

# Linux基础知识简介

## Introduction of Linux

单细胞中心生物信息学平台  
逐码科技 SimpleHPC

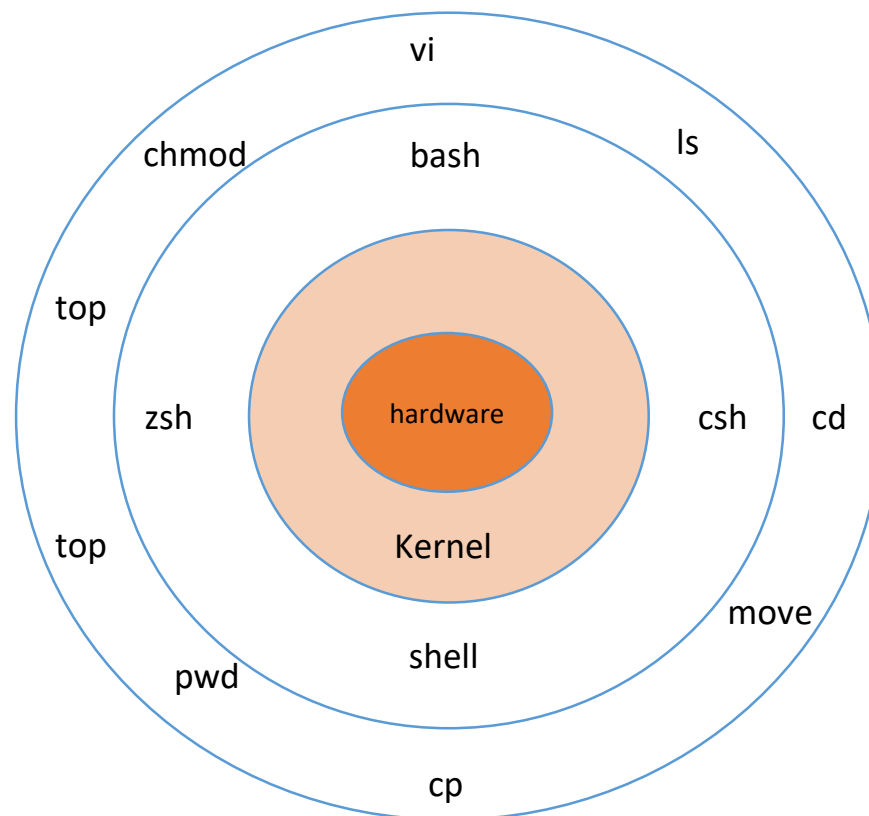
# Linux 起源

- 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

# Linux Shell

1. 特定的用户程序用来连接用户和系统内核, 调用硬件资源 CPU, 内存, 显卡等

2. 用来运行其他的应用程序



# 系统管理

- Linux系统管理的目标是确保Linux服务器和系统可靠地运行，满足用户需求和安全要求。Linux系统管理员需要掌握Linux系统的基础知识和相关技能，包括用户管理、文件系统管理、网络服务配置、安全性管理、日志管理等等。
- 本教程将从用户管理、进程管理、文件权限、软件包管理等几个方面进行简单的讲解。

# 用户管理

- Linux是一个多用户的操作系统。
- Linux的多用户概念是指多个用户同时可以使用这个系统。每个用户默认在/home下的都有自己的家目录，管理员root的家目录位于/root。用户执行任务的时候一般在自己的家目录中操作，用户之间互相不影响。

# 用户管理

- 查看所有用户账号信息

两种方法（推荐第2种）：

① `cat /etc/passwd`

② `getent passwd`

查看uid大于1000的用户账号信息：

```
getent passwd | awk -F ':' '$3>1000'
```

# 用户管理

- 增加用户

useradd USERNAME

常用参数:

-u uid 指定用户ID

-g gid 指定用户所属的组

-m 登入目录 当指定目录不存在时, 自动创建目录

-s 指定用户登录后使用的shell, 默认为 /bin/bash, 禁止用户登录可以使用-s /sbin/nologin

例:

```
useradd -g 1024 -m /cluster/home/pm pm
```

# 用户管理

- 修改密码

passwd USERNAME

常用参数:

-d 删除密码

-x 指定口令最长存活期, 天数

-i 口令过期多少天停用账户

例:

passwd pm



# 用户管理

- 删除用户

`userdel USERNAME`

常用参数:

`-r` 删除用户的同时将用户家目录一起删除

例:

`userdel -r pm`

# 用户管理

- 用户组相关

查看所有用户组:

```
getent group
```

添加用户组:

```
groupadd -g GID GROUPNAME
```

将用户添加到某个用户组:

```
usermod -aG GROUPNAME USERNAME
```

删除用户组:

```
groupdel GROUPNAME
```

# 进程管理

- Linux 进程管理指的是在 Linux 操作系统中对正在运行的进程进行管理和控制的技术。进程是 Linux 操作系统中最重要的概念之一，它是操作系统中一个正在执行的程序的实例。

# 进程管理

- ps 命令

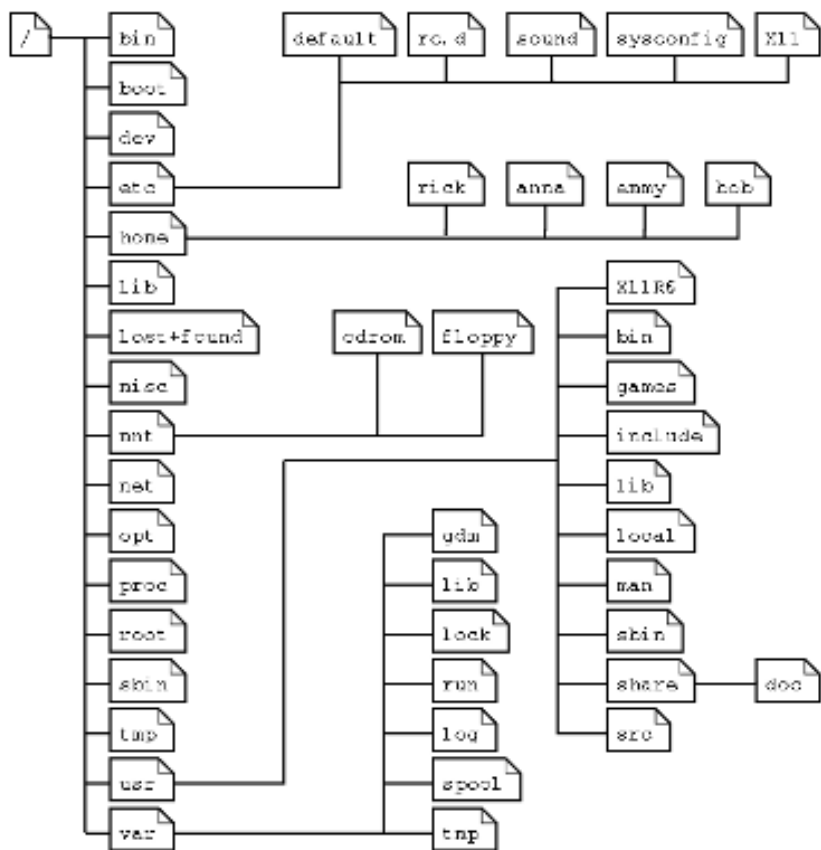
ps 命令用于列出当前系统中所有的进程，并显示每个进程的信息，如 PID（进程 ID）、PPID（父进程 ID）、CPU 占用率、内存使用情况、进程状态等。通常，只需要使用 ps 命令来查看当前系统中运行的进程即可。

- top

演示 top 使用

# Linux 文件目录

/bin/ls



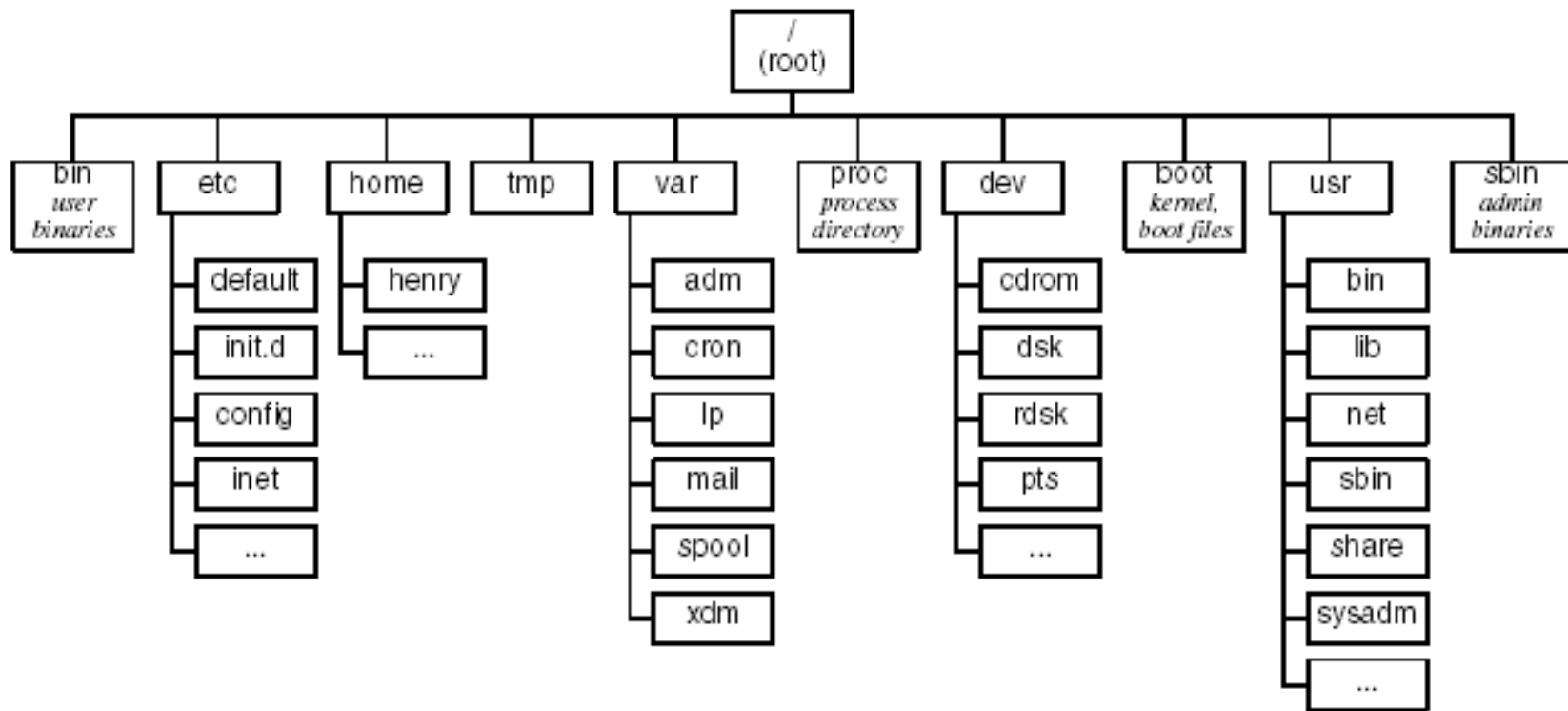
- owner , group, everyone
- Read-4, Write-2, Execute=1
- 755 == (RWX)-(RX)-(RX)

owner      group      everyone

```
-rw-r--r-- 1 root root 64 Jul 6 16:19 log1
-rw-r--r-- 1 root root 64 Jul 6 16:19 log2
-rw-r--r-- 1 root root 64 Jul 6 16:19 log3
-rw-r--r-- 1 root root 64 Jul 6 16:19 log4
```

The permissions `-rw-r--r--` are circled in blue, with arrows pointing to the labels 'owner', 'group', and 'everyone' above them.

# Linux 目录结构



# 文件目录说明

/	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

# 文件目录说明

/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



# 文件目录类命令

- 浏览目录命令:ls cd dir
- 浏览文件命令:cat more less
- 目录操作命令:mkdir rmdir
- 文件操作命令:touch vi rm cp mv ln tar gzip  
gunzip whereis

## 列出文件列表的ls命令

- 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。
- 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]# _
```

## 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
```

## mkdir、rmdir命令

- 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)

- 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
```

## 复制文件的cp命令

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

## 删除文件或目录rm命令

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：强制删除文件

```
[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

## rm命令参数-i使用

-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。

## cat命令

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



## more命令

more命令一般用于要显示的内容会超过一个画面长度的情况。为了避免画面显示时瞬间就闪过去，用户可以使用more命令，让画面在显示满一页时暂停，此时可按空格键继续显示下一个画面，或按Q键停止显示。

```
[test@linux test]$ more /etc/named.conf
```

显示 / etc/named.conf文本文件的内容

当用ls命令查看文件列表时，如果文件太多，则可配合more命令使用：

```
[test@linux etc]$ ls -al | more
```

以长格形式显示etc目录下的文件列表，显示满一个画面便暂停，可按空格键继续显示下一画面，或按Q键跳离

## less命令

less命令的用法与more命令类似，也可以用来浏览超过一页的文件。所不同的是less命令除了可以按空格键向下显示文件外，还可以利用上下键来卷动文件。当要结束浏览时，只要在less命令的提示符“:”下按Q键即可。

```
[test@linux etc]$less named.conf
```

显示/etc/named.conf的文本文件内容

```
[test@linux etc]$ls -al | less
```

以长格形式列出/etc目录中所有的内容。用户可按上下键浏览或按Q键跳离

## 显示命令head、tail

- head功能：用于显示文件前几行的内容

格式：head[参数]<文件名>

```
[root@linux root]# head -10 /etc/passwd
```

显示/etc/passwd/文件的前10行内容

- tail功能：用于显示文件后几行的内容

格式：tail[参数]<文件名>

```
[root@linux root]# tail -10 /etc/passwd
```

显示/etc/passwd/文件的倒数10行内容

## 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

## 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命令

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
```

## grep命令

功能：在文件中搜索匹配的字符并进行输出

格式：grep[参数] <要找的字符串> <要寻找字符串的源文件>

```
[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 tty1
```

## touch命令

功能：生成一个空文件或修改文件的存取/修改的时间记录值。

格式：touch [参数] <文件名>

```
[test@linux test]$ touch *
```

将当前下的文件时间修改为系统的当前时间

```
[test@linux test]$ touch -d 20070806 test
```

将test文件的日期改为20070806

```
[test@linux test]$ touch abc
```

若abc文件存在，则修改为系统的当前时间；若不存在，则生成一个为当前时间的空文件



## who或w命令

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]# _
```

## 打包命令tar

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

## tar命令范例

```
[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

## date、cal、clock

date命令可以显示/修改当前的日期时间

```
[root@linux root]# date 121010232004
```

将时间更改为12月10日10点23分2004年 [MMDDhhmmYY]

```
[root@linux root]# cal          显示日历
```

```
[root@linux root]# clock       显示日期时间
```

```
[root@linux root]# date
Wed Mar  9 17:35:55 CST 2005
[root@linux root]# cal
      March 2005
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[root@linux root]# clock
Thu 10 Mar 2005 04:38:59 PM CST 0.832619 seconds
[root@linux root]# _
```

## 显示日历或年历的cal命令

cal(calendar)

功能：显示一个日历

格式：cal [参数] 月 年

[root@linux root]# cal 显示当月的日历

[root@linux root]# cal 4 2004

显示2004年4月的日历

[root@linux root]# cal -y 2003

显示2003年的日历

## vi命令详解

- 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

- 移动光标类命令

上:k nk:向上移动n行

下:j nj:向下移动n行

左:h nh:向左移动n列

右:l nl:向右移动n列

gg可以移到第一行 G移到最后一行

## vi命令详解

### 三. 屏幕翻滚类命令

Ctrl+u: 向文件首翻半屏

Ctrl+d: 向文件尾翻半屏

Ctrl+f: 向文件尾翻一屏

Ctrl+b: 向文件首翻一屏

z然后按“-”——将光标所在的那一行移至屏幕底部

z然后按"."——将光标所在的那一行移至屏幕中部

nz后按回车: 将第n行滚至屏幕顶部, 不指定n时将当前行滚至屏幕顶部

### 四. 插入文本类命令

i: 在光标前

I: 在当前行首

a: 光标后

A: 在当前行尾

o: 在当前行之下新开一行

O: 在当前行之上新开一行

r: 替换当前字符

R: 替换当前字符及其后的字符, 直至按ESC键

s: 从当前光标位置处开始, 以输入的文本替代指定数目的字符

S: 删除指定数目的行, 并以所输入文本代替之

ncw或nCW: 修改指定数目的字

nCC: 修改指定数目的行

## vi命令详解

### 五. 输入控制命令

- : n1,n2 co n3: 将n1行到n2行之间的内容拷贝到第n3行下
- : n1,n2 m n3: 将n1行到n2行之间的内容移至到第n3行下
- : n1,n2 d : 将n1行到n2行之间的内容删除
- : w : 保存当前文件
- : e filename: 打开文件filename进行编辑
- : x: 保存当前文件并退出
- : q: 退出vi
- : q!: 不保存文件并退出vi
- : !command: 执行shell命令command
- : n1,n2 w!command: 将文件中n1行至n2行的内容作为command的输入并执行之, 若不指定n1, n2, 则表示将整个文件内容作为command的输入
- : r!command: 将命令command的输出结果放到当前行



## vi命令详解

### 六. 删除命令

ndw或ndW: 删除光标处开始及其后的n-1个字

d0: 删至行首

d\$: 删至行尾

ndd: 删除当前行及其后n-1行

x或X: 删除一个字符, x删除光标后的, 而X删除光标前的

Ctrl+u: 删除输入方式下所输入的文本

### 七. 搜索及替换命令

/pattern: 从光标开始处向文件尾搜索pattern

?pattern: 从光标开始处向文件首搜索pattern

n: 在同一方向重复上一次搜索命令

N: 在反方向上重复上一次搜索命令

: s/p1/p2/g: 将当前行中所有p1均用p2替代

: n1, n2s/p1/p2/g: 将第n1至n2行中所有p1均用p2替代

: g/p1/s//p2/g: 将文件中所有p1均用p2替换

# 系统信息类命令

- df命令
- du命令
- free命令

## df命令

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]# _
```

## du命令

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]# _
```

## free命令

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]# _
```

# 网络通讯类命令

- `ssh` 登录到远程计算机上
- `netstat` 查看网络的状况
- `ping` 查询某个机器是否在工作
- `route` 设置系统网络路由

### 5.3.2 ssh登录到远程计算机上

ssh命令的一般形式

ssh 主机名/IP

其中“主机名/IP”是要连接的远程机的主机名或IP地址

ssh不但为普通终端提供终端仿真，而且支持 X Window等图形环境

scp 拷贝命令

scp 文件名 用户名@ip地址:/path/

## netstat查看网络的状况

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查询某个机器是否在工作

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

-l 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

-I 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

## route 设置系统网络路由

在本地 IP 路由表中显示和修改条目

```
route [-f] [-p] [Command] [Destination] [mask Netmask] [metric Metric] [Gateway] dev [Interface]
```

-f 清除所有网关入口的路由表;

-p 与 add 命令一起使用时使路由具有永久性;

Command 指定您想运行的命令 (Add/Change/Delete/Print);

Destination 指定该路由的网络目标;

mask Netmask 指定与网络目标相关的网络掩码 (也被称作子网掩码) ;

metric Metric 为路由指定一个整数成本值标 (从 1 至 9999) , 当在路由表(与转发的数据包目标地址最匹配)的多个路由中进行选择时可以使用;

Gateway 指定网络目标定义的地址集和子网掩码可以到达的前进或下一跃点 IP 地址。

# 问答