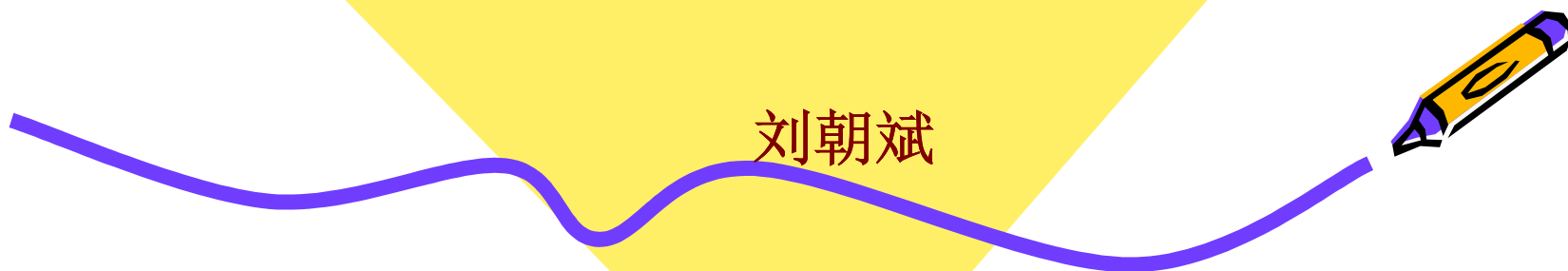




Unix/Linux体系及编程

Chap5 字符操作环境

刘朝斌



主要内容

➤ 使用终端

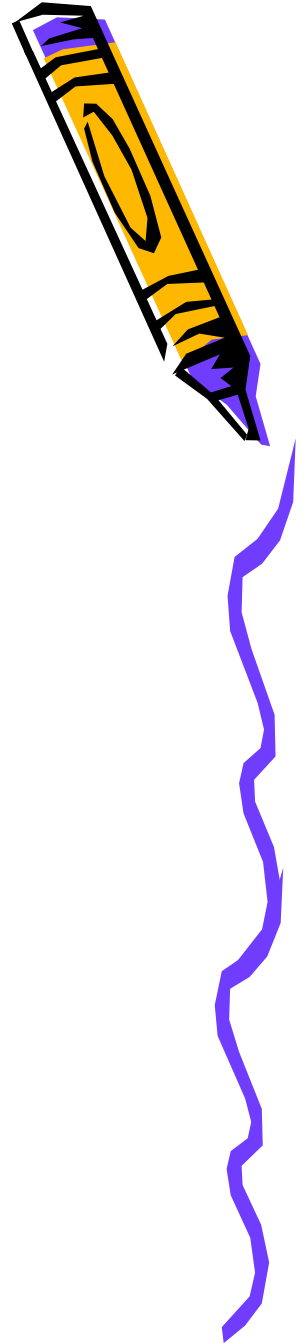
➤ 使用**Shell**

➤ 字符编辑器**vi**

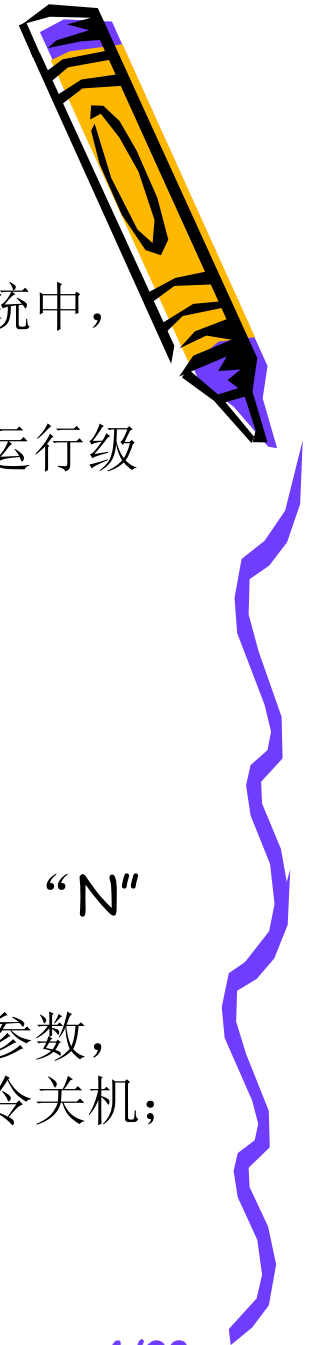


使用终端

- 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

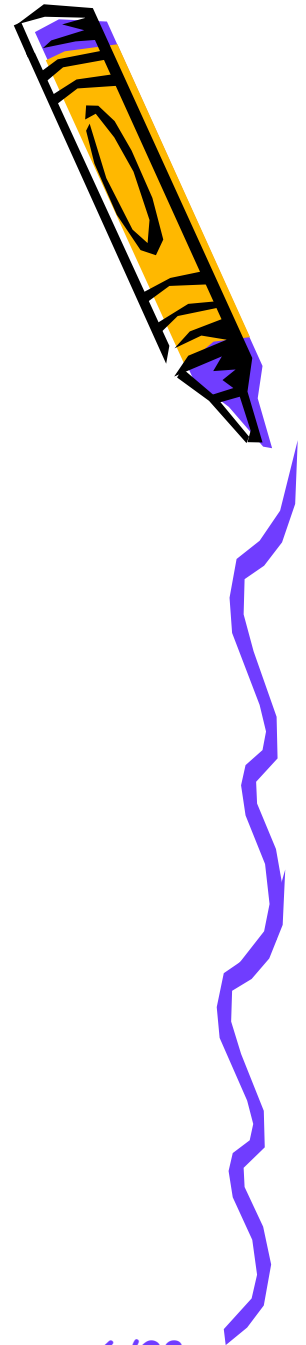


文件相关命令

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



Wednesday, April 14, 2021



文件相关命令 (**con't**)

➤ echo

➤ cat

➤ head

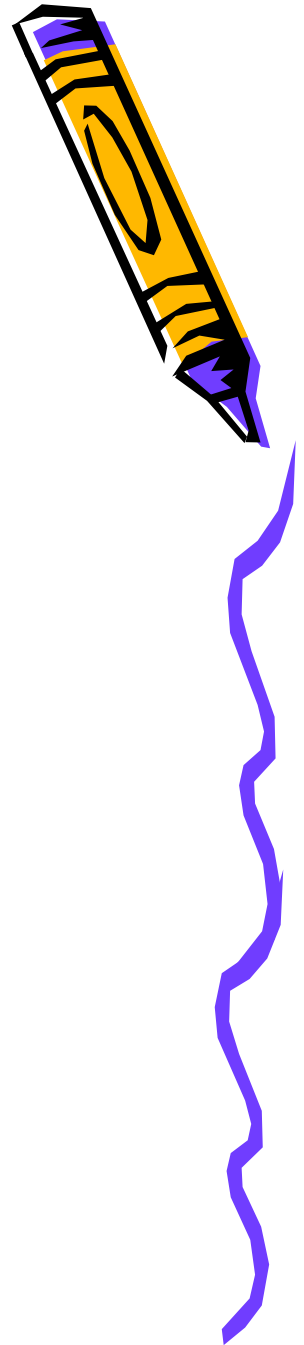
➤ tail

➤ more

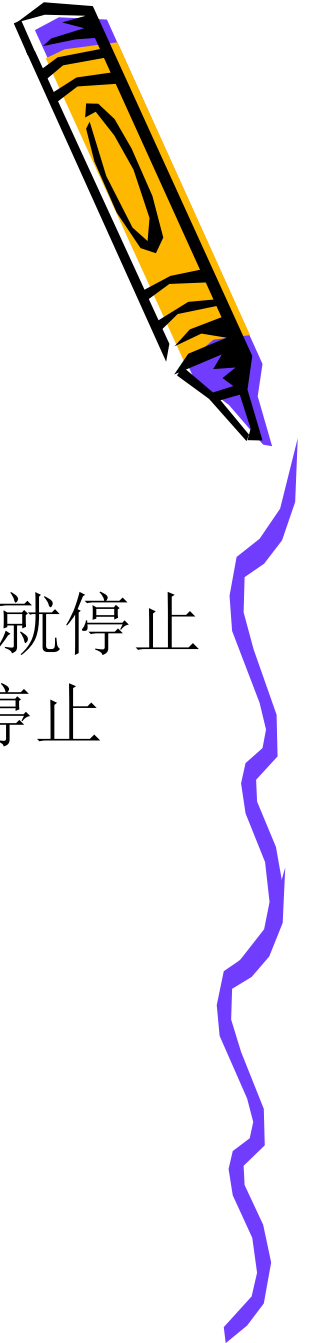
➤ less

➤ locate

➤ 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

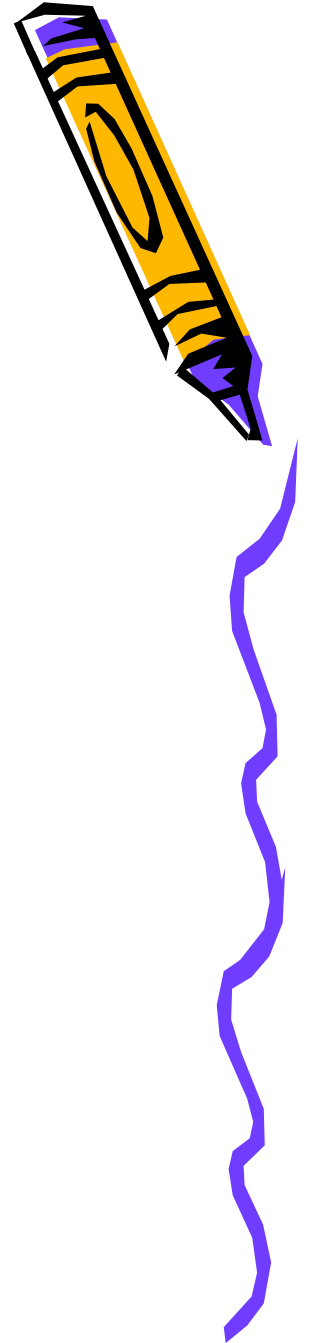
- > > //输出重定向
- > 2> //错误输出重定向
- > < //输入重定向 **wc <1.txt**

➤ 后补(追加)重定向: >>

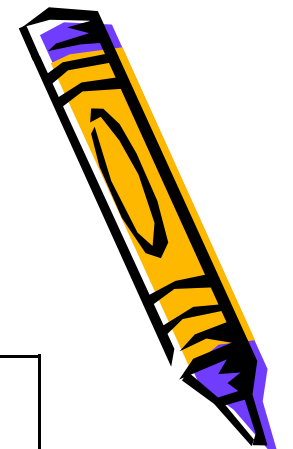
- > >> //输出追加重定向
- > 2>> //错误输出追加重定向

➤ example

➤ unistd.h



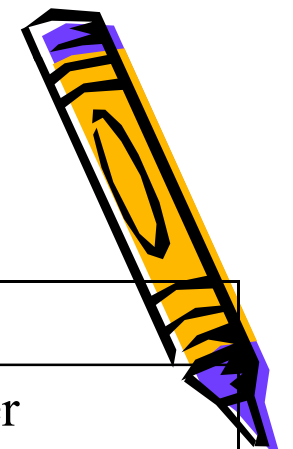
通配符和简单正则



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

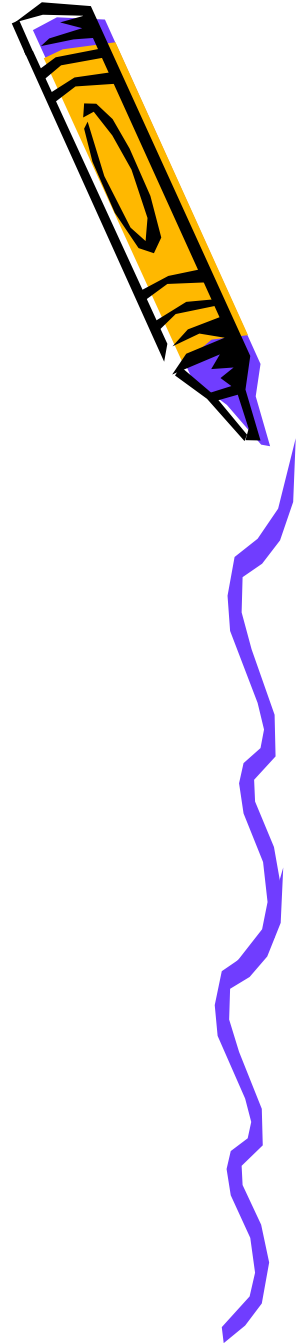


功 能	符 号	含 义
用 户 身 份	u	拥有文件的用户（所有者） user
	g	所有者所在的组群 group
	o	其他人（不是所有者或所有者的组群） others
	a	每个人或全部（u、g、和o）
权 限	r	读取权
	w	写入权
	x	执行权
执 行	+	添加权限
	-	删除权限
	=	使它成为唯一权限

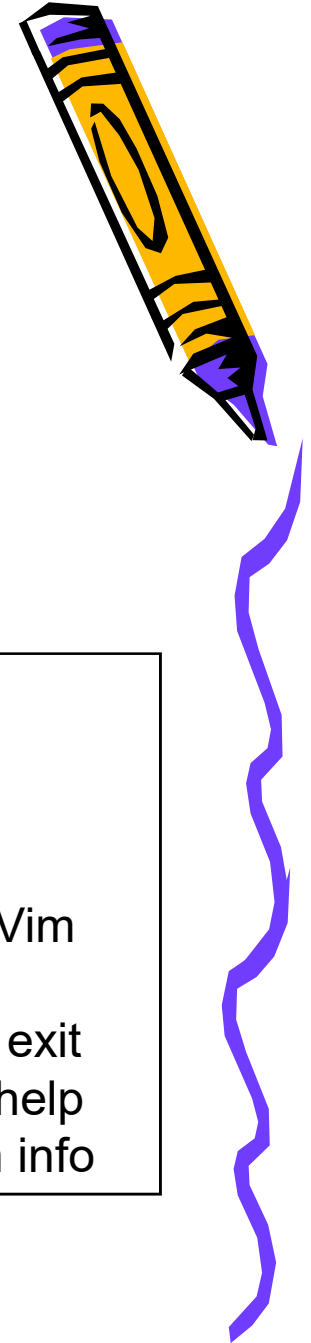


所有权和许可权限

- 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

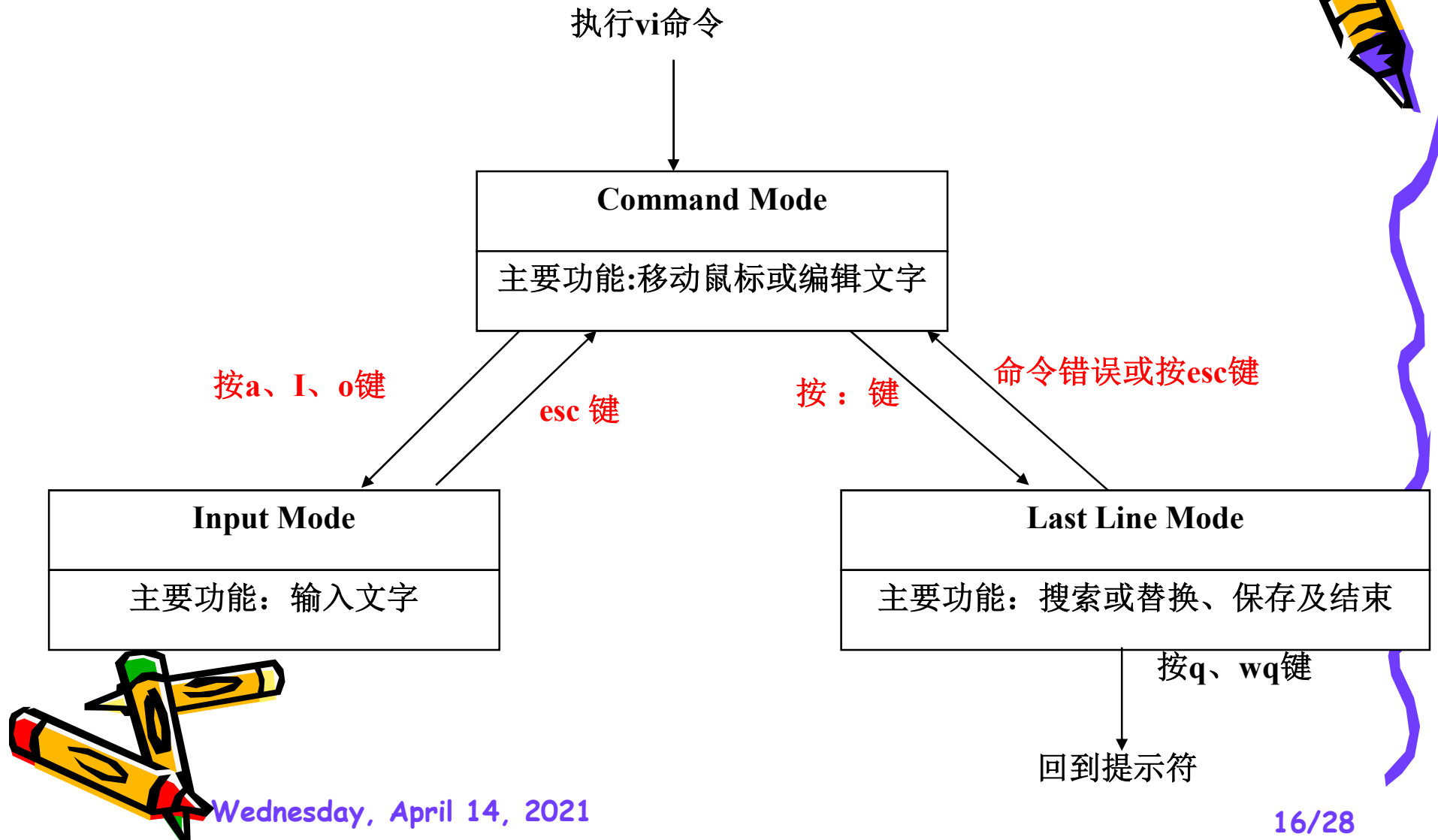
- 界面
 - › 编辑区
 - › 命令区
- 操作模式
 - › Command Mode
 - › Insert Mode
 - › Last Line Mode

编辑区

命令区

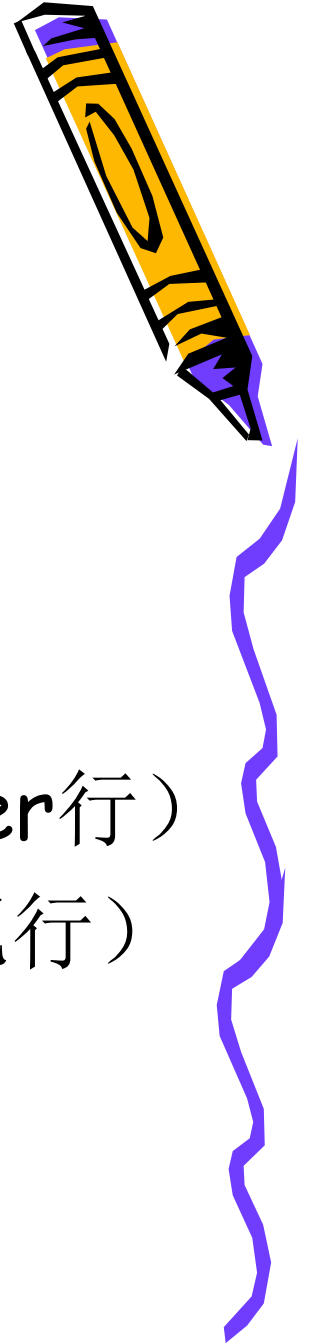


vi的3种模式转换关系



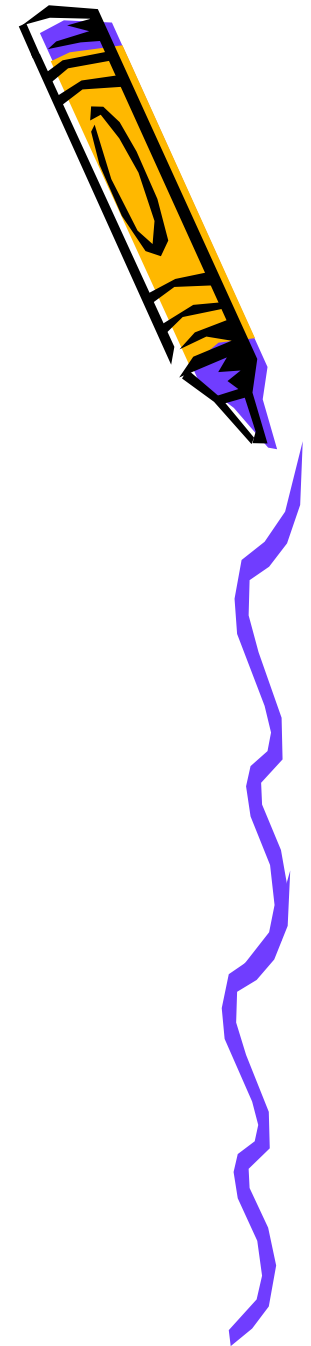
进入vi(m)

- 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）



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



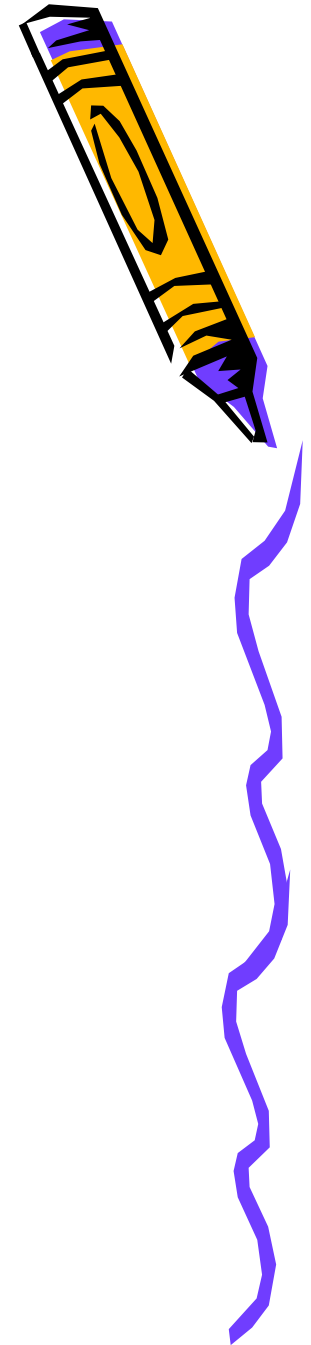
光标移动类命令

- **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



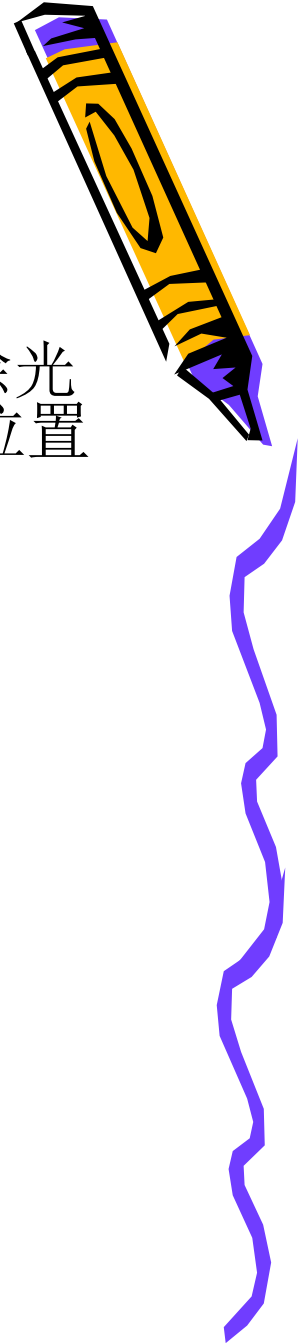
屏幕翻滚类命令

- **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
- **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`



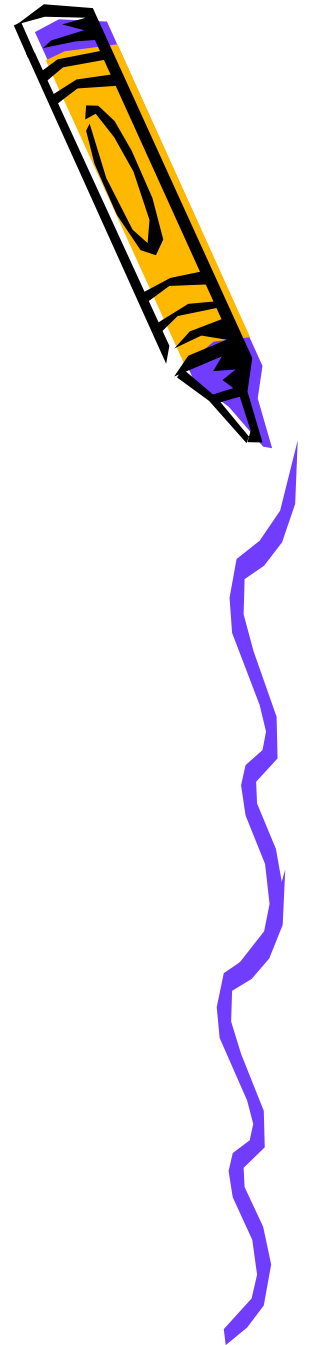
5.3.4 Last Line Mode命令



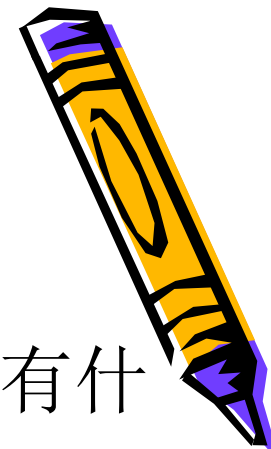
命令	说明
<code>:e</code>	在vi中编辑时，还可以使用e创建新的文件。
<code>:r</code>	加载新文件
<code>:w</code>	写文件
<code>:w!</code>	如果想写只读文件的话，可以使用w!强制写入文件。
<code>:q</code>	表示退出vi，如果不想保存退出，需要使用命令q!强制退出。
<code>:wq/ZZ</code>	将修改文件存盘，然后退出
<code>:set nu</code>	Set可以设置vi的某些特性,想取消使用命令 <code>set nonu</code>
<code>n1,n2 co n3</code>	将n1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>	
<code>!cmd</code>	执行shell下的cmd命令, <code>!cmd</code>



- vimtutor
- vimdiff



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。

