CNVL newdir1]\$ comm -12 a.txt b.txt

ddd

eee

[administrator@NeoKylin-CNVL newdir1]\$

## diff命令

使用diff命令可以逐行比较两个文本文件，**列出其不同之处**。它比comm命令完成更复杂的检查。它对给出的文件进行系统的检查，并显示出两个文件中所有不同的行，**不要求事先对文件进行排序**。

命令语法：

```
diff [选项] file1 file2
```



```
administrator@NeoKylin-CNVL:~/newdir1
文件(E) 编辑(E) 查看(V) 搜索(S) 终端(I) 帮助(H)
[administrator@NeoKylin-CNVL newdir1]$ cat a.txt
bbb
ccc
ddd
eee
fff
[administrator@NeoKylin-CNVL newdir1]$ cat b.txt
aaa
ddd
eee
ggg
hhh
[administrator@NeoKylin-CNVL newdir1]$ diff a.txt b.txt
1,2c1
< bbb
< ccc
---
> aaa
5c4,5
< fff
---
> ggg
> hhh
[administrator@NeoKylin-CNVL newdir1]$
```

## 4. 文件查找与搜索

Linux系统下文件和命令查找命令，这些命令有grep, find, locate, whereis, file, whatis以及which等。

# grep命令

使用grep命令可以查找文件中符合条件的字符串。

命令语法：

grep [选项] [查找模式] [文件名]

## 【选项】

- v 列出不包含匹配字符串的行。
- c 不显示匹配的行，只列出匹配的行数。
- l 只显示含有匹配字符串的文件名。
- r 递归地搜索目录下的所有文件和子目录。
- n 在每个匹配行前加行号显示。
- i 匹配时不区分大小写。如模式“may”可匹配may、May和MAY。
- w 匹配整个单词，而不是单词的一部分。如模式“magic”只匹配magic，而不匹配magical。

将/etc/passwd，有出现 root 的行取出来

# **grep root /etc/passwd**

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

operator:x:11:0:operator:/root:/sbin/nologin

将/etc/passwd，有出现 root 的行取出来，同时显示  
这些行在/etc/passwd的行号

# **grep -n root /etc/passwd**

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

30:operator:x:11:0:operator:/root:/sbin/nologin

**\$ cat list**

**To see a world in a grain of sand,  
And a heaven in a wild flower,  
Hold infinity in the palm of your hand,  
And eternity in an hour.**

**\$ grep and list**      **#在文件中搜索“and”，显示匹配的行**

**To see a world in a grain of sand,  
Hold infinity in the palm of your hand,**

**\$ grep -c and list #只显示含有“and”字符串的行数**  
**2**

**\$ grep -v and list #显示不含“and”字符串的行**  
**And a heaven in a wild flower,**  
**And eternity in an hour.**

**\$ grep -i and list #搜索“and”，忽略大小写**  
**To see a world in a grain of sand,**  
**And a heaven in a wild flower,**  
**Hold infinity in the palm of your hand,**  
**And eternity in an hour.**

字符类的反向选择 **[^]** : 如果想要搜索到有 **oo** 的行, 但不想要 **oo** 前面有 **g** :

```
[root@www ~]# grep -n '[^g]oo' regular_express.txt  
2:apple is my favorite food.  
3:Football game is not use feet only.  
18:google is the best tools for search keyword.  
19:gooooooogle yes!
```

假设 **oo** 前面不想要有小写字母 :

```
[root@www ~]# grep -n '[^a-z]oo' regular_express.txt  
3:Football game is not use feet only.
```



## find命令

使用find命令可以查找文件系统中**符合条件的文件或目录**，可以指定文件的名称、类别、时间、大小以及权限等不同信息的组合，只有完全相符的文件才会被列出来。

命令语法：

```
find [路径] [选项] [-print]
```

- name filename**      #查找名为**filename**的文件
- perm**      #按执行权限来查找
- user username**      #按文件属主来查找
- group groupname**      #按文件所属组来查找
- mtime -n +n**      #按文件**更改**时间来查找文件，**-n**指**n**天以内，**+n**指**n**天以前
- atime -n +n**      #按文件**访问**时间来查**GIN: 0px">**
- ctime -n +n**      #按文件**创建**时间来查找文件，**-n**指**n**天以内，**+n**指**n**天以前

- type b/d/c/p/l/f**      **#查是块设备、目录、字符设备、管道、符号链接、普通文件**
- size [+ -]n[bckM]**      **#查找大小为n的文件，后面的字符表示单位。c为字节，b为块(512 B)，k为千(1024 B)字节，M为兆(1024 KB)字节。缺省为b。“+n”表示超过n，“-n”表示不超过n，“n”表示正好n。**

**【说明】**未指定搜索条件时，显示目录下的所有文件，包含隐藏文件。未指定目录时，默认为当前目录。未指定操作时，缺省的操作是**-print**。



### 例3.21 按条件查找并显示结果：

```
$ find -name 'word*' -print
```

#在当前目录下查找名字以word开头的文件

```
$ find /home -mtime -2
```

#在/home下查最近两天内改动过的文件

```
$ find /home -atime -1
```

# 查1天之内被存取过的文件

```
$ find /home -user cnsn
```

#列出/home目录内属于用户cnsn的文件或目录

```
find /home -size +512k
```

#查大于512k的文件

# locate命令

使用locate命令可以用于查找文件，比find命令的搜索速度快，它需要一个数据库，这个数据库由每天的例行工作（crontab）程序来建立。当建立好这个数据库后，就可以方便地搜寻所需文件了。

命令语法：

```
locate [-d <数据库文件>][--help][--version][范本样式]
```

## 【例】 查找文件httpd.conf。

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

```
[administrator@NeoKylin-CNVL 桌面]$ locate httpd.conf  
/etc/httpd/conf/httpd.conf  
/usr/lib/tmpfiles.d/httpd.conf  
[administrator@NeoKylin-CNVL 桌面]$
```

## whereis命令

使用whereis命令可以查找指定文件、命令和手册页的位置。

命令语法：

whereis [选项] [文件名]

只搜索二进制文件（参数-b）

man说明文件（参数-m）

源代码文件（参数-s）

如果省略参数，则返回所有信息

**【例】** 查找mv命令的二进制文件在什么目录下。

```
[administrator@NeoKylin-CNVL 桌面]$  
[administrator@NeoKylin-CNVL 桌面]$ whereis -b mv  
mv: /usr/bin/mv  
[administrator@NeoKylin-CNVL 桌面]$
```



# file命令

使用file命令可以查询指定文件的文件类型。

命令语法：

file [选项] [文件名]

【例】 查看/boot目录下所有文件的文件类型。

```

[administrator@NeoKylin-CNVL 桌面]$ file /boot/*
/boot/config-3.12.11-201.nk.1.x86_64: ASCII text
/boot/grub2: directory
/boot/initramfs-0-rescue-2914a656c159456d8b5caba7ff4afc69.img: regular file, no
read permission
/boot/initramfs-3.12.11-201.nk.1.x86_64.img: regular file, no
read permission
/boot/initrd-plymouth.img: gzip compressed data, from Unix, last modified: Thu Jan 14 15:20:15 2016, max compression
/boot/lost+found: directory
/boot/System.map-3.12.11-201.nk.1.x86_64: regular file, no
read permission
/boot/vmlinuz-0-rescue-2914a656c159456d8b5caba7ff4afc69: Linux kernel x86
boot executable bzImage, version 3.12.11-201.nk.1.x86_64 (mockbuild@kojibuilder3
) #1 SMP Sun Jun, RO-rootFS, swap_dev 0x4, Normal VGA
/boot/vmlinuz-3.12.11-201.nk.1.x86_64: Linux kernel x86
boot executable bzImage, version 3.12.11-201.nk.1.x86_64 (mockbuild@kojibuilder3
) #1 SMP Sun Jun, RO-rootFS, swap_dev 0x4, Normal VGA

```



# whatis命令

使用whatis命令可以查询指定命令的命令功能。

命令语法：

whatis [命令]

【例】 查询ls命令的功能。

```
[administrator@NeoKylin-CNVL 桌面]$ whatis ls  
ls (1) - list directory contents  
ls (lp) - list directory contents
```

# which 命令

使用which命令可以显示可执行命令的路径和它的别名。

命令语法：

which [命令]

【例】 显示ls命令的可执行命令的路径和它的别名。

```
[administrator@NeoKylin-CNVL 桌面]$ which ls  
alias ls='ls --color=auto'  
/usr/bin/ls  
[administrator@NeoKylin-CNVL 桌面]$
```

## 5 改变文件属性

用户可以用命令修改已有文件的访问权限等属性，达到控制文件的使用的目的。改变文件的访问权限用**chmod**(change mode)命令，改变文件的时间标签用**touch**命令。

### chmod命令

【功能】修改文件或目录的存取权限。

【格式】**chmod** [选项] [数字权限模式] 文件...

**chmod** [选项] [字符权限模式表达式]... 文件...

【选项】

**-R**        □□地改□指定目□及其下的文件和子目□的  
            □限属性。

文件使用者: **u g o a**

权限: **r w x**

增减: **+ 增加权限**

**- 删除权限**

**= 使它成为唯一权限**

## 例：用chmod命令修改文件的存取权限：

**\$ chmod a+x hello    #修改 hello 的权限，为所有人增加执行权**

**\$ chmod o-x hello    #修改hello的权限，取消其他用户的执行权**

```
[root@redhatAS6 han]# ls -l file2.txt
-rw-r--r--. 1 root root 12 3月  9 08:45 file2.txt
[root@redhatAS6 han]# chmod a+x file2.txt
[root@redhatAS6 han]# ls -l file2.txt
-rwxr-xr-x. 1 root root 12 3月  9 08:45 file2.txt
[root@redhatAS6 han]# chmod o-x file2.txt
[root@redhatAS6 han]# ls -l file2.txt
-rwxr-xr--. 1 root root 12 3月  9 08:45 file2.txt
```



```
$ chmod g=rx,o=x hello
```

#设hello权限为组用户可读和执行，其他人可执行，属主权限不变

```
$ chmod 664 hello
```

#设hello的权限为664

```
$ chmod go= hello      #取消组用户和其他用户对  
                        hello的任何权限
```

chmod命令的**模式表达式**中没有权限字符，表示组用户(g)和其他用户(o)没有任何权限，属主(u)的权限不变。

## touch命令

【功能】创建空文件(用文本编辑器 (vi) 或cat命令同样可以创建新文件)以及更改文件或目录的时间

【格式】touch [选项] 文件...

【选项】

- a 仅改变文件的访问时间。
- m 仅改变文件的修改时间。
- c 文件不存在时，不创建文件。
- t *STAMP* 使用STAMP指定的时间标签，

而不是系统现在时间。

## 例： 用touch命令修改文件的时间戳：

`$ touch file1.txt` #“触动”一个已有文件

```
[root@redhatAS6 han]# ls -l file1.txt
-rw-r--r--. 1 root root 13 3月  9 08:44 file1.txt
[root@redhatAS6 han]# touch file1.txt
[root@redhatAS6 han]# ls -l file1.txt
-rw-r--r--. 1 root root 13 3月  9 10:39 file1.txt
```

`$ touch file5.txt` #“触动”(新建)一个新文件  
将看到abc是一个空文件。

```
[root@redhatAS6 han]# touch file5.txt
```

```
[root@redhatAS6 han]# ls -l
```

总用量 64

-rw-r--r--.	2	root	root	6	3月	8	16:56	a
-rw-r--r--.	2	root	root	6	3月	8	16:56	b
drwxrwxr-x.	4	han	han	4096	3月	8	17:16	dd
drwxr-xr-x.	2	root	root	4096	3月	9	09:26	ddd
-rw-r--r--.	1	root	root	13	3月	9	10:39	file1.txt
-rwxr-xr--.	1	root	root	12	3月	9	08:45	file2.txt
-rw-r--r--.	1	root	root	53	3月	9	08:46	file3.txt
-rw-r--r--.	1	root	root	53	3月	9	09:21	file4.txt
-rw-r--r--.	1	root	root	0	3月	9	10:42	file5.txt

- **chown命令**
- **chown命令来改变文件所有者（属主）及用户组（属组）。**
- **chown [选项]... [所有者][:[组]] 文件...**
- **chown 所有者 文件名**

```
[root@redhatAS6 han]# ls -l
总用量 64
-rw-r--r--. 2 root root    6 3月    8 16:56 a
-rw-r--r--. 2 root root    6 3月    8 16:56 b
[root@redhatAS6 han]# chown han a
[root@redhatAS6 han]# ls -l
总用量 64
-rw-r--r--. 2 han  root    6 3月    8 16:56 a
-rw-r--r--. 2 han  root    6 3月    8 16:56 b
```

- **chown** [选项]... [所有者][:[组]] 文件...
- **chown** 所有者: 组 文件名

```
[root@redhatAS6 han]# ls -l
总用量 64
-rw-r--r--. 2 root root    6 3月    8 16:56 a
-rw-r--r--. 2 root root    6 3月    8 16:56 b
[root@redhatAS6 han]# chown han:han a
[root@redhatAS6 han]# ls -l
总用量 64
-rw-r--r--. 2 han  han    6 3月    8 16:56 a
-rw-r--r--. 2 han  han    6 3月    8 16:56 b
```

- **chgrp命令**
- 在shell中，使用chgrp命令来改变文件所在用户组（属组）。
- **chgrp [选项参数] [组] [文件]**

```
[root@redhatAS6 han]# ls -l
总用量 64
-rw-r--r--. 2 root root    6 3月   8 16:56 a
-rw-r--r--. 2 root root    6 3月   8 16:56 b
```

```
[root@redhatAS6 han]# chgrp han a
[root@redhatAS6 han]# ls -l
总用量 64
-rw-r--r--. 2 root han     6 3月   8 16:56 a
-rw-r--r--. 2 root han     6 3月   8 16:56 b
```



## 6. 设置文件掩码

用户可以用**umask命令**查看和设置文件创建掩码。

### umask命令

**【功能】** 设置、显示文件创建时的权限掩码。

**【格式】** umask [选项] [掩码]

**【选项】**

**-S**       以字符形式显示掩码对应的权限。

**【说明】** 若指定了掩码，则将该掩码作为新建文件的权限掩码。若未指定掩码，则显示现在的权限掩码。

## 例： umask命令的用法示例：

**\$ umask**                      **#显示当前掩码**

**\$ umask -S 007**            **#重新设置掩码为007，用字符方式显示**

## 3.3.5 系统信息显示

Linux系统下信息显示命令，这些命令有uname, hostname, dmesg, cal以及date等。

## uname命令

使用uname命令可以显示计算机以及操作系统的相关信息。

命令语法：

uname [-amnrsv]

- m: 显示计算机硬件类型
- n: 显示在网络上的主机名称
- r: 显示操作系统的内核版本
- s: 显示操作系统名称
- v: 显示操作系统的版本
- a: 显示全部信息

```
[administrator@NeoKvlin-CNVL 桌面]$ uname -a  
Linux NeoKvlin-CNVL.cs1cloud.internal 3.12.11-201.nk.1.x86_64 #1 SMP Sun Jun 8 2  
0:11:46 CST 2014 x86_64 x86_64 x86_64 GNU/Linux  
[administrator@NeoKvlin-CNVL 桌面]$
```

# hostname命令

使用hostname命令可以显示或修改计算机的主机名。

命令语法：

hostname [计算机名]

【例】 显示当前计算机主机名。

```
[administrator@NeoKylin-CNVL 桌面]$  
[administrator@NeoKylin-CNVL 桌面]$ hostname  
NeoKylin-CNVL.cs1cloud.internal  
[administrator@NeoKylin-CNVL 桌面]$
```

# dmesg命令

使用dmesg命令可以显示开机信息。

- dmesg用来显示内核环缓冲区（kernel-ring buffer）内容，内核将各种消息存放在这里。
- 在系统引导时，内核将与硬件和模块初始化相关的信息填到这个缓冲区中。
- 内核环缓冲区中的消息对于诊断系统问题通常非常有用。在运行dmesg时，它显示大量信息。通常通过less或grep使用管道查看dmesg的输出，这样可以更容易找到待查信息。

# cal命令

使用cal命令可以显示计算机系统的日历。

命令语法：

cal [选项] [月 [年]]



# date命令

使用date命令可以显示和设置计算机系统的日期和时间。

命令语法：

date [参数] [显示时间格式]（以+开头，后面接格式）

## 3.3.6 信息交流

Linux系统下信息交流命令，这些命令有echo，msg， wall以及write等。

# echo命令

使用echo命令可以在计算机显示器上显示一段文字，一般起到一个提示的作用。

命令语法：

echo [ -n ] [字符串]

## mesg命令

使用mesg命令可以设置是否允许其他用户用write命令给自己发送信息。

命令语法：

mesg [y|n]

【例】 显示当前计算机是否允许其他用户给自己发送信息的状态。

【例】 禁止其他用户发送信息到当前终端。

【例】 允许其他用户发送信息到当前终端。

```
[administrator@NeoKylin-CNVL 桌面]$ mesg  
is y  
[administrator@NeoKylin-CNVL 桌面]$ mesg n  
[administrator@NeoKylin-CNVL 桌面]$ mesg  
is n  
[administrator@NeoKylin-CNVL 桌面]$ mesg y  
[administrator@NeoKylin-CNVL 桌面]$ nesg  
bash: nesg: 未找到命令...  
[administrator@NeoKylin-CNVL 桌面]$ mesg  
is y  
[administrator@NeoKylin-CNVL 桌面]$
```

## wall命令

使用wall命令可以对全部已登录的用户发送信息。

命令语法：`wall [消息]`

【例】向所有用户发出“hello”的信息。

```
[administrator@NeoKylin-CNVL 桌面]$ wall 'hello'
[administrator@NeoKylin-CNVL 桌面]$
Broadcast message from administrator@NeoKylin-CNVL.cs1cloud.internal (pts/1) (Thu Mar 31 10:35:43 2016):

hello
```

# write命令

使用write命令可以向同时登录的用户发送消息。

命令语法：

**write** [用户账号] [终端名称]

**【例】** 向tty3终端上的root用户发送信息。

```
#write root tty3
```

```
hello
```

**Ctrl+c**退出发送状态

## 3.3.7 其他命令

Linux系统下的其他命令，这些命令有clear，sync，uptime以及last等。



# clear命令

使用clear命令可以清除屏幕上的信息，类似于Windows系统命令行中的cls命令。

命令语法：

```
clear
```

# sync命令

使用sync命令可以更新inode表，并将缓冲文件写到硬盘中。

命令语法：

sync

# uptime命令

使用uptime命令可以显示系统已经运行了多长时间，它依次显示下列信息：现在时间、系统已经运行了多长时间、目前有多少登录用户、系统在过去的1min, 5min和15min内的平均负载。

命令语法：

uptime [-V]

【例】 显示系统运行时间。

```
11:10:42 up 4 days, 19:23,  2 users,  load average: 3.17, 3.45, 3.65
```

# last命令

使用last命令可以显示用户最后登录的信息。

命令语法：

last [选项]

- x: 显示系统关闭、用户登录和退出的历史
- n: 指定输出记录的条数
- i: 只显示指定的IP上登录的情况

## 例：显示root最近的登录记录：

```
$ last root
```

```
root tty1 Mon Aug 13 09:39 still logged in
```

```
root tty2 Sun Aug 12 09:29 - down (23:25)
```

```
root tty2 Fri Aug 10 18:17 - down (1+20:15)
```

```
root pts/0 Sun Aug 5 10:14 - 10:19 (00:04)
```

```
root tty1 Sun Aug 5 08:37 - 08:46 (00:08)
```

```
...
```

```
$
```

last命令的输出中，退出时间为“down”表示系统关机时间。

## w命令

**【功能】** 显示目前登录系统的用户以及他们正在执行的程序。

**【格式】** w [-s] [用户名]

**【说明】** -s选项表示以短格式显示，否则以长格式显示。短格式的输出格式为如下5列：  
用户名 登录的终端名 登录的远程主机名 空闲时间  
正执行的命令

## 例： 显示所有用户的登录与活动情况：

```
# w -s
```

```
22:19:30 up 3:59, 3 users, load average: 0.12, 0.09,  
0.23
```

USER	TTY	FROM	IDLE	WHAT
root	tty1	-	0.00s	w -s
jhm	tty7	:0	3.00s	usr/bin/gnome-session
jhm	pts/1	:0.0	7.24s	gnome-terminal

```
#
```

命令的输出表示，root在本地tty1控制台登录，正在执行w -s命令；jhm在本地tty7控制台登录，运行的是gnome会话程序；同时jhm还在gnome虚拟终端登录，运行的是gnome终端程序。

## 小 结

虽然目前Linux图形界面的使用已经相当方便，但是有些操作还是需要在传统的文字界面下使用比较灵活。除此之外，使用文字界面登录Linux系统，系统资源的损耗也比较少，从而可以提高系统性能。



## 小 结

在Linux系统下**文本显示和处理**的命令主要有cat, more, less, head, tail, sort, uniq, cut, comm以及diff等。

在Linux系统下**文件和命令查找**的命令主要有grep, find, locate, whereis, file, whatis以及which等。

## 小 结

在Linux系统下**信息显示**的命令主要有uname, hostname, dmesg, cal以及date等。

在Linux系统下**信息交流**的命令主要有echo, mesg, wall以及write等。

```
[jhm@cilinux ~]$ cd ./project/hello  
[jhm@cilinux hello]$ pwd  
/home/jhm/project/hello  
[jhm@cilinux hello]$
```

```
[jhm@cilinux hello]$ cd ../../book  
[jhm@cilinux book]$ pwd  
/home/jhm/book  
[jhm@cilinux book]$
```

```
[jhm@cilinux ~]$ cat doc1  
To see a world in a grain of sand,  
And a heaven in a wild flower,  
Hold infinity in the palm of your hand,  
And eternity in an hour.
```

```
[jhm@cilinux ~]$ cat -n doc1  
1 To see a world in a grain of sand,  
2 And a heaven in a wild flower,  
3 Hold infinity in the palm of your hand,  
4 And eternity in an hour.
```

```
[jhm@cilinux ~]$ ls -ld temp
drwxr--r--  2 jhm jhm 4096  3月  2 07:25 temp
[jhm@cilinux ~]$ mkdir -p ./backup/version1
[jhm@cilinux ~]$ ls backup
version1
[jhm@cilinux ~]$
```

```
[jhm@cilinux hello]$ ls
file hello.c hello.h hello.o hello.save makefile
[jhm@cilinux hello]$ cat hello.h
ni
[jhm@cilinux hello]$ cat hello.save
wo
[jhm@cilinux hello]$ cp hello.h hello.save
[jhm@cilinux hello]$ ls
file hello.c hello.h hello.o hello.save makefile
[jhm@cilinux hello]$ cat hello.h
ni
[jhm@cilinux hello]$ cat hello.save
ni
[jhm@cilinux hello]$
```

文件名未变

```
[jhm@cilinux hello]$ ls
file hello.c hello.h hello.o hello.save makefile
[jhm@cilinux hello]$ cp hello.h list
[jhm@cilinux hello]$ ls
file hello.c hello.h hello.o hello.save list makefile
[jhm@cilinux hello]$ cat list
ni
[jhm@cilinux hello]$
```

---

```
[jhm@cilinux ~]$ ls
backup book memo myfirst project wenjian
[jhm@cilinux ~]$ cd project
[jhm@cilinux project]$ ls
file hello hello.c hoc
[jhm@cilinux project]$
```

---

```
[jhm@cilinux ~]$ cp -v myfirst project
'myfirst' -> 'project/myfirst'
[jhm@cilinux ~]$ cd project
[jhm@cilinux project]$ ls
file hello hello.c hoc myfirst
[jhm@cilinux project]$
```

---

```
[jhm@cilinux ~]$ cd wenjian
[jhm@cilinux wenjian]$ ls
bin lib list
[jhm@cilinux wenjian]$ cp -v list ../book
'list' -> '../book/list'
[jhm@cilinux wenjian]$ cd ../book
[jhm@cilinux book]$ ls
doc1 doc2 include.c.gz init.c.gz list math.c
[jhm@cilinux book]$
```





```
[jhm@cilinux backup]$ ls
bak  print.c  print.h  solo.c  version1
[jhm@cilinux backup]$ mkdir newdir
[jhm@cilinux backup]$ cp *.c newdir
[jhm@cilinux backup]$ ls newdir
print.c  solo.c
[jhm@cilinux backup]$
```

---

```
[jhm@cilinux hello]$ ls
file hello.c hello.o hello.save list makefile temp
[jhm@cilinux hello]$ mv -v hello.o hello.save temp
'hello.o' -> 'temp/hello.o'
'hello.save' -> 'temp/hello.save'
[jhm@cilinux hello]$ ls temp
hello.o hello.save
[jhm@cilinux hello]$ ls
file hello.c list makefile temp
[jhm@cilinux hello]$
```

```
[jhm@cilinux hello]$ cd temp
[jhm@cilinux temp]$ ls
hello.o hello.save
[jhm@cilinux temp]$ mv -v hello.o hello.save ../../hello
'hello.o' -> '../../hello/hello.o'
'hello.save' -> '../../hello/hello.save'
[jhm@cilinux temp]$ ls
[jhm@cilinux temp]$ cd ..
[jhm@cilinux hello]$ ls
file hello.c hello.o hello.save list makefile temp
[jhm@cilinux hello]$
```



```
[jhm@cilinux hello]$ mv hello.h list  
[jhm@cilinux hello]$ ls  
file  hello.c  hello.o  hello.save  list  makefile  
[jhm@cilinux hello]$
```

**hello.h移走了**

```
[jhm@cilinux ~]$ cat doc1
To see a world in a grain of sand,
And a heaven in a wild flower,
Hold infinity in the palm of your hand,
And eternity in an hour.
[jhm@cilinux ~]$ sort doc1
And a heaven in a wild flower,
And eternity in an hour.
Hold infinity in the palm of your hand,
To see a world in a grain of sand,
[jhm@cilinux ~]$ █
```

```
[jhm@cilinux hoc]$ ls  
a.out  look  temp  word.c  word.o  
[jhm@cilinux hoc]$
```

```
[jhm@cilinux hoc]$ ls temp  
word.save  
[jhm@cilinux hoc]$
```

```
[jhm@cilinux hoc]$ find -name 'word*' -print  
./word.c  
./temp/word.save  
./word.o  
[jhm@cilinux hoc]$
```