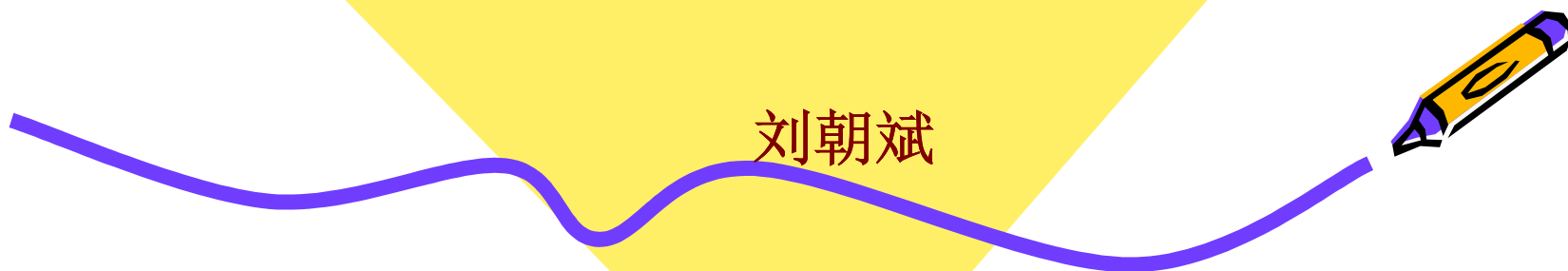




Unix/Linux体系及编程

Chap5 字符操作环境

刘朝斌



主要内容

➤ 使用终端

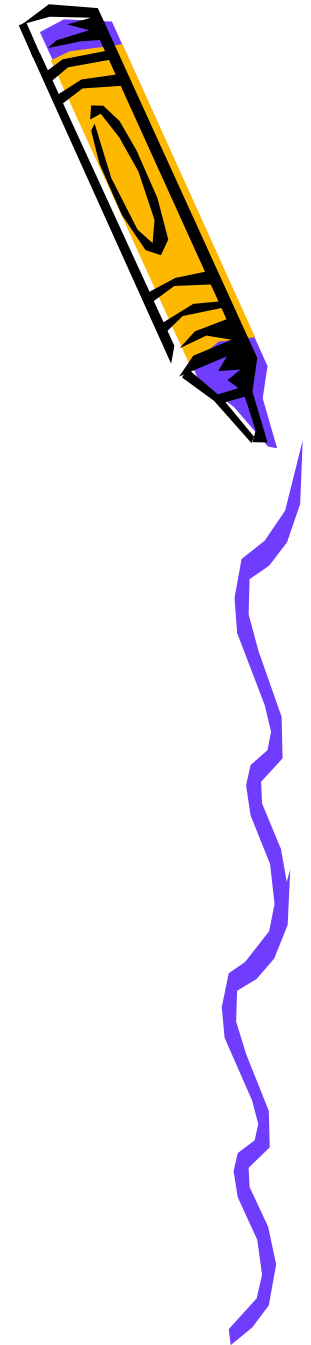
➤ 使用**Shell**

➤ 字符编辑器**vi**

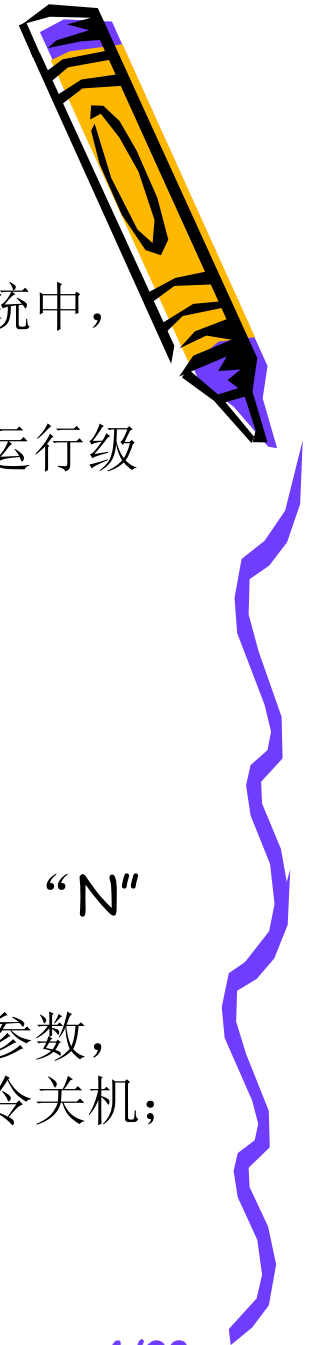


使用终端

- 虚拟终端简介
 - › 终端启动
 - › 虚拟控制台：ctrl+Fn (F1-F6)
 - › 从虚拟控制台启动X Window (**startx**)
- Shell的历史
 - › 系统中的**shell**
 - › 启动**shell**
- 转换系统模式
 - › 设置启动模式 **/etc/inittab (man) -Redhat**
 - › 运行时设为命令行模式 **init3**
 - › **cat /etc/init/rc-sysinit.conf (Ubuntu):**
DEFAULT_RUNLEVEL=5



运行级别



- *0: 关机级别
 - *1: 单用户运行级别, 运行`rc.sysinit`和`rc1.d`目录下的脚本
 - *2: 多用户, 但系统不会启动NFS, 字符模式, 在有些linux系统中, 级别2为默认模式, 具有网络功能, 如ubuntu.debian
 - *3: 多用户, 字符模式, 系统启动具有网络功能, redhat常用运行级别
 - *4: 用户自定义级别
 - *5: 图形界面模式, redhat常用运行级别
 - *6: 重启级别
 - *S: 单用户运行级别, 只运行`rc.sysinit`文件
- **runlevel**: `runlevel` 显示上次的运行级别和当前的运行级别, “N”表示没有上次的运行级别。 **man runlevel**
 - `init [0123456]`: (切换运行级别) 即在 `init` 命令后跟一个参数, 此参数是要切换到的运行级的运行级代号, 如: 用 `init 0` 命令关机; 用 `init 6` 命令重新启动。



使用Shell

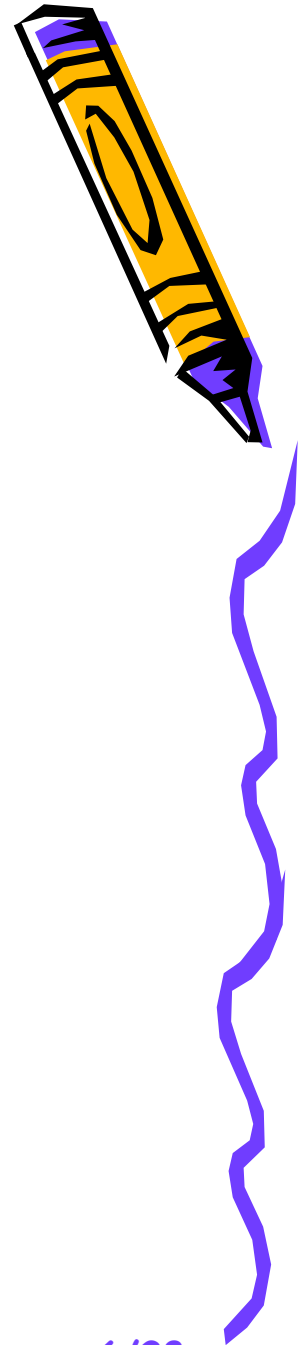
- 为什么使用Shell
- shell环境变量
- 显示环境变量 `echo; env; printenv; set`
`cat /etc/environment`
- 修改环境变量

`env` - run a program in a modified environment



文件相关命令

- ls
- cd
- pwd
- rm
- mkdir
- rmdir
- cp
- mv
- lpr/ lpq/lprm



Linux lpr命令

卢欢添加 了解就行



lpr(line printer, 按行打印)实用程序用来将一个或多个文件放入打印队列等待打印。

lpr 可以用来将资料送给本地或是远端的主机来处理。

语法

```
lpr [ -P printer ]
```

参数：

- -p Printer: 将资料送至指定的打印机 Printer，预设值为 lp。

实例

下面的命令行将在名为mailroom的打印机上打印report文件：

```
$ lpr -P mailroom report
```

使用一条打印命令可打印多个文件，下面的命令行在名为laser1的打印机上打印3个文件：

```
$ lpr -P laser1 05.txt 108.txt 12.txt
```

Linux lpq命令

卢欢添加 了解就行



Linux lpq命令用于查看一个打印队列的状态，该程序可以查看打印机队列状态及其所包含的打印任务。

语法

lpq [l] [P] [user]

参数说明：

- -P 指定一个打印机，否则使用默认打印机或环境变量PRINTER指定的打印机
- -l 打印组成作业的所有文件的信息。。

实例

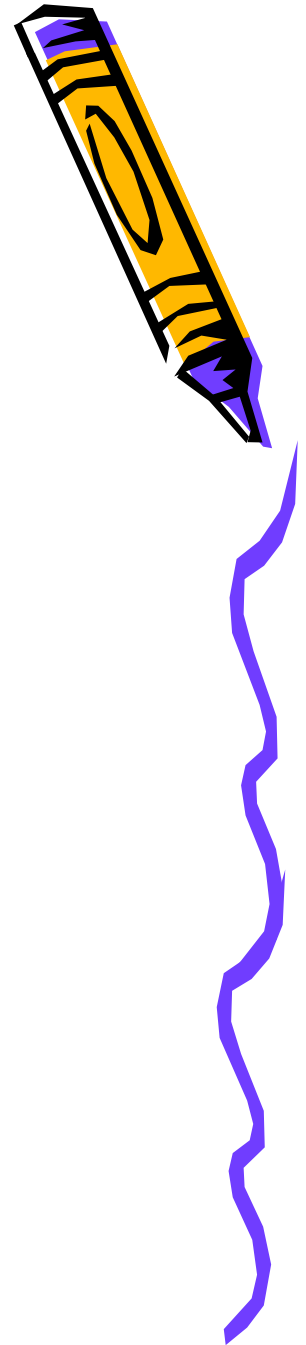
为系统默认的打印机printer的一个空队列。

```
$ lpq
printer is ready
no entries
```

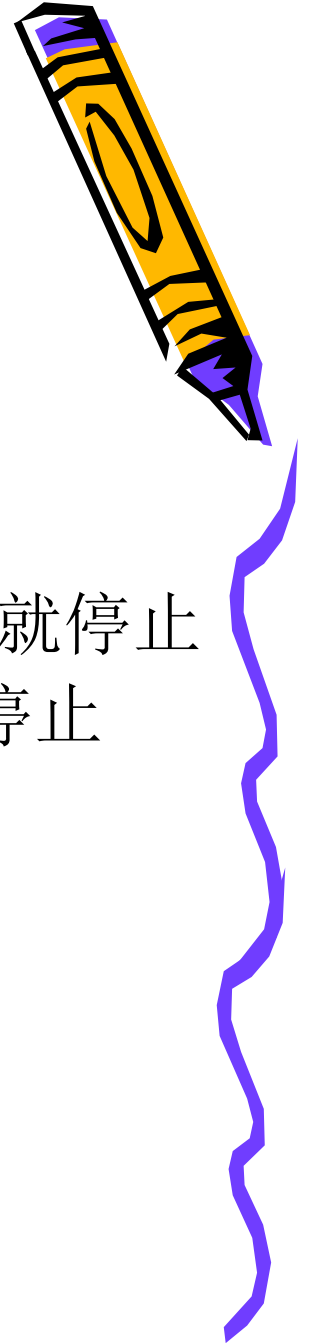

文件相关命令 (con't)

- **echo** echo - display a line of text
- **cat** cat - concatenate files and print on the standard output
- **head** head - output the first part of files
- **tail** tail - output the last part of files
- **more** more - file perusal filter for crt viewing
- **less** less - opposite of more
- **locate** locate - find files by name

➤ 简单 **grep** 命令



man; 命令历史和Tab 自动补全



- **man:** manual pager
- 使用多重命令
 - `date;tom;uname`
 - `date&&tom&&uname` / 第一个操作数是false就停止
 - `tom||date||uname` / 第一个操作数是true就停止
- 查看命令历史 **history**
- **!**
- **Tab:** 补齐（命令&参数）
- 上下箭头



管道: pipelines

- `$ echo "cherry apple peach"`
cherry apple peach
- `$ echo "cherry apple peach" | tr " " "\n"`
cherry
apple
peach
- `$ echo "cherry apple peach" | tr " " "\n" | sort`
apple
cherry
peach
- `$ echo "cherry apple peach" | tr " " "\n" | sort -r`
peach
cherry
apple



重定向

➤ 使用重定向

- > > //输出重定向
- > 2> //错误输出重定向
- > < //输入重定向 **wc <1.txt**

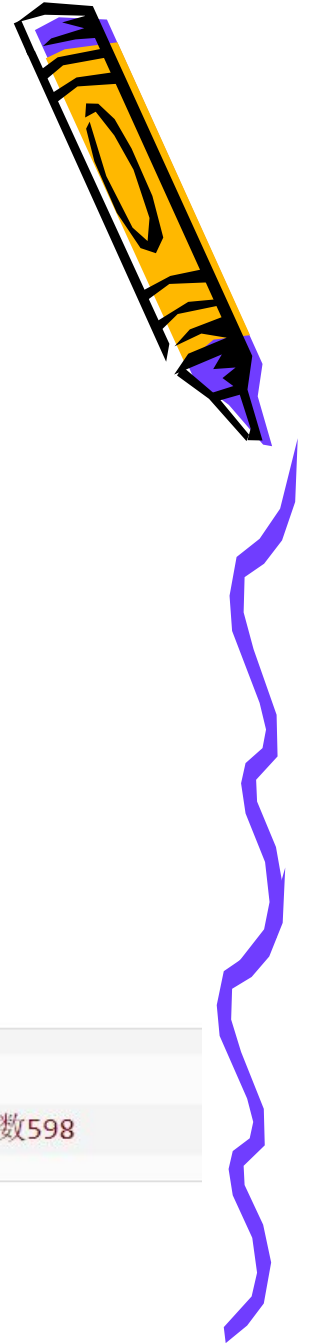
➤ 后补(追加)重定向: >>

- > >> //输出追加重定向
- > 2>> //错误输出追加重定向

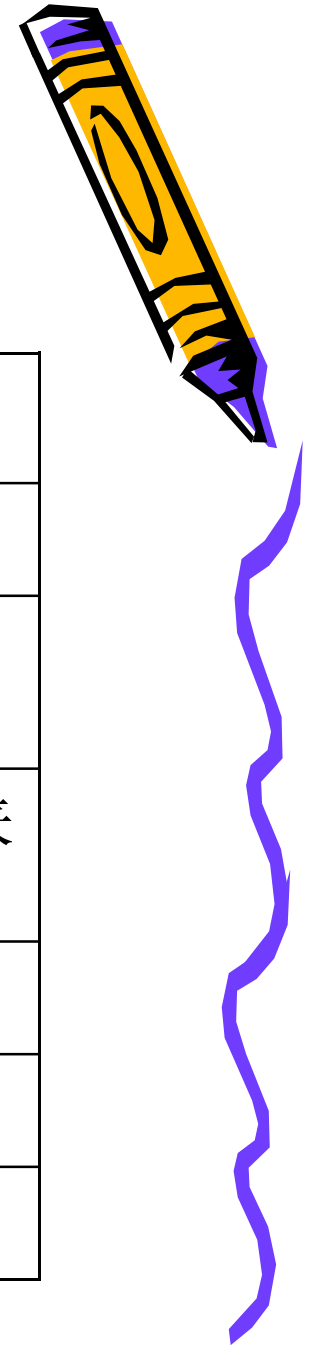
➤ example

➤ unistd.h

```
$ wc testfile          # testfile文件的统计信息
3 92 598 testfile      # testfile文件的行数为3、单词数92、字节数598
```



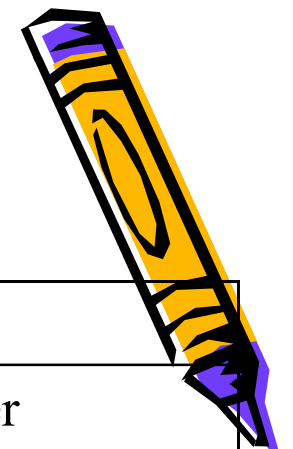
通配符和简单正则



通配符	含 义
*	匹配所有字符
?	匹配字符串中的一个字符（例如 sneaker?.txt）
\	转义符，与特殊字符一起使用来表示某个特殊字符
*	匹配 “*” 字符
\?	匹配 “?” 字符
\)	匹配 “)” 字符



权限符号含义

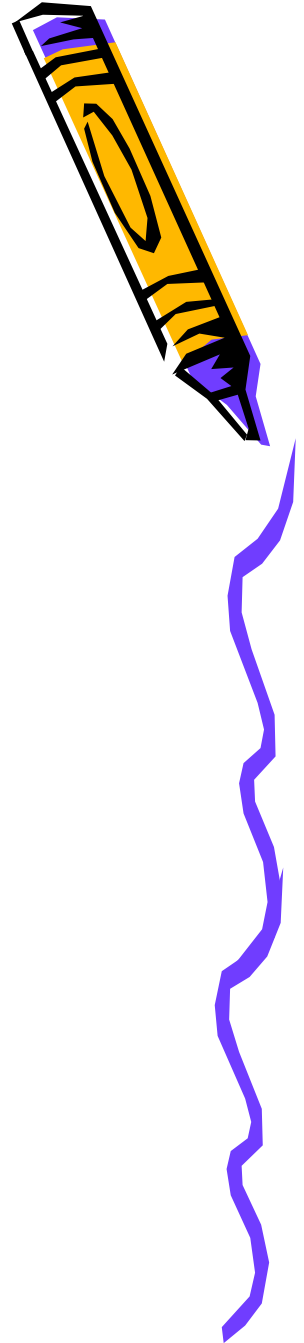


功 能	符 号	含 义
用 户 身 份	u	拥有文件的用户（所有者） user
	g	所有者所在的组群 group
	o	其他人（不是所有者或所有者的组群） others
	a	每个人或全部（u、g、和o）
权 限	r	读取权
	w	写入权
	x	执行权
执 行	+	添加权限
	-	删除权限
	=	使它成为唯一权限

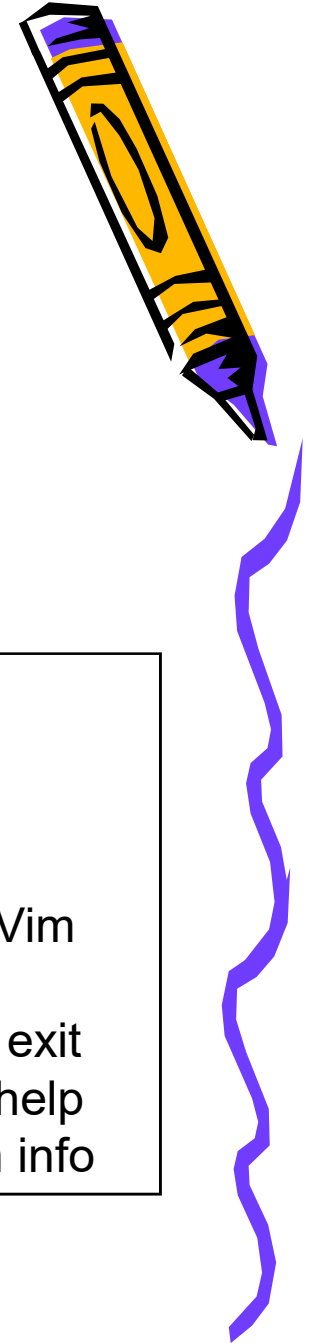


所有权和许可权限

- 权限类型
- 使用 **chmod** 命令改变权限
 - > **chmod u+x filename**
 - > **chmod a+x,o-w filename**
 - > **chmod u=rwx,g=rw,o=x filename**
 - > **chmod **octal** filename**
 - > **chmod -R 755 filename**



5.3 字符编辑器vi



- 字符模式配置及系统维护工作
- vi的执行与离开

- 启动vi
- 离开vi

VIM – Vi Improved

version 5.8.7

y Bram Moolenaar et al.

Vim is freely distributable

type : help Uganda<Enter> if you like Vim

type : q<Enter> to exit

type : help<Enter> or <F1> for on-line help

type : help version5<Enter> for version info

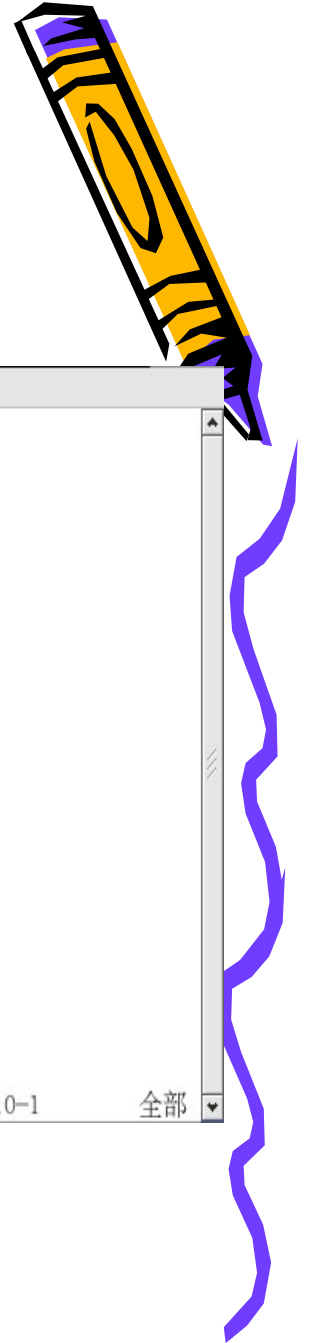


5.3.2 vi的操作模式

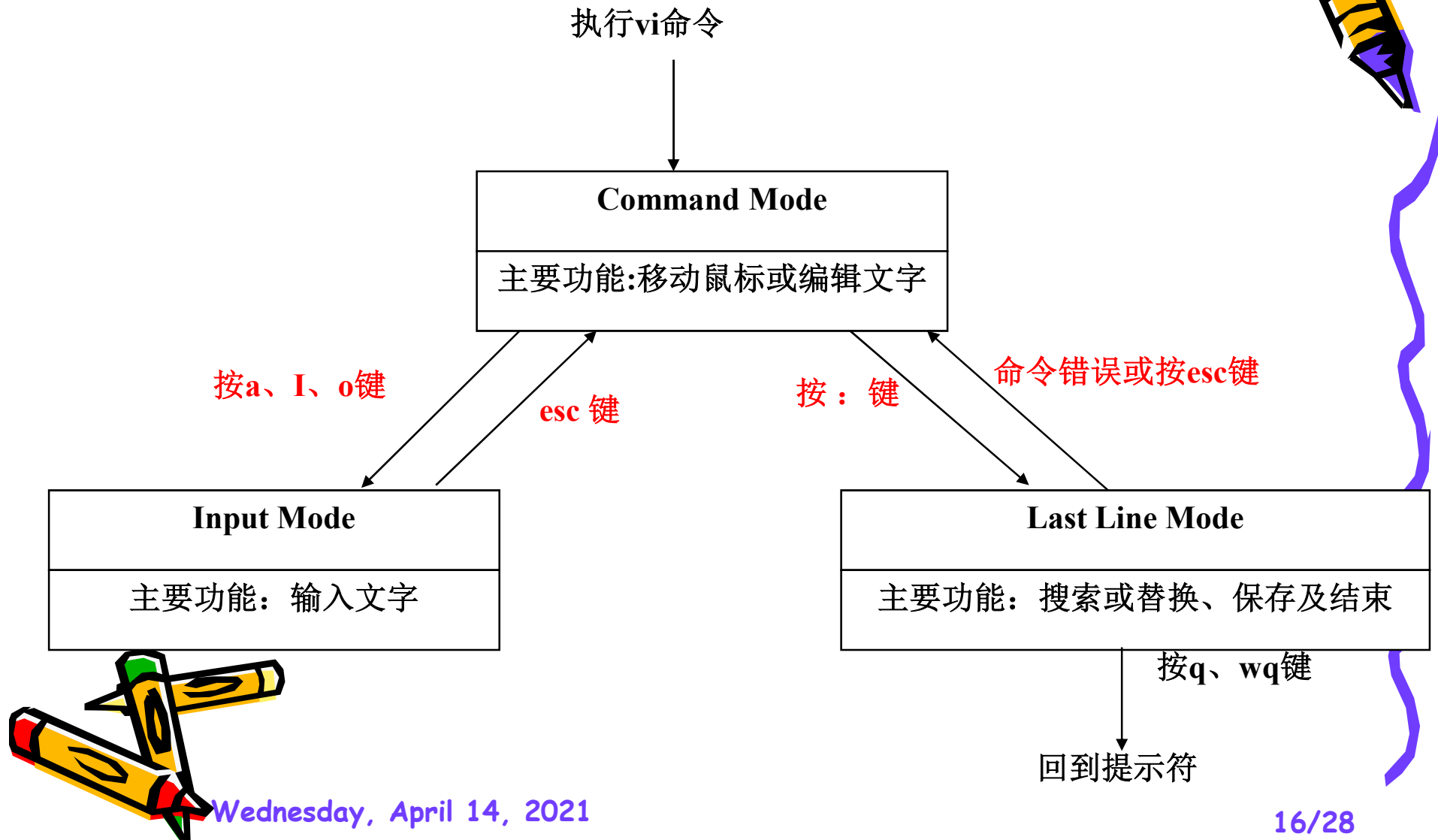
- 界面
 - › 编辑区
 - › 命令区
- 操作模式
 - › Command Mode
 - › Insert Mode
 - › Last Line Mode

编辑区

命令区



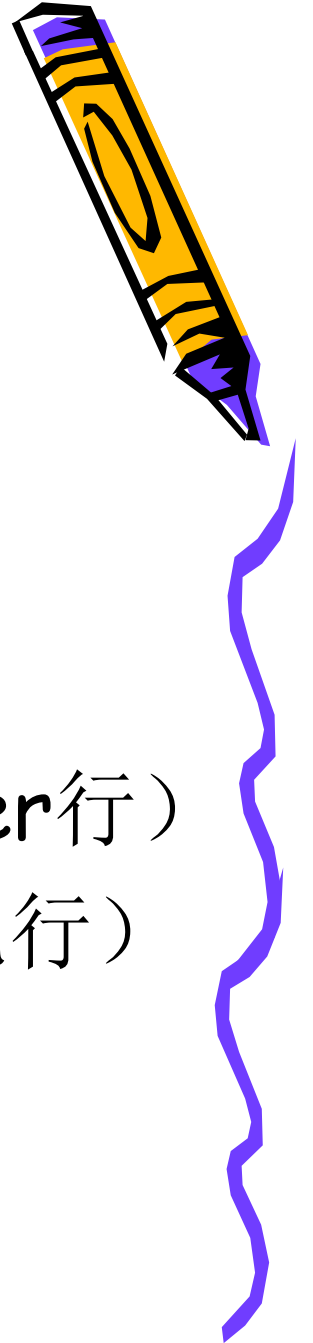
vi的3种模式转换关系



进入vi(m)

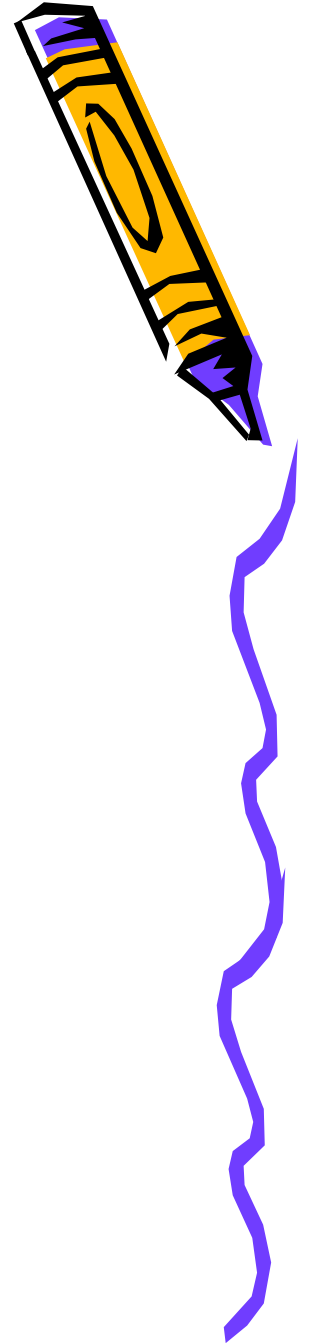
- vi
- vi filename
- vi + filename （光标在该最后一行）
- vi +number filename （光标在number行）
- vi +/pattern filename （光标在匹配行）

一定要注意，+紧跟着/



5.3.3 Command Mode相关命令

- 移动光标
- 复制文字
- 删除文字
- 位置显示及其他命令



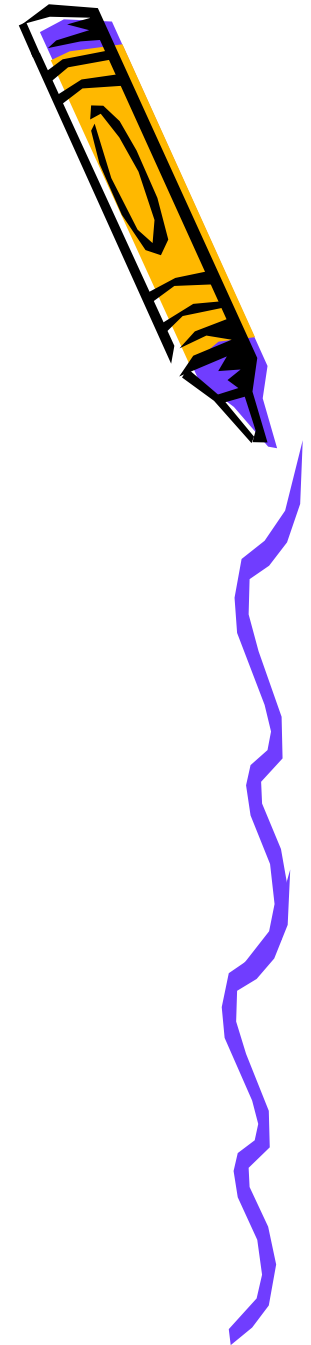
光标移动类命令

- **h, l, j, k**
- **space** 光标右移一个字符
- **Backspace** 光标左移一个字符
- **Enter** 光标下移一行
- **)** 光标移至句尾
- **(** 光标移至句首
- **0** (数字零) 光标移至当前行首
- **\$** 光标移至当前行尾
- **nG** 光标移至第**n**行首
- **n\$** 光标下移到第**n-1**行尾
- **n+** 光标下移**n**行
- **n-** 光标上移**n**行



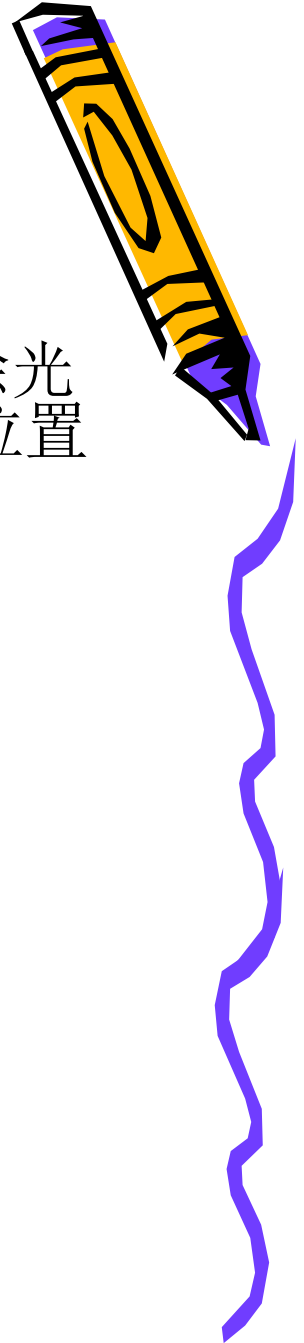
屏幕翻滚类命令

- **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
- **zz**: 将当前行移动到屏幕中央



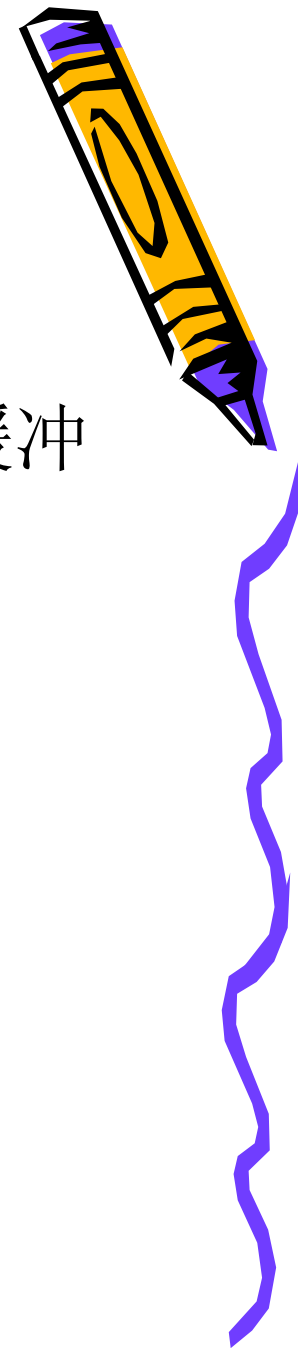
删除相关命令

- **x**或**X**: 删除一个字符, **x**删除光标后的, 而**X**删除光标前的 **#x**, **#X** 例如, **[6x]**表示删除光标所在位置的“后面”6个字符。
- **dd** 删除光标所在行。
- **#dd** 从光标所在行开始删除**#**行
- **dw** 删除一个**word**
- **ndw** 删除光标处开始及其后的**n-1**个字
- **do** 删至行首 数字**0**
- **d\$** 删至行尾
- **u** 撤销上一次操作
- **Ctrl+r** 反撤销



复制相关命令

- **yw** 将光标所在之处到字尾的字符复制到缓冲区中。
- **#yw** 复制**#**个字到缓冲区
- **yy** 复制光标所在行到缓冲区。
- **#yy**, 复制**#**行
- **p** 将缓冲区内的字符贴到光标所在位置
- **c[n]w**: 改写光标后**1(n)**个词

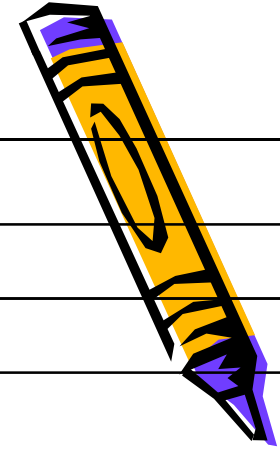


搜索及替换命令

- `/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/` 将当前行中第一个`p1`用`p2`替代
- `: s/p1/p2/g` 将当前行中所有`p1`均用`p2`替代
- `: m,ns/p1/p2/g` 将第`m`至`n`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
- `: %s/p1/p2/g`



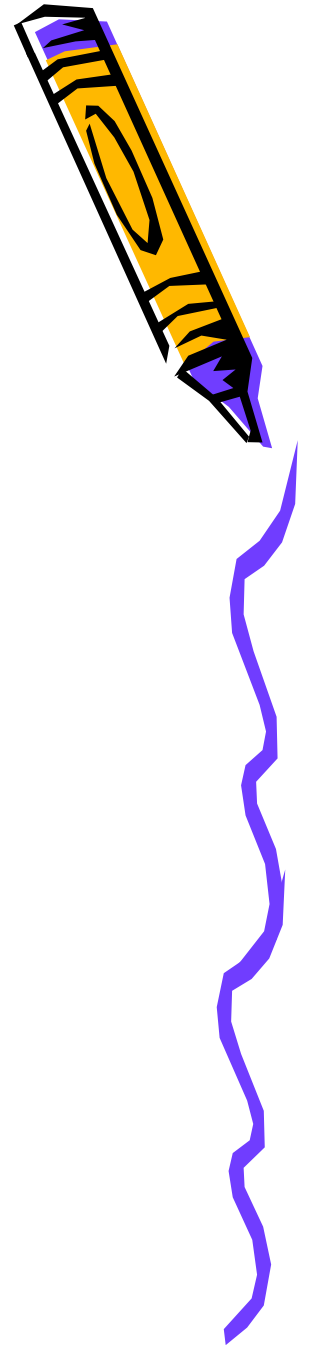
5.3.4 Last Line Mode命令



命 令	说 明
<code>:e</code>	在vi中编辑时，还可以使用e创建新的文件。
<code>:r</code>	加载新文件
<code>:w</code>	写文件
<code>:w!</code>	如果想写只读文件的话，可以使用w!强制写入文件。
<code>:q</code>	表示退出vi，如果不想保存退出，需要使用命令q!强制退出。
<code>:wq/ZZ</code>	将修改文件存盘，然后退出
<code>:set nu</code>	Set可以设置vi的某些特性,想取消使用命令 <code>set nonu</code>
<code>n1,n2 co n3</code>	将n1行到n2行之间的内容拷贝到第n3行下
<code>n1,n2 m n3</code>	将n1行到n2行之间的内容移至到第n3行下
<code>:Sex</code>	水平分割一个窗口，浏览文件系统，:q退出
<code>:Vex</code>	垂直分割一个窗口，浏览文件系统，:q退出 两次ctrl+w切换
<code>:set spell</code>	检查拼写
<code>!cmd</code>	执行shell下的cmd命令, r!cmd



- vimtutor
- vimdiff



Linux系统有7个运行级别(runlevel)

Linux系统有7个运行级别(runlevel)

运行级别0: 系统停机状态, 系统默认运行级别不能设为0, 否则不能正常启动

运行级别1: 单用户工作状态, root权限, 用于系统维护, 禁止远程登陆

运行级别2: 多用户状态(没有NFS)

运行级别3: 完全的多用户状态(有NFS), 登陆后进入控制台命令行模式

运行级别4: 系统未使用, 保留

运行级别5: X11控制台, 登陆后进入图形GUI模式

运行级别6: 系统正常关闭并重启, 默认运行级别不能设为6, 否则不能正常启动

```
hope@ubuntuhope:~/Chapter5$ runlevel
N 5
hope@ubuntuhope:~/Chapter5$
```

Linux locate命令用于查找符合条件的文档，他会去保存文档和目录名称的数据库内，查找合乎范本样式条件的文档或目录。

一般情况我们只需要输入 **locate your_file_name** 即可查找指定文件。

语法

```
locate [-d ] [--help] [--version] [范本样式...]
```

实例

查找 passwd 文件，输入以下命令：

```
locate passwd
```

搜索 etc 目录下所有以 sh 开头的文件：

```
locate /etc/sh
```

```
hope@ubuntuhope:~/Chapter5$ locate 1.t
/lib/systemd/system/runlevel1.target
/lib/systemd/system/runlevel1.target.wants
/usr/share/doc/git/RelNotes/1.5.0.1.txt
/usr/share/doc/git/RelNotes/1.5.1.1.txt
/usr/share/doc/git/RelNotes/1.5.1.txt
/usr/share/doc/git/RelNotes/1.5.2.1.txt
```

忽略大小写搜索当前用户目录下所有以 r 开头的文件：

```
locate -i ~/r
```

附加说明

locate 与 find 不同: find 是去硬盘找，locate 只在 /var/lib/slocate 资料库中找。

locate 的速度比 find 快，它并不是真的查找，而是查数据库，一般文件数据库在 /var/lib/slocate/slocate.db 中，所以 locate 的查找并不是实时的，而是以数据库的更新为准，一般是系统自己维护，也可以手工升级数据库，命令为：

Head 命令

head 命令可用于查看文件的开头部分的内容，有一个常用的参数 `-n` 用于显示行数，默认为 10，即显示 10 行的内容。

命令格式：

```
head [参数] [文件]
```

参数：

- `-q` 隐藏文件名
- `-v` 显示文件名
- `-c<数目>` 显示的字节数。
- `-n<行数>` 显示的行数。

实例

要显示 runoob_notes.log 文件的开头 10 行，请输入以下命令：

```
head runoob_notes.log
```

显示 notes.log 文件的开头 5 行，请输入以下命令：

```
head -n 5 runoob_notes.log
```

显示文件前 20 个字节：

```
head -c 20 runoob_notes.log
```

Linux tail 命令

tail 命令可用于查看文件的内容，有一个常用的参数 **-f** 常用于查阅正在改变的日志文件。

tail -f filename 会把 filename 文件里的最尾部的内容显示在屏幕上，并且不断刷新，只要 filename 更新就可以看到最新的文件内容。

命令格式：

```
tail [参数] [文件]
```

参数：

- -f 循环读取
- -q 不显示处理信息
- -v 显示详细的处理信息

实例

要显示 notes.log 文件的最后 10 行，请输入以下命令：

```
tail notes.log          # 默认显示最后 10 行
```

要跟踪名为 notes.log 的文件的生长情况，请输入以下命令：

```
tail -f notes.log
```

此命令显示 notes.log 文件的最后 10 行。当将某些行添加至 notes.log 文件时，tail 命令会继续显示这些行。显示一直继续，直到您按下 (Ctrl-C) 组合键停止显示。

显示文件 notes.log 的内容，从第 20 行至文件末尾：

```
tail -n +20 notes.log
```

显示文件 notes.log 的最后 10 个字符：

```
tail -c 10 notes.log
```

Linux more命令

Linux more 命令类似 cat，不过会以一页一页的形式显示，更方便使用者逐页阅读，而最基本的指令就是按空白键（space）就往下一页显示，按 b 键就会往回（back）一页显示，而且还有搜寻字串的功能（与 vi 相似），使用中的说明文件，请按 h。

语法

```
more [-dlfpcsu] [-num] [+/pattern] [+linenum] [fileNames..]
```

参数：

- -num 一次显示的行数

实例

逐页显示 testfile 文档内容，如有连续两行以上空白行则以一行空白行显示。

```
more -s testfile
```

从第 20 行开始显示 testfile 之文档内容。

```
more +20 testfile
```

常用操作命令

- Enter 向下n行，需要定义。默认为1行
- Ctrl+F 向下滚动一屏
- 空格键 向下滚动一屏

Linux less 命令

less 与 more 类似，less 可以随意浏览文件，支持翻页和搜索，支持向上翻页和向下翻页。

语法

```
less [参数] 文件
```

1、查看文件

```
less log2013.log
```

4、浏览多个文件

```
less log2013.log log2014.log
```

说明：

输入：n后，切换到 log2014.log

输入：p 后，切换到log2013.log

Linux umask命令

Linux umask命令指定在建立文件时预设的权限掩码。

umask可用来设定[权限掩码]。[权限掩码]是由3个八进制的数字所组成，将现有的存取权限减掉权限掩码后，即可产生建立文件时预设的权限。

语法

```
umask [-S][权限掩码]
```

实例

使用指令"umask"查看当前权限掩码，则输入下面的命令：

```
$ umask                                #获取当前权限掩码
```

执行上面的指令后，输出信息如下：

```
0022 hope@ubuntuhope:~/Chapter5$ umask
0002
```


小结&习题



➤ 思考题

- (1) 为什么使用字符操作界面，使用其有什么优点？
- (2) 终端与**Shell**是什么关系？
- (3) 系统命令是如何通过**Shell**进入操作系统内核的？

➤ 上机题

- (1) 在虚拟终端上查看系统文件结构。
- (2) 使用**vi**编辑一个个人简介 “学号.txt”。
- (3) 切换**vi**的三种模式。
- (4) 用编辑完成，不保存文件退出。

