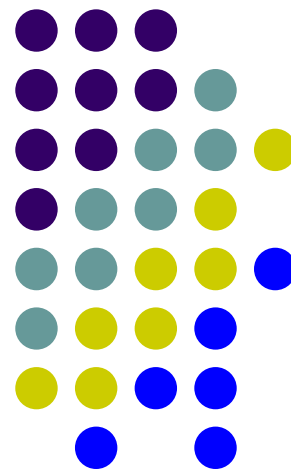
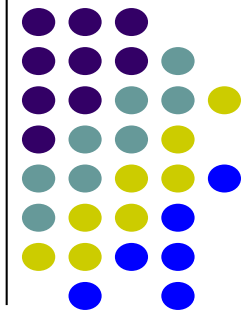


第5章

Linux常用命令



本次章内容

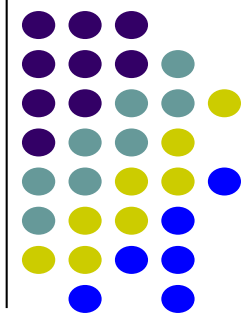


Shell命令简介

常用的Shell命令

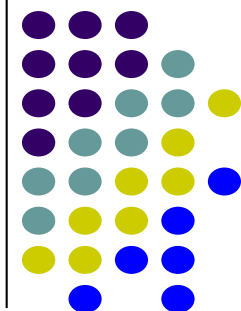
深入Shell

5.1 Shell命令简介



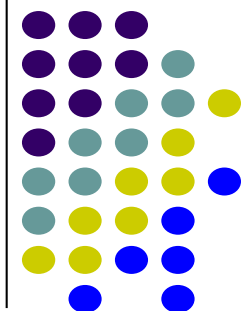
- 字符界面下，用户对Linux的操作通过Shell命令来实现。Shell是Linux内核与用户之间的接口，负责解释执行用户从终端输入的命令行。
- Shell可以执行的命令分为两大类：**内置命令**和**实用程序**。
- **内置命令**：为提高执行效率，将一部分最常用命令的解释器构置于shell内部。

Shell命令简介



- **实用程序**又分为：
 - **Linux程序**：Linux自带的命令，位于/bin和/sbin
 - **应用程序**：位于/usr/bin 和/usr/sbin等目录中的应用程序
 - **Shell脚本**：shell语言编写的脚本程序
 - **用户程序**：用户编写的其他可执行程序

Shell命令简介



- Shell对于用户输入的命令，有以下**3种处理方式**：
 - 如果用户输入的是**内置命令**，那么由Shell的内部解释器进行解释，并交由内核执行。
 - 如果用户输入的是**实用程序命令**，而且给出了命令的路径，那么Shell会按照用户提供的路径在硬盘中查找，并交由内核执行；否则输出提示信息。
 - 如果用户输入的是**实用程序命令**，但没给出路径，那么Shell会根据PATH环境变量所指定的路径依次进行查找，找到就调入内存，交由内核执行；否则输出提示信息。

Shell命令格式



- Linux中的命令格式一般为：

命令名 [选项] [参数]

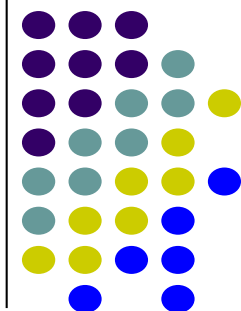
- 命令名是描述该命令功能的英文单词或缩写。命令名必不可少；
- 选项是执行该命令的限定参数或功能参数。选项可以有多个，通常以“-”开头；

例如： `ls -l /home`。

如果有多个选项，既可以连在一起写，如`ls -al`，也可以分开写，如`ls -a -l`。

- 参数是执行该命令所必需的对象，如文件，目录等。根据命令的不同，参数可以有一个或多个，也可以没有。

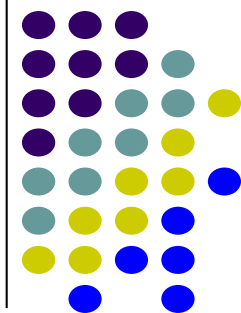
Shell命令格式



注意：

- Linux系统**严格区分**英文字母的大小写
- 命令名、选项、**参数之间必须有空格**。Shell自动过滤多余的空格
- 一个命令行可以输入**多个命令**，之间用“**；**”分隔
- 输入命令或文件名时可以通过按**Tab键来补齐**，可以用上下箭头来上下翻动显示历史命令。

注销、重启与关机

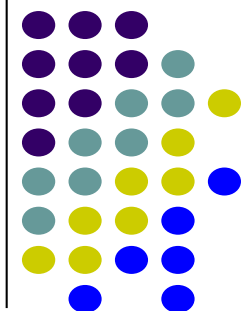


1. 注销

- 注销即退出登录状态：“**exit**”命令或**Ctrl+d**。
- 当用户执行完各种操作后，就要及时退出系统，这是一个良好的习惯，即使是暂时离开机器也同样如此。
- 例如：

```
[root@ Linux root] # exit
```
- 在提示符# 后键入命令**exit**后回车即可退出系统，重新出现**login**提示符。一个用户注销不影响其他登录用户的操作。

注销，重启与关机



2. 重启

- 有时在对系统的某一配置做了修改，或者安装了新的软件，需要重新启动才能使所做的修改生效，这时就要用到**重新启动命令reboot**，

例如：

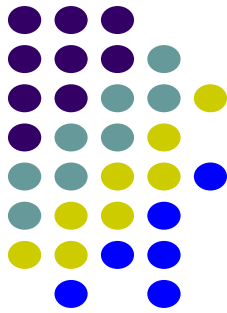
```
[root@ Linux root] # reboot
```

或 “shutdown -r now”（需要超级用户的权限）

参数说明：

-r 关闭并重新启动此计算机

注销，重启与关机



3、关机

- 当做完所有的工作要关闭系统时，切不可直接关掉电源，一定要首先执行关闭系统命令。命令如下：

```
[root@ xLinux root] #halt
```

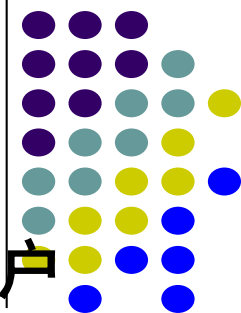
- 或者 “**shutdown -h now**”

参数说明：

-h：关机后关闭电源；

-time：设定关机前的时间

注销，重启与关机



- 由于Linux是多用户操作系统，同一时间可能有多个用户正在使用，立即关机可能导致其他用户的工作被突然打断，通常，系统管理员在关机或重新启动之前都会提前发出提示信息，提醒所有的用户系统即将关闭或重启，并预留一段时间让用户结束各自的工作，并退出登录。
- 常用的关机或重启命令如下：

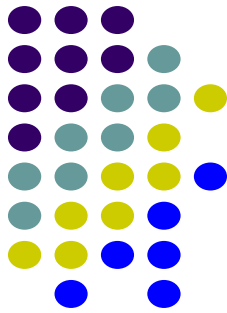
```
[root@ xLinux root] # shutdown -h 10      #10分钟后关机
```

```
[root@ xLinux root] # shutdown -r 10      #10分钟后重启
```

练习1：

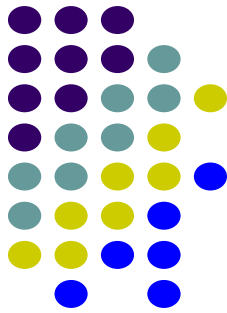
设置3：40开始关机，并关闭电源。

5.2 常用Shell命令



1. 与时间相关的命令
2. 管理文件和目录的命令
3. 其他Shell命令

与时间相关的命令



date命令

格式：**date [MMDDhhmm[YYYY]]**

功能：查看或修改系统时间。

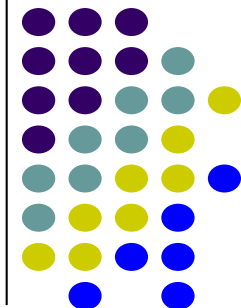
- data命令显示的内容依次为星期、月份、日期、小时、分钟，秒和年。

例如：将当前时间修改为8月15日20时04分

[root@ Linux root] # date 08152004

用户必须拥有超级用户权限才能修改系统时间。必须按照月份、日期、小时、分钟，秒和年的顺序修改，年份可以占4位，也可以占2位，其他部分占2位，不足2位添0补足。年份可省略，其他部分不可省。

与时间相关的命令



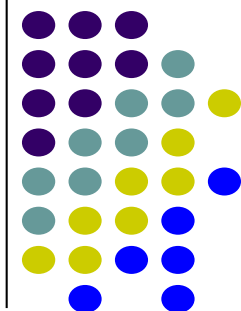
cal命令

格式：**cal** [YYYY]

功能：显示日历。

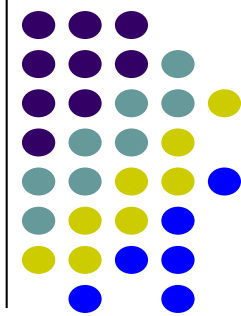
```
[root@Linux2018 root]# cal
      三月 2018
日 一 二 三 四 五 六
    1  2  3
 4  5  6  7  8  9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31
```

管理文件和目录的命令



命令	功能
pwd	判定当前目录
cd	改变所在目录
ls	查看目录下的内容
cat	显示文件的内容
grep	在文件中查找某字符
touch	创建文件
cp	复制文件
mv	移动文件
rm	删除文件
more	分屏显示文件内容
tail	显示文本结尾部分
mkdir	创建目录
rmdir	删除空目录
find	查看满足表达的文件和目录

管理文件和目录的Shell命令



pwd命令:

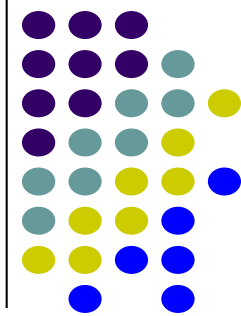
格式: pwd

功能: 显示当前目录的绝对路径。

注:

- 绝对路径是指从根目录 (/) 开始到当前目录 (文件) 的路径;
- 而相对路径是指从当前目录到其下子目录 (文件) 的路径。
- 目录之间的层次关系总是用 “/” 来表示。

管理文件和目录的Shell命令



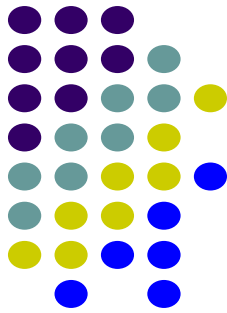
cd命令:

格式: `cd [目录]`

功能: 切换到指定目录。

注:

- 超级用户的主目录是/root;
- 而普通用户的主目录是/home下与该用户同名的子目录。
- 例如, 普通用户hellen主目录默认就是/home/hellen。



例如

- 从**相对路径**切换到/home/tux/doc目录

```
$ pwd
```

```
/home/tux
```

```
$ cd doc
```

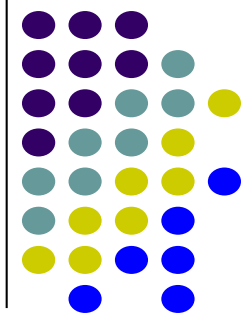
- 从**绝对路径**直接切换到/home/tux/doc目录

```
$ cd /home/tux/doc
```

练习2:

进入user用户主目录，显示当前的路径。

管理文件和目录的Shell命令



ls 命令

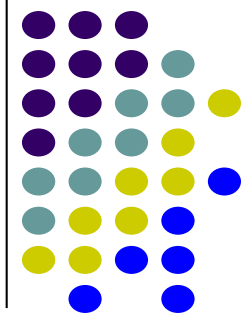
格式：ls [选项] [文件|目录]

功能：显示指定目录中的文件和子目录信息。当不指定目录时，显示当前目录中的文件和子目录信息。

主要选项说明：

- -a ， 显示所有文件和子目录，包括以 “.”开头的隐藏文件。
- -l ， 显示文件和子目录的详细信息，包括文件类型、权限、所有者和所属组群、文件大小，修改时间等。

管理文件和目录的Shell命令



- -R ， 列出目录和其子目录下的文件。
- -d ， 参数是目录， 只显示目录的信息， 不显示所包含文件的信息。
- -t ， 按时间顺序显示文件， 新文件排在前面。ls命令默认按字母顺序排列。

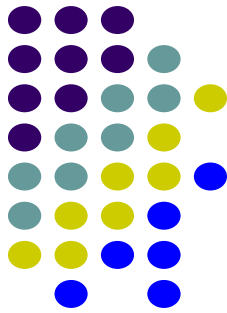
练习3:

查看当前目录中**所有文件和子目录的详细信息**。

练习4:

按时间顺序查看当前目录下的后缀为 “.c” 的文件， 列出文件的**详细信息**。

管理文件和目录的Shell命令



cat命令

格式：cat [选项] 文件列表

功能：显示文本文件的内容。

主要选项说明：

-n (number) 在每一行前显示行号

-u 输出不经过缓冲区

注：使用cat命令可以查看文本文件的内容。如果查看其他类型的文件只能显示乱码。

练习5：

显示Readme.txt 文件的内容，并在每一行前加行号。

管理文件和目录的命令



more命令

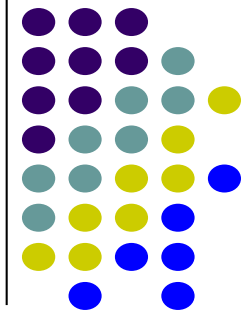
格式：more 文件

功能：分屏显示文本文件的内容。

例如：分屏显示

注：按Enter键显示下一行内容；按空格键显示下一屏的内容，按q键，则退出more命令。

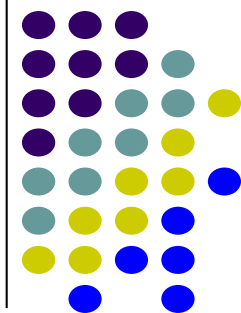
管理文件和目录的命令



less 命令

Less命令与more命令非常相似，和more命令唯一的不同是more命令只能向后翻页，而less命令可以使用上下方向键、PaUp或PaDn向前向后翻页。

管理文件和目录的Shell命令



tail 命令

格式: **tail** [选项] 文件

功能: 显示文本文件的结尾部分, 默认显示文件的最后10行

head命令与tail命令很相似, head命令显示文本文件的开头部分, 默认显示文件的开头10行。

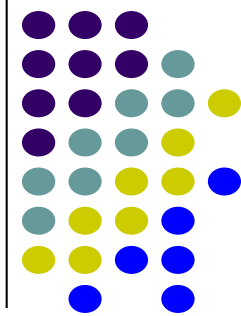
主要选项说明:

-n: 数字, 指定显示的行数。

练习6:

查看/etc/passwd文件的后5行。

管理文件和目录的Shell命令



cp命令

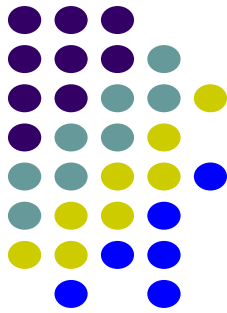
格式：**cp** [选项] 源文件或目录 目标文件或目录

功能：**复制文件或目录**

主要选项说明：

- r 或-R: 若给出的源文件是一目录文件，此时cp将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名。
- b: 若存在同名文件，则在覆盖之前备份原来的文件。
- f: 强制覆盖同名文件。
- p: 保持文件原来的属性

管理文件和目录的Shell命令



- 例如，复制文件

将/dir1/a复制到/dir2目录下，命名为b：

```
$ cp /dir1/a /dir2/b
```

- **cp 命令复制多个文件到目录**，命令语法：

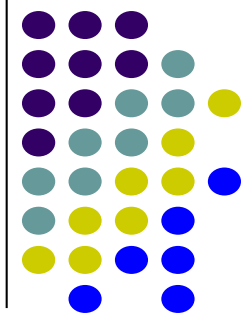
cp 文件名1 文件名2 ... 目标目录名(最后的项目必须是目录)

例如：将a b 二个文件复制到newdir目录下

```
$ cp a b newdir/
```

练习7：将文件exam.c 复制到/user/wang目录下，并改名为shiyang.c

管理文件和目录的Shell命令



mkdir 命令

格式：mkdir [选项] 目录

功能：创建目录

主要选项说明：

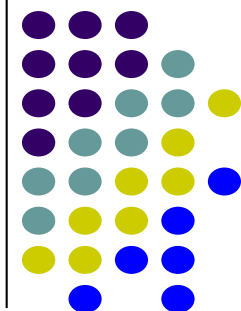
-m: 对新创建目录的设置访问权限，也可以用chmod命令。

-p: 一次性创建多级目录。

例如：创建名为test的目录，并在其下创建 linux目录

```
$ mkdir -p test/linux
```

管理文件和目录的Shell命令



rmmdir 命令

格式：rmmdir [选项] 目录名

功能：删除空目录。

例如，删除/home/limhai/newdir 目录的方法：

```
$ rmmdir /home/limhai/newdir
```

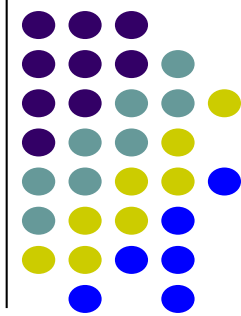
或

```
$ cd /home/limhai
```

```
$ rmmdir newdir
```

rmmdir命令删除的目录**必须是空目录**，若不是空目录，必须先
用rm 命令删除目录下所有文件。

管理文件和目录的Shell命令



rm 命令

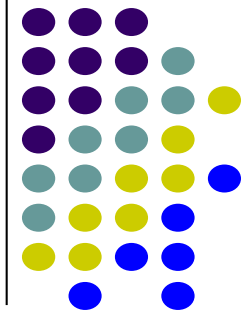
格式：rm [选项] 文件或目录

功能：删除文件或目录

主要选项说明：

- f: 强制删除，不需要确认，包括删除有写保护文件。
- r或-R: 按递归方式删除目录，即删除目录和目录里的所有内容。
- i: 交互方式操作，删除之前提示，要求用户回答“y”或“n”，避免误删除文件。

管理文件和目录的Shell命令



例如：

删除当前目录下所有以file打头的文件，删除前先询问。

```
rm -i file*
```

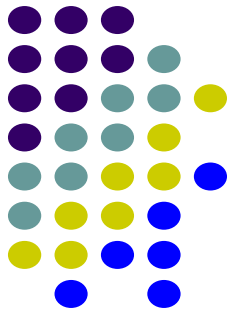
删除目录doc，无论是否是空目录。

```
rm -rf doc/
```

练习8：

将/home目录拷贝到/root目录下，然后再将root下的home及子目录强制删除。

管理文件和目录的Shell命令



mv命令

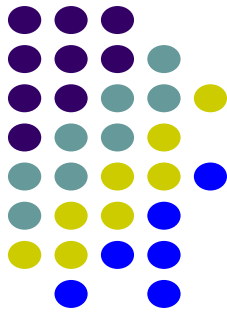
格式：mv [选项] 源文件或目录 目标文件或目标目录

功能：移动或重命名文件或目录

主要选项说明：

- b：若存在同名文件，则在覆盖之前备份原来的文件。
- f：强制覆盖同名文件。
- i：交互方式操作。如果mv操作将导致对已存在的目标文件的覆盖，此时系统询问是否重写，要求用户回答“y”或“n”，这样可以避免误覆盖文件。

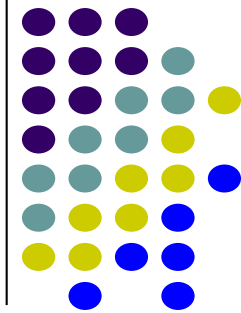
管理文件和目录的Shell命令



例如

- 用mv 命令修改文件名，命令语法为
`$ mv 原文件名 目标文件名`
比如，`$ mv myfile1 myfile2`
- 用mv 命令移动文件，命令语法：
`$ mv 文件名1 文件名2 目标目录名`
 - (mv命令可以移动多个文件，但最后的项目必须是目录)
比如，`mv myfile1 myfile2 newdir/`

管理文件和目录的Shell命令



find 命令

格式：**find** [路径] 表达式

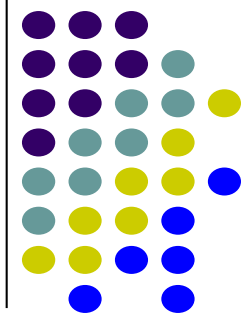
功能：从指定路径开始向下搜索满足表达式的文件和目录。

不指定路径，则查找当前目录。当查找到用户不具有执行权限的目录时，屏幕将显示“权限不够”等提示信息。

主要表达式：

- name 文件 按文件名查找，可使用通配符
- group 组群名 查找文件所属组群为指定组群的文件
- user 用户名 查找文件所有者为指定用户的文件
- type 文件类型 按照文件类型查找，其中d为目录文件，l为符号链接文件。
- size n [+|-] 查找指定大小的文件，“+”表示超过，“-”表示不足。

管理文件和目录的Shell命令



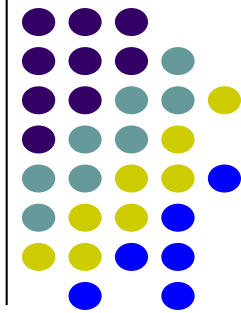
例：查找/ect目录中以fs开头的文件

```
$ find /ect -name fs*
```

例：查找当面目录找中所有大于100KB的文件和目录

```
$ find -size +100k
```

管理文件和目录的Shell命令



grep命令

格式：**grep** [选项] 字符串 文件列表

功能：从指定文件中查找符合条件的字符串，默认显示指定字符串所在行的内容。

主要选项说明：

-n: 显示行号

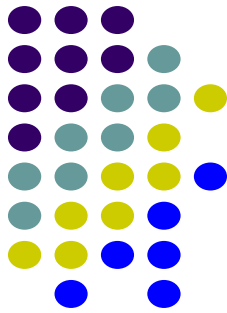
-v: 显示不包括指定字符串的行

-i: 查找时不区分大小写

例如：显示文件file中包含字符串abc的行

```
$ grep "abc" file
```

管理文件和目录的Shell命令



touch 命令

格式: **touch** [选项] 文件或目录

功能: 若指定文件不存在, 则创建一个空文件; 若文件存在则将更新文件时间

例如

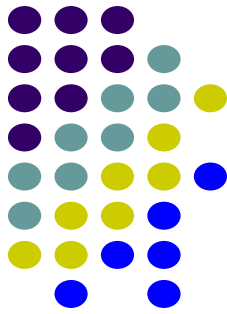
创建一个新文件:

```
$ touch newfile
```

将文件file的最后修改时间设置为当前时间:

```
$ touch file
```

修改文件权限的命令



chmod命令

格式1: chmod 数字模式 文件

格式2: chmod 功能模式 文件

功能: 修改文件的访问权限。

功能模式由以下三部分组成:

对象: u 文件所有者

g 同组用户

o 其他用户

操作符: + 增加权限

- 删除权限

= 赋予给定权限

权限: r 读取权限

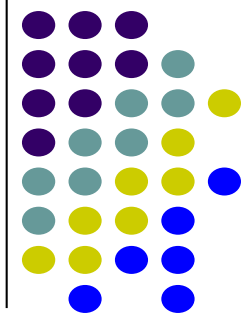
w 写入权限

x 执行权限

练习9: 给文件file1加上其他用户可执行属性的命令

练习10: 修改/home/abc的属性为: 属主具有一切权限, 同组用户和其他用户具有可读可写的权限。

修改文件权限的命令



数字模式：

- 一般是三个数字：

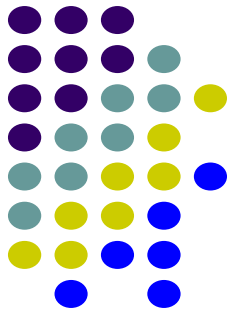
第一个数字表示文件所有者的权限

第二个数字表示与文件所有者同一个用户组的其他用户的权限

第三个数字表示其它用户组的权限。

- 权限分为三种：读（ $r=4$ ），写（ $w=2$ ），执行（ $x=1$ ）。综合起来还有可读可执行（ $rx=5=4+1$ ）、可读可写（ $rw=6=4+2$ ）、可读可写可执行（ $rwX=7=4+2+1$ ）。
- 例如，`chmod 755` 设置用户的权限为：
 - 1.文件所有者可读可写可执行
 - 2.与文件所有者同属一个用户组的其他用户可读可执行
 - 3.其它用户组可读可执行

其他Shell命令



wc 命令（文件内容统计命令）

格式： **wc [选项] 文件**

功能： 统计文件的行数、字数和字节数。

主要选项说明：

-c 仅显示文件的字节数

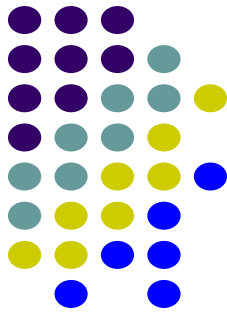
-l 仅显示文件的行数

-w 仅显示文件的字数

练习11：

统计文件/etc/fstab的行数、单词数、字符数。

其他Shell命令

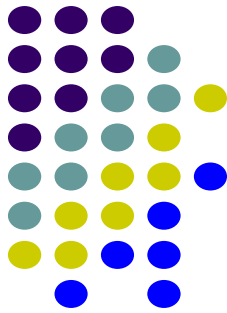


clear 命令

格式： **clear**

功能： 清楚当前终端的屏幕内容。

其他Shell命令



man命令

格式：man 命令名

功能：显示指定命令的手册页帮助信息。

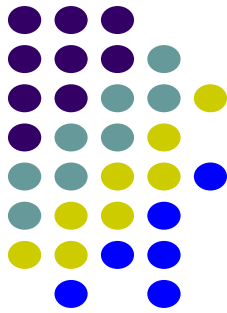
例如：

```
[root@ Linux root] # man ls
```

该命令输入命令后屏幕显示该命令在Shell手册页的第一屏帮助信息，可以使用上下方向键翻页查看。

按“q”键则退出man命令。

其他Shell命令



help命令

格式： **命令名--help**

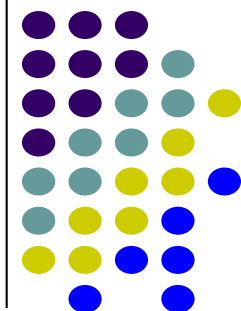
功能：显示指定命令的帮助信息。

例如：查看ls命令的帮助信息，并分屏显示。

```
[root@ Linux root] # ls --help|more
```

- 帮助信息较长时，可使用
[Shift+PgDn][Shift+PgUp]组合键向前向后翻页。
- 使用--help选项可以获取命令的帮助信息，但**不是所有的命令都有此选项**。

5.3 深入Shell



通配符

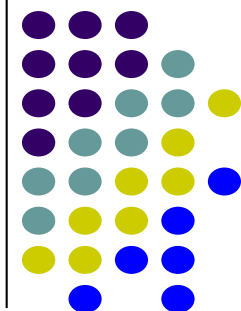
- Shell命令中可以使用通配符来同时匹配多个文件以方便操作。

1. 通配符 “ * ”

代表任意长度的任何字符，如“a*”可以表示“abc” “about” 等以“a”开头的字符串。

注意：通配符“*”不能与“.”开头的文件名匹配。例如，“*”不能匹配到名为“.file”的文件，而必须使用“.*”才能匹配到类似“.file”的文件。

通配符



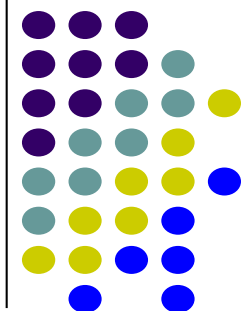
2. 通配符 “ ? ”

代表任意一个字符，如 “a?” 可以表示 “ab” “at” 等以 “a” 开头，并且仅有两个字符的字符串。

3. 字符组通配符 “ [] ” “ - ” “ ! ”

“[]” 表示指定的字符范围，“[]” 内的任意一个字符都用于匹配。“[]” 内的字符范围可以由直接给出的字符组成，也可以由起始字符、“-” 和终止字符组成。例如 “[abc]*” 或 “[a-c]*” 都表示所有以 “a” “b” 或者 “c” 开头的字符串。而如果使用 “!”，则表示不在此范围之内的其他字符。

通配符



例如：

ls *.png 列出所有PNG图片文件

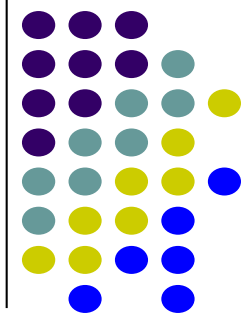
ls a? 列出所有字母a开头,文件名只有两个字符的所有文件

ls [abc]* 列出首字母是a、b或者c的所有文件

ls [!abc]* 列出首字母不是a、b或者c的所有文件

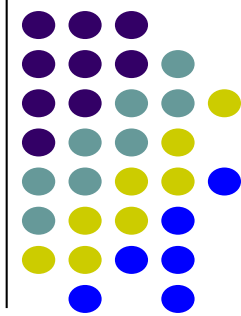
ls [a-z]* 列出首字母是小写字母的所有文件

重定向



- Shell中不使用系统的标准输入、标准输出或标准错误输出端口，重新指定至文件的情况称为**重定向**。
- 根据输出效果的不同，与输出相关的重定向可分为**输出重定向**、**附加输出重定向**和**错误输出重定向**3种。
- 与输入相关的重定向只有一种，称为**输入重定向**。

输出重定向



1. 输出重定向

输出重定向就是命令执行的结果不在标准输出（屏幕）上显示，而是保存到某一文件的操作，利用符号“>”来实现。

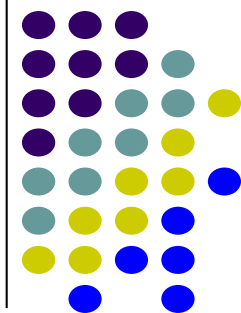
例如，将当前目录中所有文件和子目录的详细信息保存到list文件。

```
[root@ Linux root] # ls -al >list
```

上述命令执行后屏幕上不会出现任何信息，而是将本应该出现在屏幕上的内容全部保存到指定的文件中。指定的文件不需要预先创建。

cat命令可用于查看文本文件中的内容，而如果与输出重定向相配合，则有更加强大的功能。

输出重定向



(1) 创建文本文件

格式：`cat > 文件`

说明：输入此类命令后，屏幕光标闪烁，用户输入文件内容。所有的内容输入完成后，按“Enter”键将光标移动到下一行，然后按“Ctrl +D”组合键结束输入，再次出现Shell命令提示符。

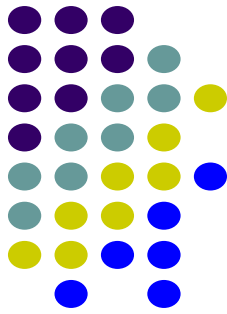
例如：

```
[root@ Linux root] # cat > f1
```

```
This is a file named f1
```

```
[root@ xLinux root] #
```


输出重定向



(2) 合并文本文件

格式：**cat > 文件列表 > 文件**

说明：将文件列表中所有文件的内容合并到指定文件。

例如：**将f1和f2文件合并生成f文件**

```
[root@ Linux root] # cat f1
```

```
This is a file named f1
```

```
[root@ Linux root] # cat f2
```

```
This is a file named f2
```

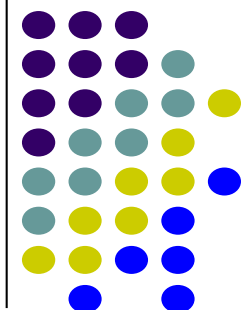
```
[root@ Linux root] # cat f1 f2> f
```

```
[root@ Linux root] # cat f
```

```
This is a file named f1
```

```
This is a file named f2
```

输出重定向



2. 附加输出重定向

附加输出重定向的功能与输出重定向基本相同。两者的不同之处在于附加输出重定向**将输出内容追加到原有的内容的后面**，而不会覆盖其内容，利用符号“**>>**”来实现附加输出重定向功能。

例如：向f1文件添加内容。

```
[root@Linux root] # cat >> f1
```

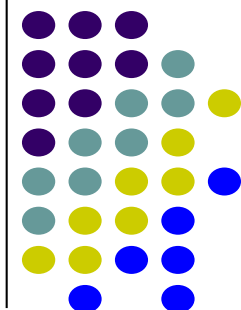
Append to f1

```
[root@Linux root] # cat f1
```

This is a file named f1

Append to f1

输出重定向



3. 错误输出重定向

Shell中标准输出与错误输出是两个独立的输出操作。前者是输出命令执行的结果，后者是输出命令执行中的错误信息。错误输出也可以进行重定向，并可分为两种情况：

- 程序的执行结果显示在屏幕上，而错误信息重定向到指定文件，使用 “**2>**” 符号。
- 程序的执行结果和错误信息都重定向到同一文件，使用 “**&>**” 符号。

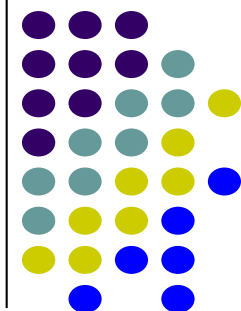
例如，查看/temp目录的文件和子目录信息，如有错误信息，则保存到err文件。

```
[root@ Linux root] # ls /temp 2>err
```

```
[root@ Linux root] # cat err
```

```
Ls: cannot access /temp: No such file or directory
```

输出重定向



4. 输入重定向

输入重定向跟输出重定向相反，是指**不从标准输入（键盘）读入数据，而是从文件读入数据**，用“<”符号来实现。由于大多数命令都以参数的形式在命令行上指定输入文件，所以输入重定向**不常使用**。但是，少数命令（如patch命令）不接受文件名作为参数，必须使用输入重定向。

例如：用输入重定向的方式查看f1文件的内容。

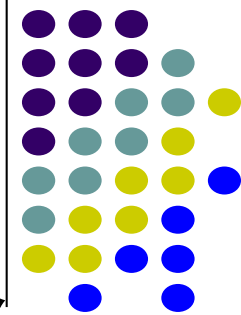
```
[root@ Linux root] # cat <f1
```

```
This is a file named f1
```

```
Append to f1
```

此时cat<f1命令的输入结果与cat f1命令完全相同。

管道



- **管道**是Shell的另一大特征，其将多个命令前后连接起来形成一个管道流。管道流中的每一个命令都作为一个单独的进程运行，**前一命令的输出结果传送到后一个命令作为输入，从左到右依次执行每个命令。**
- 利用 “|” 符号实现管道功能。

例如：利用管道统计当前目录中的文件和子目录的数目。

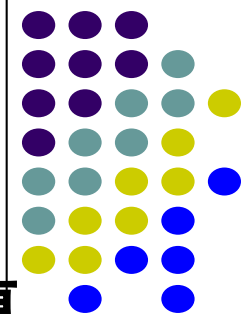
```
[root@ Linux root] # ls |wc -l
```

10

此时，屏幕上并不会显示ls命令执行的结果，因为ls命令执行的结果通过管道交给wc -l来执行，屏幕最后显示wc -l执行后的结果。

练习12：将当前目录下ls命令输出结果用grep命令把包含字符串“abc”的行过滤出来，作为wc命令的输入：

历史记录

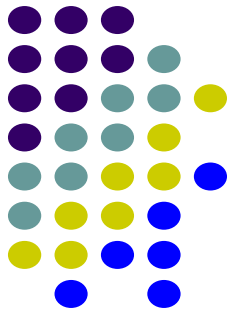


- 利用Shell命令进行操作时，用户需要对次反复输入相关命令，Shell提供历史记录、别名和自动补齐等功能，可以简化Shell命令输入工作。

1. 历史记录

Shell记录一定数量的已执行命令，当再次执行时，不用再次输入，直接调用即可。这些记录保存在用户目录名为**.bash_history**的隐藏文件。当用户关机或退出登录，本次操作中使用过的所有Shell命令就会追加保存到该文件中。Bash默认保存1000条。

历史记录

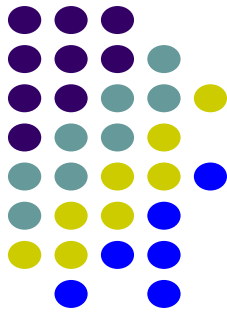


2. 历史记录使用方法

使用上下方向键、[PgUp]或[PgDn]，在Shell命令提示符后出现已执行过的命令。直接按[Enter]键就可以在执行这一命令。

先利用history命令查看Shell命令的历史记录，然后调用已执行过的Shell命令。

历史记录



3. history命令

格式：history [数字]

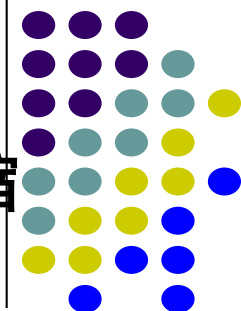
功能：查看Shell命令的历史记录。数字参数指定最近执行过的Shell命令的个数。如果不使用数字参数，则查看Shell命令的所有历史记录。

4. 再次执行已经执行过的Shell命令

格式：! 序号

功能：执行指定序号的Shell命令，而“!!”命令再次执行刚刚执行过的那个Shell命令。

别名



别名命令alias是按照Shell命令标准格式所写命令行的缩写，用于减少输入。

格式：**alias** [别名= '标准Shell命令行']

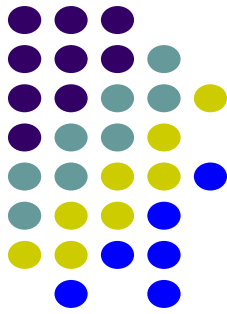
功能：查看和设置别名。

1.查看别名

无参数的**alias**命令查看用户可使用的所有别名命令以及对应的标准Shell命令。

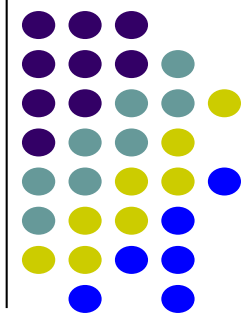
```
[root@Linux root]# alias
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-ti
lde'
[root@Linux root]#
```

别名



- **Shell规定：当别名与标准Shell命令同名时，别名命令优先于标准Shell命令执行。也就是说。**
- **如果要使用标准的Shell命令，需要在命名名前添加 “\” 字符。**

设置别名



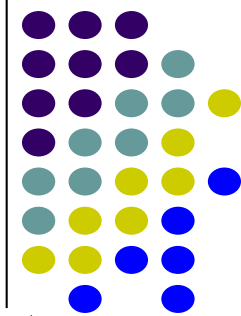
- 使用带参数的alias命令可以设置用户别名。
- 设置别名时，“=”的两边不能有空格，并在标准Shell命令行的两端使用单引号。

例如：设置别名命令ctab,其功能是在vi中打开/etc/inittab文件。

```
[root@ Linux root] # alias ctab='cat /etc/inittab'
```

- 别名的有效期仅持续到用户退出登录为止。
- 若希望别名命令每次登录都有效，可以将alias命令写入用户主目录中的.bashrc文件中。

自动补全



- **自动补全**是指用户在输入命令行时不需要输入完整的命令，只需要输入前几个字母，系统就自动找出匹配的文件或命令。利用[Tab]键实现自动补全功能。
- 自动补全**文件或目录名**
- 自动补全**命令名**