

第5章

Vim编辑器



内容

①

Vim编辑器简介

Vim进入与退出

Vim的两种模式

Vim光标移动

Vim文本删除

Vim文本修改

②

Vim文本复制

Vim文本合并

撤销与恢复

重复命令

组合命令

Vim缓存寄存器

ex模式

常用Vim设置

内容

①

Vim编辑器简介

Vim进入与退出

Vim的两种模式

Vim光标移动

Vim文本删除

Vim文本修改

②

Vim文本复制

Vim文本合并

撤销与恢复

重复命令

组合命令

Vim缓存寄存器

ex模式

常用Vim设置

Vim文本复制命令（yank/put）

- y: 复制，需要与移动光标的命令组合使用
- Y: 复制光标所在的行
- p: 将复制的内容放到光标后面
- P: 将复制的内容放到光标前面



文本复制命令：例子

- yy: 复制一行
- 2yy/y2y: 复制光标所在行开始的2行
- yw: 复制光标所在位置至词尾
- 2p: 将2份复制的内容粘贴到光标后面

Vim文本合并命令（Joint）

- J: 合并光标所在行及其下一行



撤销与恢复命令

- u: 撤销上一步操作
- U: 把当前行恢复成它被编辑之前的状态
- Ctrl+r: 恢复上一步撤销之前的状态



重复命令

➤ . : 重复最后一步编辑操作



组合命令

- **5dd:** 删除从光标所在行开始的后面5行
- **10w:** 往后移动10个单词
- **8h:** 光标向左移动8个字符
- **yj:** 复制光标所在行及其下一行
- **dgg:** 删除到文件首
- **c\$:** 删除光标所在字符至行尾，并进入输入模式
- **y3j:** 复制光标所在行开始的3行.....

数字 操作 光标移动命令

Vim缓存寄存器

Vim中有9种类型的寄存器，寄存器的主要功能就是缓存操作过程中删除、复制、搜索等的文本内容

1. **未命名寄存器** `""`: Vim的默认寄存器，文本来源命令: `d/x/c/s/y`
2. **10个数字命名寄存器** `"0 - "9`: 缓存yank (0) 和delete (1-9) 行操作 (如dd) 命令产生的文本
3. **1个非行删除内容缓存寄存器** `"-`: 缓存delete在非行操作 (如dw) 时产生的文本
4. **26*2个字母命名寄存器** `"a - "z / "A - "Z`: 完全由用户指定内容的寄存器
5. **4个只读寄存器** `". "% "# "`:
6. **表达式寄存器** `"=`: 使用Vim强大的表达式功能
7. **GUI选择寄存器** `"* "+ "~`: Vim缓存在GUI中选择的文本
8. **黑洞寄存器** `"_`: 类似Linux中的/dev/null文件，只进不出，可用来滤掉影响默认寄存器的内容
9. **最后搜索模式寄存器** `"/`: 缓存Vim最后一次的搜索内容

缓存寄存器使用

- “ayy 复制当前行到a寄存器
- “byy 复制当前行到b寄存器
- “ap 将a寄存器的内容复制到当前行后面
- “bp 将b寄存器的内容复制到当前行后面

ex模式

- Vim底层是交互式行编辑器ex，Vim中保留了ex模式
- 在Vim中输入Q或gQ进入ex模式，在ex模式中输入vi或visual进入vi模式
- Vim在命令模式下输入冒号，即可调用ex命令（末行模式）
- 与sed命令类似，ex命令由位置和操作两部分组成，比如1p，1是位置第1行，p是print操作（print操作符p可以省略）。如果省略位置，表示操作的是当前行

常用ex命令

ex命令格式： 位置+操作

位置类型：

- 绝对位置：如1（第1行）； 15,23（第15到23行）
- 相对位置：根据当前位置得到的相对位置，如.+2（当前行后面第2行）
- 搜索到的位置：如/protein/（光标所在行后面第一个包含protein的行）

常见操作：

操作	缩写	例子
• print	p	:1p
• delete	d	:10,20d
• yank	ya或y	:5,8ya
• put	pu	:pu
• move	m	:10,15m20
• copy	co或t	:25,28t40
• substitute	s	:1,\$s/protein/Protein/g

相对位置

+号和-号放在数字中间的时候，分别表示前一个数字的后面或前面多少行，如“**.+10**”表示当前行的后面第**10**行，如：

- `::-3,.p` 显示当前行前面第3行到当前行
- `:26,$m.-2` 将26行到文件尾的内容移动到当前行前面第二行的后面

搜索位置

模式匹配到的行作为操作的位置，如：

- `:/pattern/d` 删除第一个包含模式的行
- `:/pattern/+d` 删除第一个包含模式的行的下一行
- `:/pattern1/,/pattern2/d` 删除第一个包含模式1的行到第一个包含模式2行之间的内容

命令举例

- `:3,18d` 删除第3到18行
- `:150,180m23` 将150到180行的内容移动到23行后面
- `:23,28co100` 将23到28行内容复制到100行后面

ex中，“.”表示当前行（可以省略），\$表示最后一行，%表示所有行（相当于1,\$），如：

- `:d` 删除当前行
- `:%t$` 复制所有行，然后放到末尾
- `:20,.m$` 把20到当前行移动到文件尾

ps: 在ex里，可以用`:=`得到当前的行号，用`:=`得到文件的总行数，用`:/pattern/=`则打印出模式第一次出现的行号

ex使用缓存寄存器

- :160,224ya a 复制160到224行内容到a寄存器
- :240,254ya b 复制240到254行内容到b寄存器
- :pu a 把a的内容复制到光标所在行后面
- :pu b 把b的内容复制到光标所在行后面

多文件编辑：不同窗口打开多个文件

➤ `vi a.pl b.pl`

同时打开两个文件

➤ `:n`

切换至下一个文件

➤ `:N`

切换至上一个文件

或

➤ `:e another_file`

打开另一个文件

➤ `:e #`

在两个文件间切换

➤ `:ls`

列出打开的文件

多文件编辑：同一窗口打开多个文件

Vim分屏：

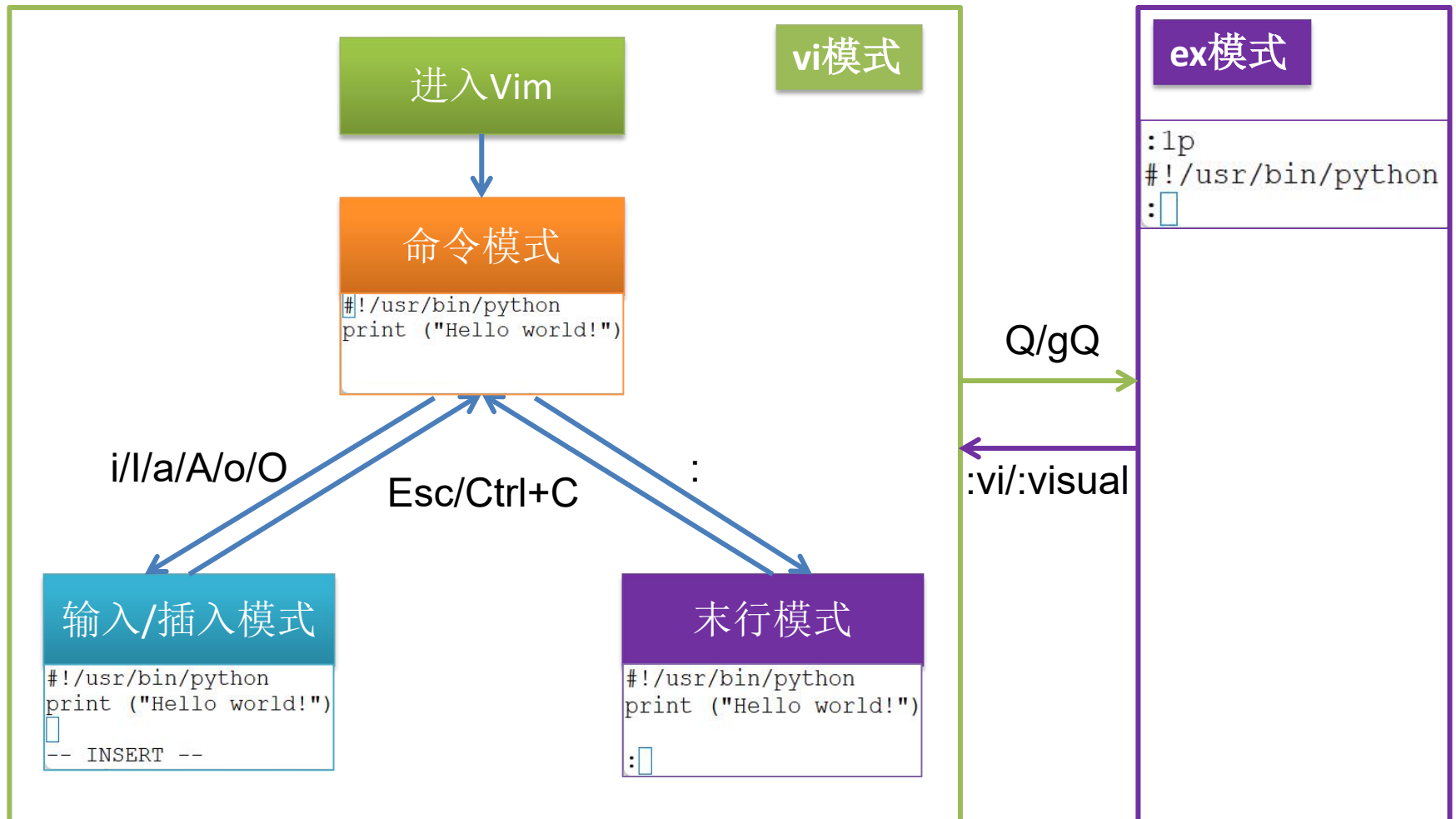
- `ctrl + w, v / :vsplit / :vsp / :vs` 左右分屏
- `ctrl + w, s/n / :split / :sp` 上下分屏
- `ctrl + w, w/p/h/j/k/l` 在不同的屏间移动光标
- `ctrl + w, q` 关掉光标所处分屏

- `$ vi -O2 file1 file2` 左右分屏打开多个文件
- `$ vi -o2 file1 file2` 上下分屏打开多个文件

常用Vim设置

- `:set nu/nonu` 显示/不显示行号
- `:set tabstop=n` 制表符显示为n个空格
- `:set autoindent/ai` 设置自动缩进
- `:set smartindent/si` 设置智能自动缩进
- `:set list` 制表符和行尾分别用`^I`,`$`显示

Vim编辑器模式总结



课堂作业7

课后作业6

- (1) 利用`zcat`和重定向将文件
`/home/pub/seq/at_protein_partial.fa.gz`解压到本目录下的文件
`at_protein_partial.fa`;
- (2) 用`vi`打开文件`at_protein_partial.fa`，做如下操作并保存：
 - a) 将第81行到第85行移动到第37行后面；
 - b) 将第90行到第92行复制到第130行后面；
 - c) 将所有的RR（大写）替换成rr（小写）。



The End