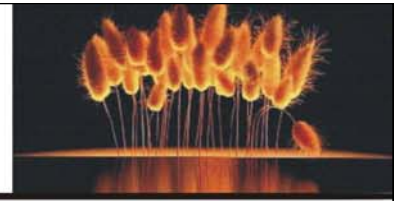




- 一. 第一章: Linux操作系统简介
- 二. 第二章: Linux操作系统安装和基本配置
- 三. 第三章: Linux操作系统的文件系统结构
- 四. 第四章: Linux操作系统的用户管理
- 五. 第五章: Linux操作系统常用命令详解
- 六. 第六章: Linux操作系统的进程管理



第一章：Linux操作系统简介





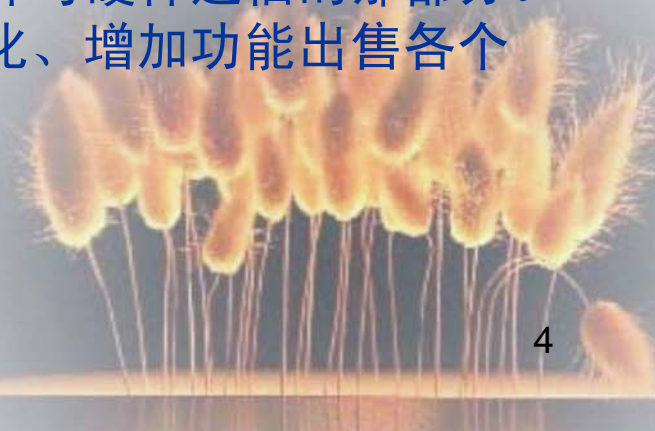
- Linux 以它的高效性和灵活性著称。具有多任务、多用户的能力。
- Linux 之所以受到广大计算机爱好者的喜爱，其主要原因有两个，一是它属于自由软件，用户不用支付任何费用就可以获得它和它的源代码，并且可以根据自己的需要对它进行必要的修改。另一个原因是，它具有 Unix 的全部功能，任何使用 Unix 操作系统或想要学习 Unix 操作系统的人都可以从 Linux 中获益。



1.1 Linux操作系统起源



- Linux是一个诞生于网络、成长于网络且成熟于网络的操作系统。
- 1991年，芬兰大学生Linus Torvalds萌发了开发一个自由的UNIX操作系统的想法，当年Linux诞生，为了不让这个羽毛未丰的操作系统夭折，Linus将自己的作品Linux通过Internet发布。从此一大批知名的、不知名的hack、编程人员加入到开发过程中来，Linux逐渐成长起来。
- Linux一开始是要求所有的源码必须公开，并且任何人均不得从Linux交易中获利。然而这种纯粹的自由软件的理想对于Linux的普及和发展是不利的，于是Linux开始转向GPL，成为GNU阵营中的主要一员。
- Linux凭借优秀的设计，不凡的性能，加上IBM、INTEL、CA、ORACLE等国际知名企业的大力支持，市场份额逐步扩大，逐渐成为主流操作系统之一。
- Linux只是内核，即操作系统中允许用户的软件与硬件通信的那部分。
- Linux产商借网络爱好者升级的内核，通过优化、增加功能出售各个版本的linux操作系统

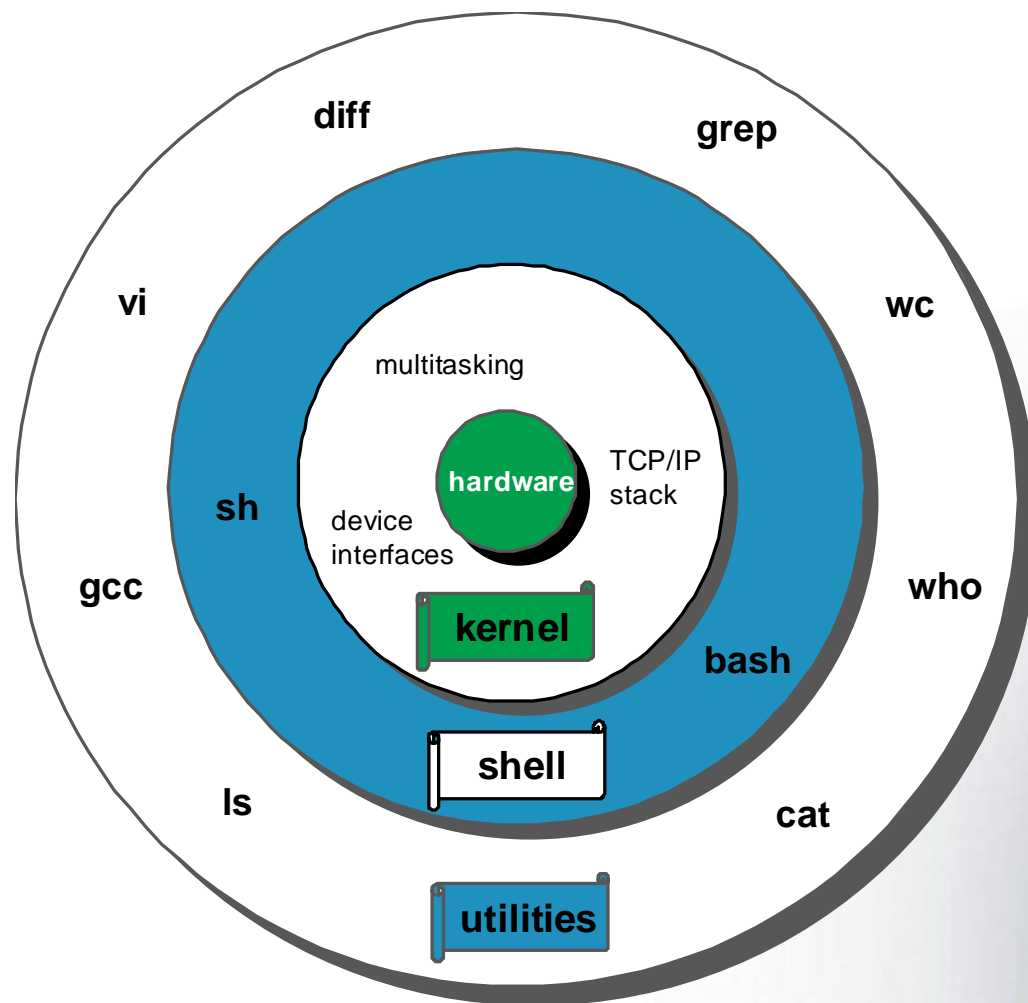


1.2 Linux操作系统特点



- 一. **开放性**: 指系统遵循世界标准规范, 特别是遵循开放系统互连 (OSI) 国际标准。
- 二. **多用户**: 是指系统资源可以被不同用户使用, 每个用户对自己的资源 (例如: 文件、设备) 有特定的权限, 互不影响。
- 三. **多任务**: 它是指计算机同时执行多个程序, 而且各个程序的运行互相独立。
- 四. **良好的用户界面**: Linux向用户提供了两种界面: 用户界面和系统调用。Linux还为用户提供了图形用户界面。它利用鼠标、菜单、窗口、滚动条等设施, 给用户呈现一个直观、易操作、交互性强的友好的图形化界面
- 五. **设备独立性**: 是指操作系统把所有外部设备统一当作成文件来看待, 只要安装它们的驱动程序, 任何用户都可以象使用文件一样, 操纵、使用这些设备, 而不必知道它们的具体存在形式。Linux是具有设备独立性的操作系统, 它的内核具有高度适应能力。
- 六. **提供了丰富的网络功能**: 完善的内置网络是Linux一大特点。
- 七. **可靠的安全系统**: Linux采取了许多安全技术措施, 包括对读、写控制、带保护的子系统、审计跟踪、核心授权等, 这为网络多用户环境中的用户提供了必要的安全保障。
- 八. **良好的可移植性**: 是指将操作系统从一个平台转移到另一个平台使它仍然能按其自身的方式运行的能力。Linux是一种可移植的操作系统, 能够在从微型计算机到大型计算机的任何环境中和任何平台上运行。诺基亚的Maemo、谷歌的Android和Chrome OS, 以及英特尔的Moblin。

1.3 Linux操作系统结构



Kernel

系统启动时将内核装入内存
管理系统各种资源

Shell

用户界面，提供用户与内核
交互处理接口是命令解释
器，提供强大的编程环境

bash, ash, pdksh, tcsh, ksh,
csh, zsh...

Utility

提供各种管理工具，应用程
序



版本号码

稳定版本
2. 4. 32

开发版本
2. 5. 19-6

主版本号. 稳定(偶) / 开发版本 (奇) . 发布号-patch号

当前最新版本 [linux-2.6.31.tar.gz](http://www.kernel.org/pub/linux/kernel/linux-2.6.31.tar.gz)

<http://www.kernel.org/pub/linux/kernel/>

1.5 Linux操作系统发行版本



fedora^f



CentOS



debian



gentoo linux™

还有许多……



第二章：Linux操作系统安装和基本配置

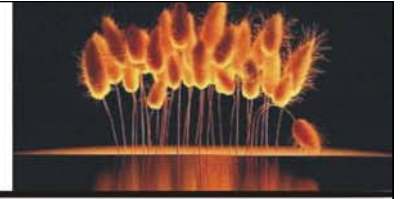




■ RPM

■ 从源码编译安装





第三章：Linux系统的文件系统及其结构

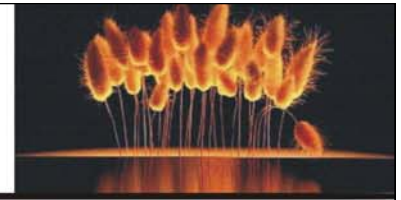


3.1 文件与目录的基本概念



- **文件：**是用来存储信息的基本结构，它是被命名（文件名）的存储在某种介质（磁盘，光盘，磁带等）上的一组信息的集合。
- **从技术上讲，文件不能存贮任何数据，它只是一个用来指向它们相应的索引节点（inode）的名字，索引节点包含了文件的真正信息。**
- **文件名：**是文件的标识，由字母，数字，下划线，圆点组成。
- **扩展名：**为了便于识别和管理，用扩展名作为文件名的一部分，中间用. 隔开
- **目录文件：**用来管理和组织大量的文件，常称为目录

3.2 文件系统基本概念



- 文件系统是包括在一个磁盘（包括光盘、软盘、闪盘及其它存储设备）或分区的目录结构；一个可应用的磁盘设备可以包含一个或多个文件系统；如果您想进入一个文件系统，首先您要做的是挂载（mount）文件系统；为了挂载（mount）文件系统，您必须指定一个挂载点；
- 文件系统是在一个磁盘（包括光盘、软盘、闪盘及其它存储设备）或分区组织文件的方法，如NTFS或FAT；
- 文件系统是文件的数据结构或组织方法。在Linux中，文件系统涉及两个非常独特的事情，目录树或在磁盘或分区上文件的排列；文件系统是基于操作系统的，建立在磁盘媒质上的可见体系结构，例如这种结构对于一个Linux用户来说可以用ls 或其它工具可以看到；
- 文件系统是基于被划分的存储设备上的逻辑上单位上的一种定义文件的命名、存储、组织及取出的方法；
- 在计算机业，一个文件系统是有组织存储文件或数据的方法，目的是易于查询和存取。文件系统是基于一个存储设备，比如硬盘或光盘，并且包含文件物理位置的维护；也可以说文件系统也是虚拟数据或网络数据存储的方法，比如NFS。

3.3 Linux文件系统类型



■ ext2 文件系统

ext2文件系统应该说是Linux正宗的文件系统，早期的Linux都是用ext2，但随着技术的发展，大多Linux的发行版本目前并不用这个文件系统了；比如Redhat和Fedora 大多都建议用ext3，ext3文件系统是由ext2发展而来的。对于Linux新手，我们还是建议您不要用ext2文件系统；ext2支持undelete（反删除），如果您误删除文件，有时是可以恢复的，但操作上比较麻烦；

■ ext3 文件系统：是由ext2文件系统发展而来

ext3 is a Journalizing file system for Linux（ext3是一个用于Linux的日志文件系统），ext3支持大文件；但不支持反删除（undelete）操作；Redhat和Fedora都力挺ext3；

■ reiserfs 文件系统

reiserfs 文件系统是一款优秀的文件系统，支持大文件，支持反删除（undelete）；操作反删除比较容易；reiserfs 支持大文件；

3.4 常用文件系统比较



Filesystem	File Size Limit	Filesystem Size Limit
ext2/ext3 with 1 KiB blocksize	16448 MiB (~ 16 GiB)	2048 GiB (= 2 TiB)
ext2/3 with 2 KiB blocksize	256 GiB	8192 GiB (= 8 TiB)
ext2/3 with 4 KiB blocksize	2048 GiB (= 2 TiB)	8192 GiB (= 8 TiB)
ext2/3 with 8 KiB blocksize (Systems with 8 KiB pages like Alpha only)	65568 GiB (~ 64 TiB)	32768 GiB (= 32 TiB)
ReiserFS 3.5	2 GiB	16384 GiB (= 16 TiB)
ReiserFS 3.6 (as in Linux 2.4)	1 EiB	16384 GiB (= 16 TiB)
XFS	8 EiB	8 EiB
JFS with 512 Bytes blocksize	8 EiB	512 TiB
JFS with 4KiB blocksize	8 EiB	4 PiB
NFSv2 (client side)	2 GiB	8 EiB
NFSv3 (client side)	8 EiB	8 EiB

1024 Bytes = 1 KB; 1024 KB = 1 MB; 1024 MB = 1 GB; 1024 GB = 1 TB; 1024 TB = 1 PB; 1024 PB = 1 EiB

3.5 文件系统的创建



- 文件系统的创建

这个过程是存储设备建立文件系统的过程，一般也被称为格式化或初始化，通过一些初始化工具来进行。一般的情况下每个类型的操作系统都有这方面的工具。在Linux中有mkfs系列工具

- 创建方法

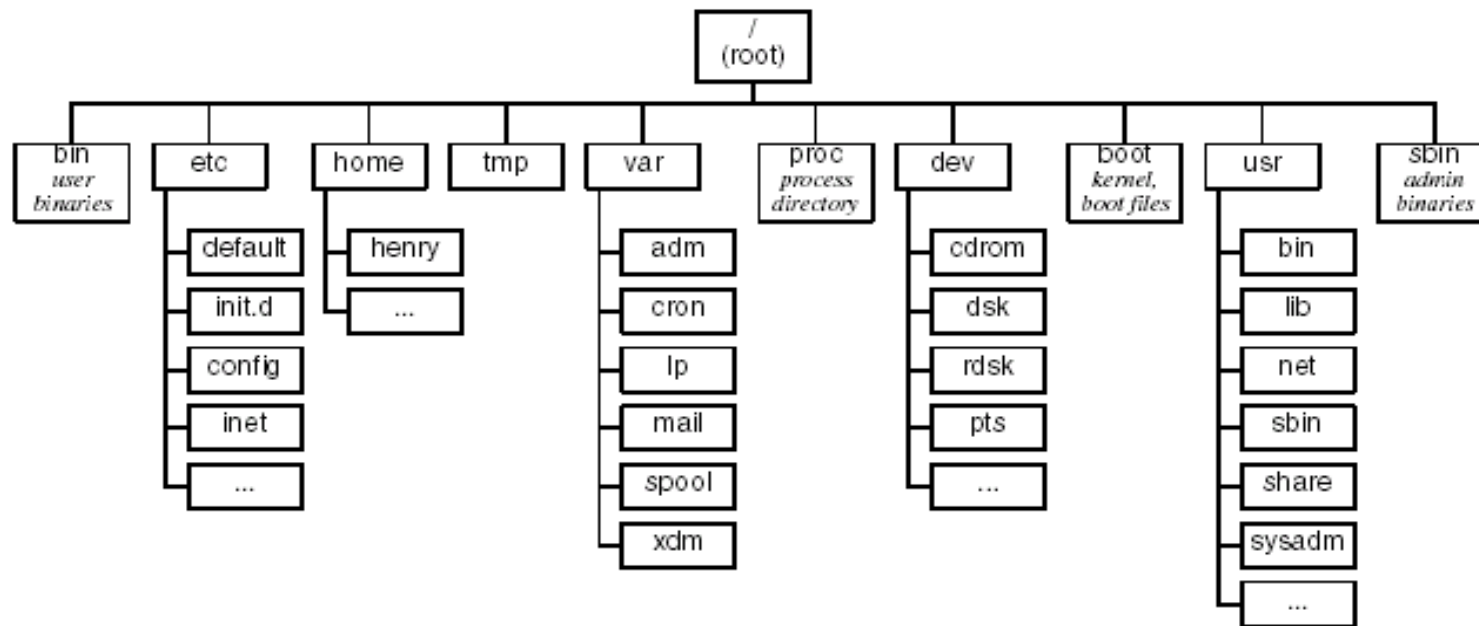
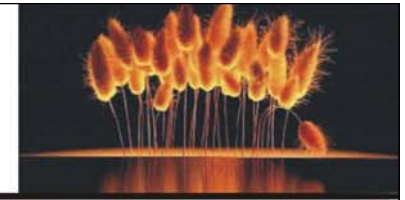
以系统第二块硬盘为例： /dev/sdb

分区： #fdisk /dev/sdb

分区完毕后，硬盘识别为 /dev/sdb1

文件系统创建： #mkfs -t 文件系统类型 /dev/sdb1

3.6 Linux系统的文件结构



3.6 Linux系统的文件结构



/	Linux系统根目录，包含所有目录
/bin	Binary的缩写，存放用户的可执行程序，例如ls, cp, 也包含其它的SHELL如：bash等
/boot	包含vmlinuz, initrd. img等启动文件，随便改动可能无法正常开机
/dev	接口设备文件目录，如你的硬盘：sda
/etc	Passwd等系统设置与管理的文件
/etc/x11	X Windows System的设置目录
/home	一般用户的主目录
/lib (/lib64)	包含执行/bin和/sbin目录的二进制文件时所需的共享函数库library
/mnt	各项装置的文件系统加载点，例如：/mnt/cdrom是光驱的加载点
/opt	提供空间，较大的且固定的应用程序存储文件之用
/proc	PS命令查询的信息与这里的相同，都是系统内核与程序执行的信息
/root	管理员的主目录

3.6 Linux系统的文件结构



/sbin	系统启动时所需的二进制程序
/tmp	Temporary, 存放暂存盘的目录
/usr	存放用户使用系统命令和应用程序等信息
/usr/bin	存放用户可执行程序, 如grep, mdir等
/usr/doc	存放各式程序文件的目录
/usr/include	保存提供C语言加载的header文件
/usr/include/X11	保存提供X Windows程序加载的header文件
/usr/info	GNU程序文件目录
/usr/lib (/lib64)	函数库
/usr/lib(/lib64)/X11	函数库
/usr/local	提供自行安装的应用程序位置
/usr/man	存放在线说明文件目录
/usr/sbin	存放经常使用的程序, 如showmount
/usr/src	保存系统的源码文件
/usr/X11R6/bin	存放X Windows System的执行程序
/var	Variable, 具有变动性质的相关程序目录, 如log

3.6.1 Linux文件类型的定义



普通文件

■ 文本文件：ASCII码形式存储

- 开头，如：`-rw-r--r-- 1 root root 39599 Mar 8 12:15 x`

■ 二进制文件：以二进制形式存储在计算机中，不可直接读，要通过相应的软件读取

- 开头，如：`-rwxrwxrwx 1 root root 46888960 Dec 9 2005 x.sh`

目录文件：d字母开头

如：`drwxr-xr-x 2 root root 4096 Aug 2 2006 bin`

三. 设备文件

■ 块设备文件：b字母开头

如：`brw-rw---- 1 root disk 3, 1 Jan 30 2003 hda1`

■ 字符设备文件：c字母开头

如：`crw----- 1 root root 4, 1 Jul 31 13:49 ttyl`

3.6.2 Linux文件属性的定义



```
[root@localhost ~]# ls -lih
总计 104K
2408830 drwxr-xr-x 2 root root 4.0K 04-21 12:46 mkuml-2004.07.17
2408260 drwxr-xr-x 2 root root 4.0K 04-21 22:15 mydir
2408258 lrwxrwxrwx 1 root root 7 04-21 22:16 sun001.txt -> sun.txt
2408263 -rw-r--r-- 2 root root 39K 04-20 14:17 sun.txt
```

解释：

第一字段：inode；

inode：索引节点。每个存储设备或存储设备的分区（存储设备是硬盘、软盘、U盘）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。

第二字段：文件种类和权限；

第三字段：硬链接个数；

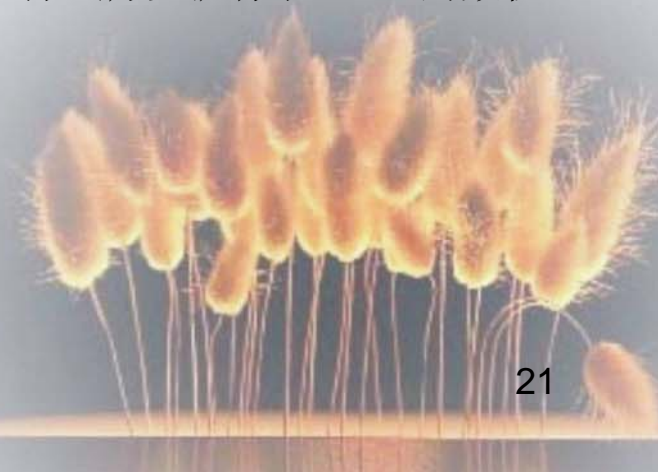
第四字段：属主；

第五字段：所归属的组；

第六字段：文件或目录的大小；

第七字段和第八字段：最后访问或修改时间；

第九字段：文件名或目录名



3.6.3 Linux文件权限的定义



```
[root@zhz home]# ls -l
total 32
drwxrwxrwx    38  down      root          4096 Jul  5 19:09 down
drwx-----    9   glh       glh           4096 Mar 26 19:08 glh
drwx-----    4   lei       lei           4096 Mar 21 08:40 lei
drwxr-xr-x    5   root      root          4096 Apr 12 2006 software
```

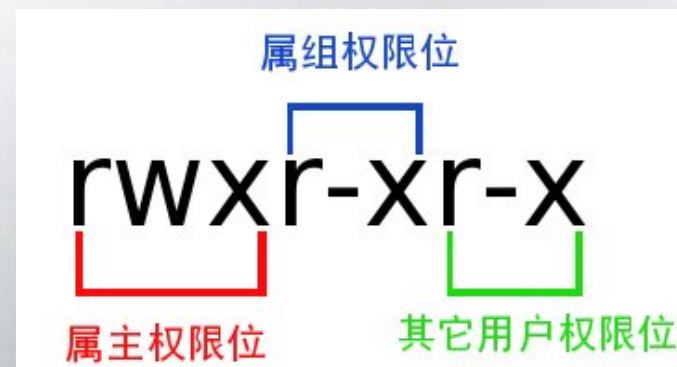
- Linux文件或目录的权限位是由 9 个权限位来控制，每三位为一组，它们分别是：

文件属主 (Owner) 的读r、写w、执行x

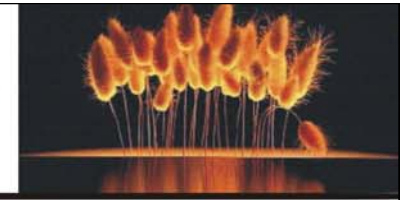
用户组 (Group) 的读r、写w、执行x

(Other) 其它用户的读r、写w、执行x；

如果权限位不可读、不可写、不可执行，是用-来表示。



3.6.4 Linux文件权限的修改



- 文件/目录权限设置命令 `chmod`
 - `chmod +x filename` 改变文件属性为可执行
 - `chmod 777 filename` 改变文件属性为rwx-rwx-rwx





第四章：Linux操作系统的用户管理





理解Linux的单用户多任务，多用户多任务概念

■ Linux 的单用户多任务

单用户多任务；一个用户，为了完成工作，执行了多个任务；当然其它的人还能以远程登录过来，也能做其它的工作。

■ Linux 的多用户、多任务

有时可能是很多用户同时用同一个系统，但并不所有的用户都一定都要做同一件事，所以这就有多用户多任务之说；

举个例子，比如一台服务器，上面有FTP用户、系统管理员、web用户、常规普通用户等，在同一时刻，有的可能在上传软件包管理子站；在与此同时，可能还会有系统管理员在维护系统；不同用户所具有的权限也不同，要完成不同的任务得需要不同的用户，也可以说不同的用户，可能完成的工作也不一样；

值得注意的是：多用户多任务并不是大家同时挤到一起在一台机器的键盘和显示器前来操作机器，多用户可能通过远程登录来进行，比如对服务器的远程控制，只要有用户权限任何人都是可以上去操作或访问的；



第五章：Linux操作系统常用命令详解

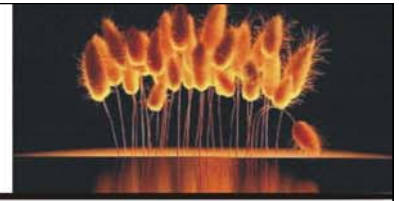




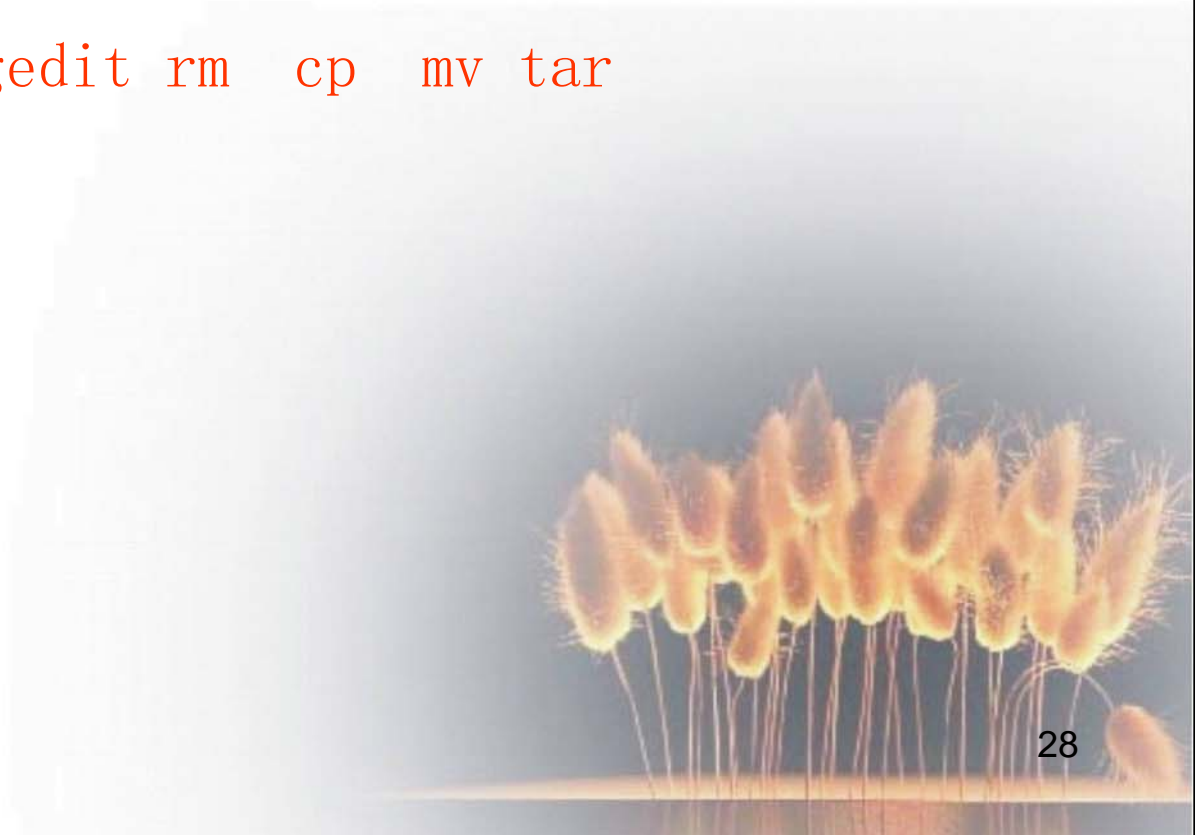
- 文件目录类命令
- 系统信息类命令
- 通信网络类命令



5.1 文件目录类命令



- 浏览目录命令: `ls` `cd` `dir`
- 浏览文件命令: `cat`
- 目录操作命令: `mkdir` `rmdir`
- 文件操作命令: `vi` `gedit` `rm` `cp` `mv` `tar`





一. `ls(list)` 是一个非常有用的命令，用来显示当前目录下的内容。配合参数的使用，能以不同的方式显示目录内容。下面是一些常用的范例。

二. 格式: `ls[参数] [路径或文件名]`

```
[root@linux root]# ls
anaconda-ks.cfg  install.log  install.log.syslog
[root@linux root]# ls -a
.                .bash_logout .cshrc          install.log.syslog .Xresources
..               .bash_profile .gtkrc          .kde
anaconda-ks.cfg  .bashrc      install.log     .tcshrc
[root@linux root]# ls -l
total 32
-rw-r--r--    1 root    root        3811 Mar 10  2005 anaconda-ks.cfg
-rw-r--r--    1 root    root       24636 Mar 10  2005 install.log
-rw-r--r--    1 root    root         0 Mar  9  2005 install.log.syslog
[root@linux root]# _
```

5.1.2 切换目录的cd命令



cd (change directory) 命令可以让用户切换当前目录。

范例如下：

[test@linux home]\$ cd test 切换到当前目录下的test子目录

[test@linux test]\$ cd .. 切换到上一层目录

[test@linux home]\$ cd / 切换到系统根目录

[test@linux /]\$ cd 切换到用户自家目录（或执行cd~）

[test@linux test]\$ cd /usr/bin 切换到/usr/bin目录





一. mkdir(make directory)命令可用来创建子目录

格式: mkdir [参数] <目录名>

[test@linux test]\$ mkdir dir 在当前目录下建立dir目录

[test@linux test]\$ mkdir -p dir1/dir2

在当前目录下创建dir1目录,并在dir1目录下创建dir2目录,也就是连续创建两个目录 (dir1/和dir1/dir2)

二. rmdir(remove directory)命令可用来删除“空”的子目录

格式: rmdir [参数]<目录名>

[test@linux test]\$ rmdir dir 删除“空”的子目录dir

[test@linux test]\$ rmdir -p dir1/dir2 删除dir1下的dir2目录,若dir1目录为空也删除它



cp(copy)命令可以将文件从一处复制到另一处。一般在使用cp命令时将一个文件复制成另一个文件或复制到某目录时,需要指定源文件名与目标文件名或目录。

格式: cp[参数]<源文件路径><目标文件路径>

```
[test@linux test]$ cp test1.txt test2.txt
```

将test1.txt复制成test2.txt

```
[test@linux test]$ cp test3.txt /tmp
```

将test3.txt复制到/tmp目录中

```
[test@linux test]$ cp -r test1(目录) test2(目录)
```

加-r参数,拷贝目录



功能：删除文件或目录

格式：rm[参数] <目标文件路径>

[test@linux test]\$ rm myfiles 删除一个文件

[test@linux test]\$ rm *

删除当前目录下的所有文件

-f参数：强迫删除文件

[test@linux test]\$ rm -f *.txt

强迫删除所有以后缀名为txt文件

-r参数：删除目录



-i参数：删除文件时询问

```
[test@linux test]$ rm -i *
```

删除当前目录下的所有文件

rm:backup: is a directory 遇到目录会略过

rm : remove 'myfiles.txt' ? Y

删除文件时会询问, 可按Y或N键表示允许或拒绝删除文件

注意：在有些系统的默认状态下，rm命令会对每个删除的文件一一询问。如果用户确定要删除这些文件，则可以使用参数-f来避免询问。



功能：用于显示文件的内容, 也可以将数个文件合并成一个文件。

格式：cat[参数]<文件名>

[test@linux test]\$cat test.txt 显示test.txt文件内容

[test@linux test]\$cat test.txt | more

逐页显示test.txt文件中的内容

[test@linux test]\$cat test.txt >>test1.txt

将test.txt的内容附加到test1.txt文件之后

[test@linux test]\$cat test.txt test2.txt
>readme.txt

将test.txt和test2.txt文件合并成readme.txt文件

5.1.10 移动或更改文件、目录名称的mv命令



功能：可以将文件及目录移到另一目录下，或更改文件及目录的名称

格式：[参数] <源文件或目录> <目标文件或目录>

```
[test@linux dir1]$ mv a.txt ../
```

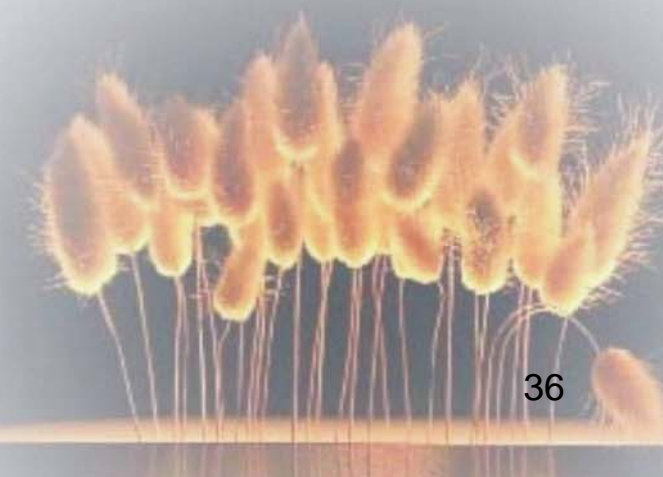
将a.txt文件移动上层目录

```
[test@linux dir1]$ mv a.txt b.txt
```

将a.txt改名为b.txt

```
[test@linux dir1]$ mv dir2 ../
```

将dir2目录上移一层



5.1.11 显示当前目录的pwd命令



功能：显示用户正在工作或当前所在的目录

格式：pwd

```
[test@linux test]$ pwd  
/home/test
```

显示用户test所在的当前目录是/home/test





Find功能：用来寻找文件或目录

格式：find [<路径>] [匹配条件]

```
[root@linux root]# find / -name httpd.conf
```

搜索系统根目录下名为httpd.conf的文件

/etc/httpd/httpd.conf

显示搜索结果

```
[root@linux root]# find /etc -name httpd.conf
```

搜索/etc目录下名为httpd.conf的文件，并显示结果

/etc/httpd/httpd.conf

显示搜索结果



功能：在文件中搜索匹配的字符并进行输出

格式：grep[参数] <要找的字串> <要寻找字串的源文件>

```
[root@linux root]# grep linux test.txt
```

搜索test.txt文件中字符串linux并输出

```
[root@linux root]# rpm -qa | grep httpd
```

搜索rpm包中含有httpd包的文件名

例：who | grep ttyl



功能：查看当前系统中有哪些用户登录

格式：who/w[参数]

```
[root@linux root]# who
```

```
root tty1
```

1个本地用户登录

```
test pts/0
```

1个远程登录用户

```
[root@linux root]# w
 5:42pm up 36 min, 2 users, load average: 0.08, 0.02, 0.06
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
root      tty1     -             5:16pm   0.00s   0.17s   0.01s   w
pp        tty2     -             5:42pm   3.00s   0.04s   0.04s   -bash
[root@linux root]# _
```

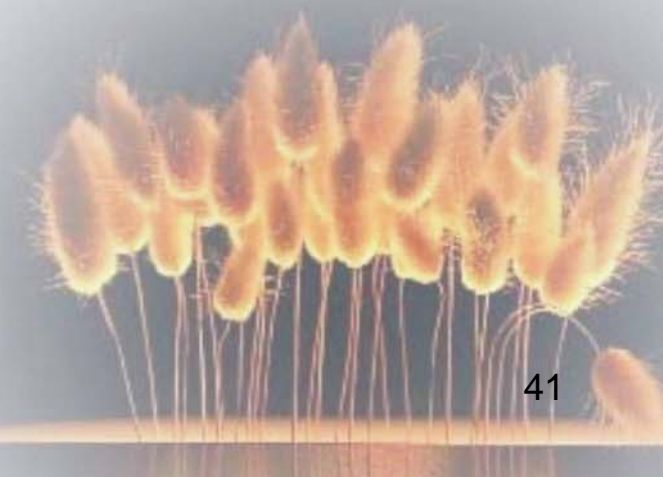


tar命令位于/bin目录下，它能够将用户所指定的文件或目录打包成一个文件，但不做压缩。一般Unix上常用的压缩方式是选用tar将许多文件打包成一个文件，再以gzip压缩命令压缩成xxx.tar.gz(或称为xxx.tgz)的文件。

注意：tar不仅可以打包文件，也可以将硬盘数据备份

常用参数：

- c：创建一个新tar文件
- v：显示运行过程的信息
- f：指定文件名
- z：调用gzip压缩命令进行压缩
- t：查看压缩文件的内容
- x：解开tar文件





```
[root@linux test]# tar -cvf test.tar *
```

将所有文件打包成test.tar, 扩展名.tar需自行加上

```
[root@linux test]# tar -zcvf test.tar.gz *
```

将所有文件打包成test.tar, 再用gzip命令压缩

```
[root@linux ljr]# tar -tf test.tar
```

查看test.tar文件中包括了哪些文件

```
[root@linux test]# tar -xvf test.tar
```

将test.tar解开

```
[root@linux test]# tar -zxvf foo.tar.gz
```

将foo.tar.gz解压缩



一. 进入vi的命令

vi filename :打开或新建文件，并将光标置于第一行首

vi +n filename : 打开文件，并将光标置于第n行首

vi + filename : 打开文件，并将光标置于最后一行首

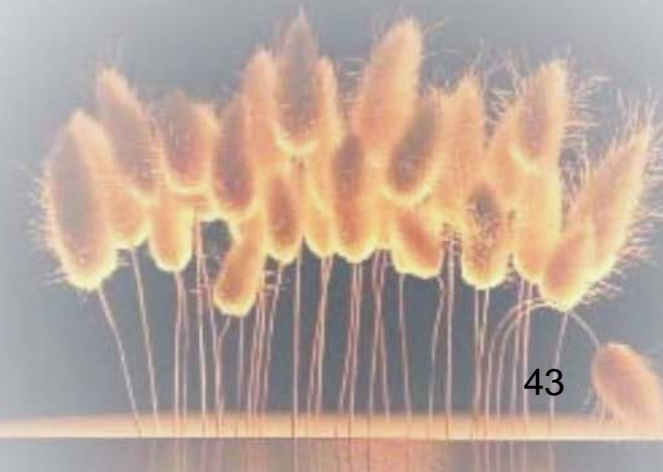
vi +/pattern filename: 打开文件，并将光标置于第一个与pattern匹配的串处

vi -r filename : 在上次正用vi编辑时发生系统崩溃，恢复filename

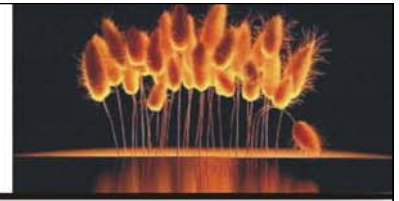
vi filename....filename : 打开多个文件，依次进行编辑

二. 进入gedit的命令

gedit filename :打开或新建文件



5.2 系统信息类命令



- dmesg命令
- df命令
- du命令
- free命令





功能：显示系统诊断信息、操作系统版本号、物理内存的大小以及其它信息

```
[root@host ~]# dmesg | more
Linux version 2.4.18-14 (bhcompile@stripples.devel.redhat.com) (gcc version 3.2
20020903 (Red Hat Linux 8.0 3.2-7)) #1 Wed Sep 4 13:35:50 EDT 2002
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 0000000000009f800 (usable)
 BIOS-e820: 0000000000009f800 - 000000000000a0000 (reserved)
 BIOS-e820: 000000000000ca000 - 000000000000cc000 (reserved)
 BIOS-e820: 000000000000dc000 - 000000000000e0000 (reserved)
 BIOS-e820: 000000000000e4000 - 00000000000100000 (reserved)
 BIOS-e820: 00000000000100000 - 000000000007ef0000 (usable)
 BIOS-e820: 000000000007ef0000 - 000000000007eff000 (ACPI data)
 BIOS-e820: 000000000007eff000 - 000000000007f00000 (ACPI NVS)
 BIOS-e820: 000000000007f00000 - 000000000008000000 (usable)
 BIOS-e820: 00000000000fec00000 - 00000000000fec10000 (reserved)
 BIOS-e820: 00000000000fee00000 - 00000000000fee01000 (reserved)
 BIOS-e820: 00000000000ffe00000 - 00000000000100000000 (reserved)
0MB HIGHMEM available.
128MB LOWMEM available.
On node 0 totalpages: 32768
zone(0): 4096 pages.
zone(1): 28672 pages.
zone(2): 0 pages.
Kernel command line: auto BOOT_IMAGE=linux ro BOOT_FILE=/boot/vmlinuz-2.4.18-14
root=LABEL=
```



功能：用于查看文件系统的各个分区的占用情况

```
[root@linux root]# df
Filesystem      1k-blocks      Used Available Use% Mounted on
/dev/sda1        806368      156924      608480   21% /
/dev/sda3        600864       16468      553876    3% /home
none             30736         0         30736    0% /dev/shm
/dev/sda2       2419288     1713288      583104   75% /usr
[root@linux root]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       787M  154M  594M  21% /
/dev/sda3       587M   17M  540M   3% /home
none            30M    0    30M   0% /dev/shm
/dev/sda2       2.3G  1.7G  569M  75% /usr
[root@linux root]# _
```



功能：查看某个目录中各级子目录所使用的硬盘空间数

格式：du [参数] <目录名>

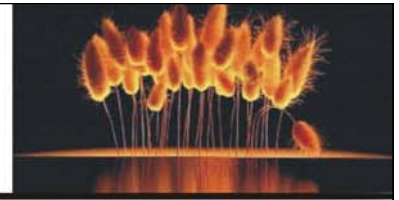
```
[root@linux root]# du
4      ../kde/share/config
4      ../kde/share/servicetypes
4      ../kde/share/mimelnk
4      ../kde/share/applnk
4      ../kde/share/services
24     ../kde/share
152    ../kde/tmp-localhost.localdomain
180    ../kde
252    .
[root@linux root]# du -h
4.0k   ../kde/share/config
4.0k   ../kde/share/servicetypes
4.0k   ../kde/share/mimelnk
4.0k   ../kde/share/applnk
4.0k   ../kde/share/services
24k    ../kde/share
152k   ../kde/tmp-localhost.localdomain
180k   ../kde
252k   .
[root@linux root]# _
```



功能：用于查看系统内存，虚拟内存（交换空间）的大小占用情况

```
[root@linux root]# free
              total        used        free      shared    buffers     cached
Mem:          61476       56264        5212          0        2528       29476
-/+ buffers/cache:      24260       37216
Swap:        128480         288       128192
[root@linux root]# free -m
              total        used        free      shared    buffers     cached
Mem:           60          54           5          0           2          28
-/+ buffers/cache:         23          36
Swap:          125           0          125
[root@linux root]# _
```


5.3 网络通讯类命令



- 一. `ifconfig`
- 二. `netstat` 查看网络的状况
- 三. `ping` 查询某个机器是否在工作





1. ifconfig 查看网络接口状态

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

```
[root@localhost ~]# ifconfig -a 查看主机所有网络接口的状态
```

```
[root@localhost ~]# ifconfig eth0 查看eth0 的状态
```

2. ifconfig 配置网络接口

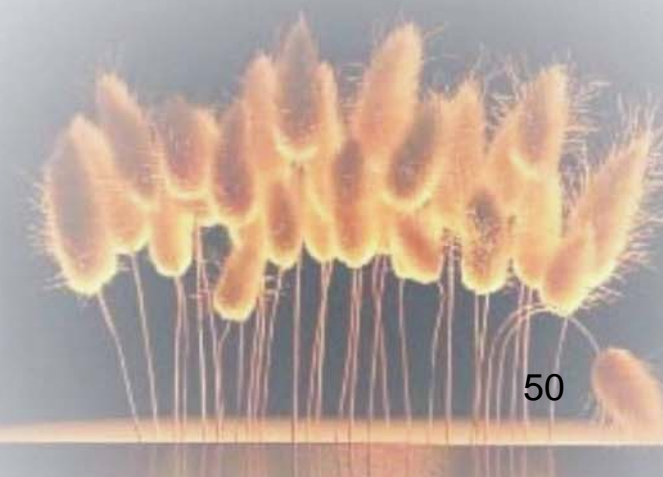
ifconfig 网络端口 IP地址 hw <HW> MAC地址 netmask 掩码地址 broadcast 广播地址 [up/down]

```
[root@localhost ~]# ifconfig eth0 down
```

```
[root@localhost ~]# ifconfig eth0 192.168.1.99 broadcast 192.168.1.255  
netmask 255.255.255.0
```

```
[root@localhost ~]# ifconfig eth0 up
```

```
[root@localhost ~]# ifconfig eth0
```





netstat

1. 作用

检查整个Linux网络状态。

2. 格式

netstat [-acCeFghilMnNoprstuvVwx] [-A] [--ip]

3. 主要参数

-a--all: 显示所有连线中的Socket。

-c--continuous: 持续列出网络状态。

-e--extend: 显示网络其它相关信息。

-g--groups: 显示多重广播功能群组组员名单。

-i--interfaces: 显示网络界面信息表单。

-l--listening: 显示监控中的服务器的Socket。

-n--numeric: 直接使用IP地址，而不通过域名服务器。

-r--route: 显示Routing Table。

-s--statistic: 显示网络工作信息统计表。

-t--tcp: 显示TCP传输协议的连线状况。

-u--udp: 显示UDP传输协议的连线状况。

-v--verbose: 显示指令执行过程。

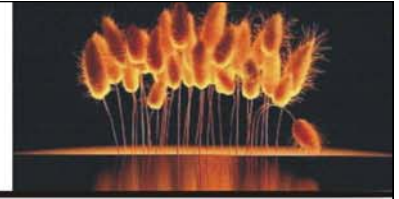
-w--raw: 显示RAW传输协议的连线状况。



ping [选项] 主机名/IP地址

命令中各选项的含义如下：

- c 数目 在发送指定数目的包后停止
- f 大量且快速地送网络封包给一台机器，看它的回应
- I 秒数 设定间隔几秒送一个网络封包给一台机器，预设值是一秒送一次
- l 次数 在指定次数内，以最快的方式送封包数据到指定机器（只有超级用户可以使用此选项）
- q 不显示任何传送封包的信息，只显示最后的结果
- r 不经由网关而直接送封包到一台机器，通常是查看本机的网络接口是否有问题
- s 字节数 指定发送的数据字节数，预设值是56，加上8字节的ICMP头，一共是64ICMP数据字节

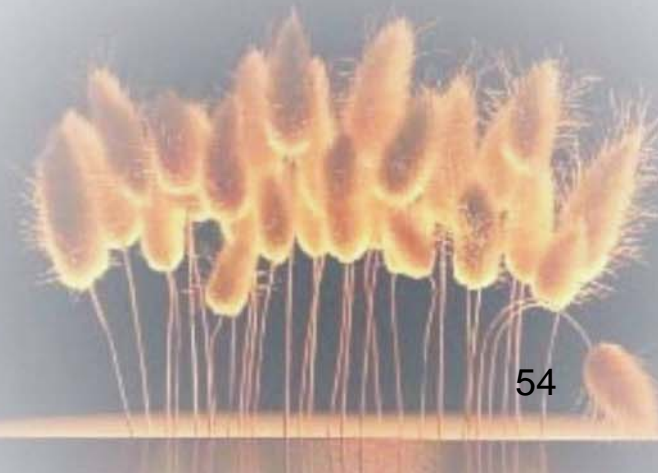


第六章：Linux操作系统的进程管理





- 一. 程序是为了完成某种任务而设计的软件，比如OpenOffice是程序。什么是进程呢？进程就是运行中的程序。
- 二. 一个运行着的程序，可能有多个进程。比如WWW服务器是apache服务器，当管理员启动服务后，可能会有好多人来访问，也就是说许多用户来同时请求httpd服务，apache服务器将会创建有多个httpd进程来对其进行服务。





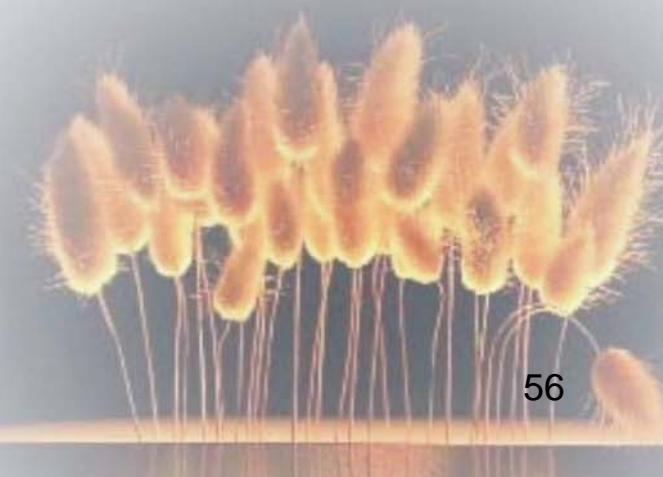
进程一般分为交互进程、批处理进程和守护进程三类。值得一提的守护进程总是活跃的，一般是后台运行，守护进程一般是由系统在开机时通过脚本自动激活启动或超级管理用户root来启动。比如在Fedora或Redhat中，我们可以定义httpd 服务器的启动脚本的运行级别，此文件位于/etc/init.d目录下，文件名是httpd，/etc/init.d/httpd就是httpd服务器的守护程序，当把它的运行级别设置为3和5时，当系统启动时，它会跟着启动。

```
[root@localhost ~]# chkconfig --level 35 httpd on
```

由于守护进程是一直运行着的，所以它所处的状态是等待请求处理任务。比如，是不是访问网站，网站的httpd服务器都在运行，等待着用户来访问，也就是等待着任务处理。

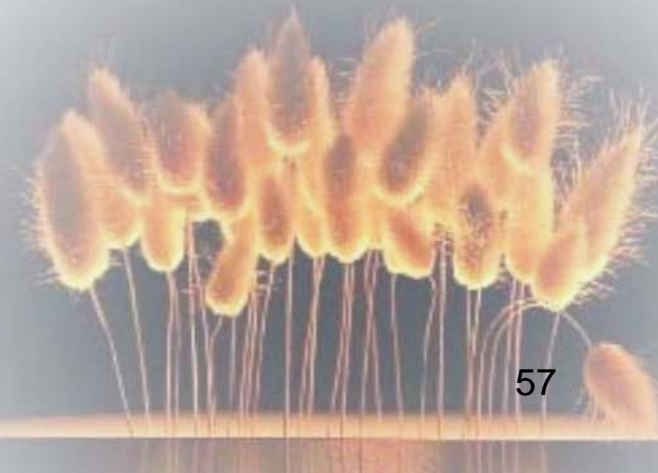


进程ID (PID): 是唯一的数值, 用来区分进程;
子进程和父进程的ID (PPID);
启动进程的用户ID (UID) 和所归属的组 (GID);
进程状态: 状态分为运行R、休眠S、僵尸Z;
进程执行的优先级;
进程所连接的终端名;
进程资源占用: 比如占用资源大小 (内存、CPU占用量)。





- 他们的关系是管理和被管理的关系，当父进程终止时，子进程也随之而终止。但子进程终止，父进程并不一定终止。比如httpd服务器运行时，我们可以杀掉其子进程，父进程并不会因为子进程的终止而终止。
- 在进程管理中，当我们发现占用资源过多，或无法控制的进程时，应该杀死它，以保护系统的稳定安全运行；





对于Linux进程的管理，是通过进程管理工具实现的，比如ps、kill、pgrep等工具





ps 的参数说明;
获得线程信息:

ps -eLf

ps axms

获得进程树:

ps -ejH

ps axjf

l 长格式输出;

u 按用户名和启动时间的顺序来显示进程;

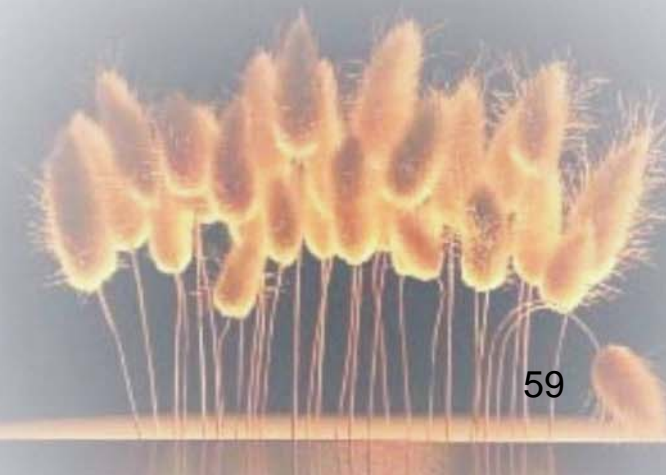
j 用任务格式来显示进程;

f 用树形格式来显示进程;

a 显示所有用户的所有进程 (包括其它用户);

x 显示无控制终端的进程;

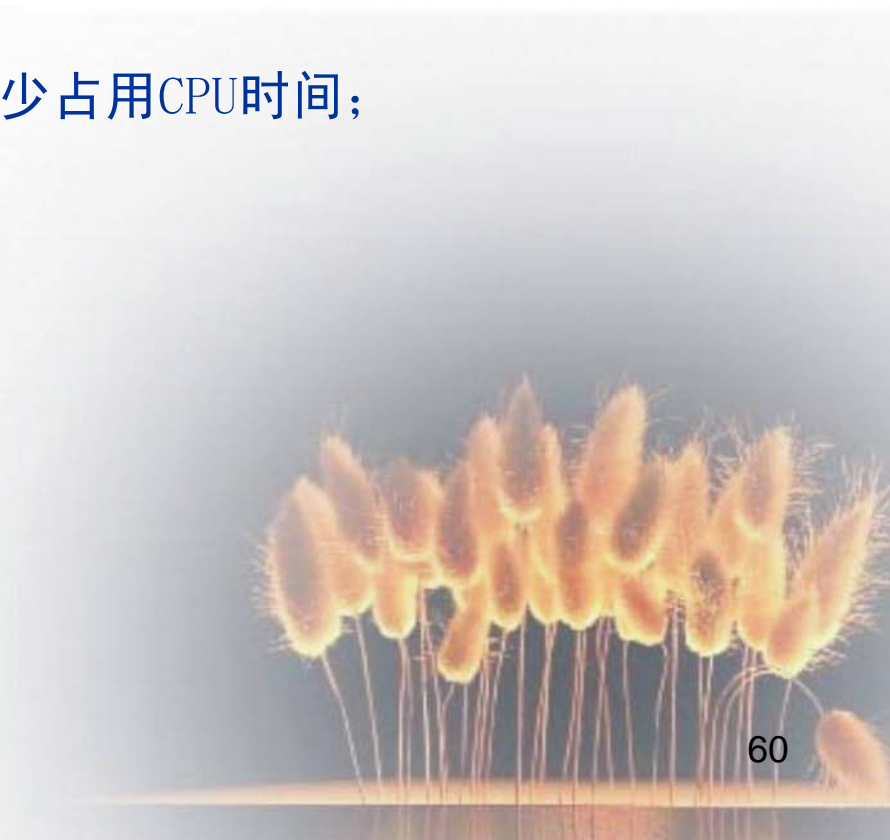
r 显示运行中的进程;



6.2.1 ps aux 或lax输出的解释



- USER 进程的属主;
- PID 进程的ID;
- PPID 父进程的ID;
- %CPU 进程占用的CPU百分比;
- %MEM 占用内存的百分比;
- NI 进程的NICE值, 数值大, 表示较少占用CPU时间;
- VSZ 进程虚拟大小;
- RSS 驻留中页的数量;
- TTY 终端ID
- WCHAN 正在等待的进程资源;
- START 启动进程的时间;
- TIME 进程消耗CPU的时间;
- COMMAND 命令的名称和参数;





STAT 进程状态

- D Uninterruptible sleep (usually IO)
- R 正在运行可中在队列中可过行的;
- S 处于休眠状态;
- T 停止或被追踪;
- W 进入内存交换 (从内核2.6开始无效);
- X 死掉的进程 (从来没见过);
- Z 僵尸进程;
- < 优先级高的进程
- N 优先级较低的进程
- L 有些页被锁进内存;
- s 进程的领导者 (在它之下有子进程);
- l is multi-threaded (using CLONE_THREAD, like NPTL pthreads do)
- + 位于后台的进程组;
- WCHAN 正在等待的进程资源; START 启动进程的时间; TIME 进程消耗CPU的时间; COMMAND 命令的名称和参数。



pgrep 是通过程序的名字来查询进程的工具，一般是用来判断程序是否正在运行。

用法：pgrep 参数选项 程序名

常用参数：

- l 列出程序名和进程ID;
- o 进程起始的ID;
- n 进程终止的ID;

举例：

```
[root@localhost ~]# pgrep -l httpd
4557 httpd
4560 httpd
4561 httpd
4562 httpd
```



- 终止一个进程，一般是通过 kill 、 killall、 pkill、 xkill 等进行。比如一个程序已经死掉，但又不能退出，这时就应该考虑应用这些工具。
- 另外应用的场合就是在服务器管理中，在不涉及数据库服务器程序的父进程的停止运行，也可以用这些工具来终止。为什么数据库服务器的父进程不能用这些工具杀死呢？原因很简单，这些工具在强行终止数据库服务器时，会让数据库产生更多的文件碎片，当碎片达到一定程度的时候，数据库就有崩溃的危险。比如mysql服务器最好是按其正常的程序关闭，而不是用pkill mysqld 或killall mysqld 这样危险的动作；当然对于占用资源过多的数据库子进程，我们应该用kill来杀掉。



kill一般是和ps 或pgrep 命令结合在一起使用的

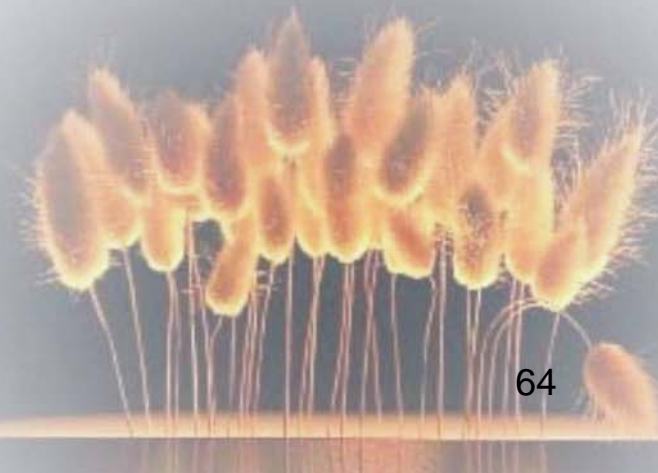
kill [信号代码] 进程ID

注：信号代码可以省略（信号代码是15，正常结束）；我们常用的信号代码还有 9，表示强制终止

例：

```
[root@localhost ~]# pgrep -l gaim  
5031 gaim
```

```
[root@localhost ~]# kill -9 5031
```





killall 通过程序的名字，直接杀死所有进程

用法：killall 正在运行的程序名

killall 也和ps或pgrep 结合使用，比较方便

通过ps或pgrep 来查看哪些程序在运行

举例：

```
[root@localhost test]# pgrep -l gaim
```

```
2979 gaim
```

```
[root@localhost test]# killall gaim
```





pkill 和killall 应用方法差不多，也是直接杀死运行中的程序；如果您想杀掉单个进程，请用kill来杀掉

应用方法：pkill 正在运行的程序名

举例：

```
[root@localhost test]# pgrep -l gaim
```

```
2979 gaim
```

```
[root@localhost test]# pkill gaim
```





和ps 相比，top是动态监视系统任务的工具，top 输出的结果是连续的数据：

- b 以批量模式运行，但不能接受命令行输入；
- c 显示命令行，而不仅仅是命令名；
- d N 显示两次刷新时间的间隔，比如 -d 5，表示两次刷新闻隔为5秒；
- i 禁止显示空闲进程或僵尸进程；
- n NUM 显示更新次数，然后退出。比如 -n 5，表示top更新5次数据就退出；
- p PID 仅监视指定进程的ID；PID是一个数值；
- q 不经任何延时就刷新；
- s 安全模式运行，禁用一些效互指令；
- S 累积模式，输出每个进程的总的CPU时间，包括已死的子进程；



命令格式: `mount [-t vfstype] [-o options] device dir`

1.-t vfstype 指定文档系统的类型，通常不必指定。mount 会自动选择正确的类型。常用类型有：

光盘或光盘映像: iso9660

DOS fat16文档系统: msdos

Windows 9x fat32文档系统: vfat

Windows NT ntfs文档系统: ntfs

Mount Windows文档网络共享: smbfs

UNIX(LINUX) 文档网络共享: nfs

2.-o options 主要用来描述设备或档案的挂接方式。常用的参数有：

loop: 用来把一个文档当成硬盘分区挂接上系统

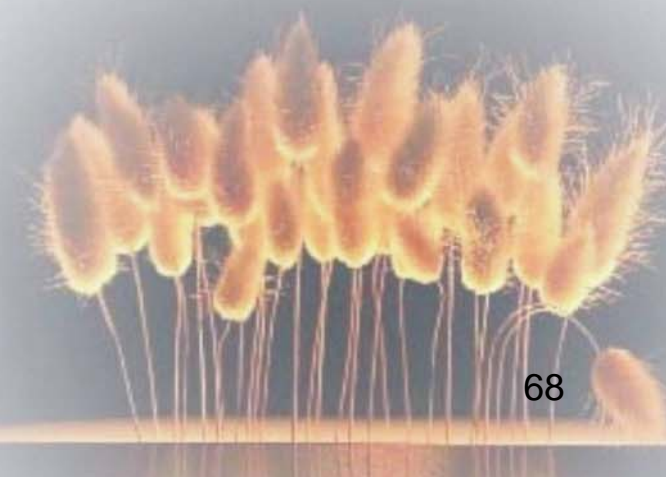
ro: 采用只读方式挂接设备

rw: 采用读写方式挂接设备

iocharset: 指定访问文档系统所用字符集

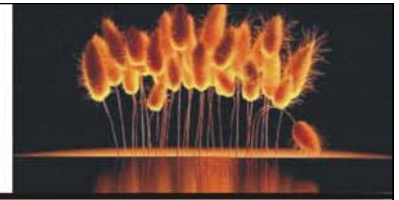
3.device 要挂接(mount)的设备。

4.dir设备在系统上的挂接点(mount point)。





- 一. 插入移动硬盘或U盘前后两次用fdisk -l查看系统的硬盘和硬盘分区情况
- 二. #mkdir -p /mnt/usb
- 三. #mount -t vfat /dev/sdd1 /mnt/usb



谢谢！！

