**vi编辑器入门**

* vi编辑器属于全屏编辑器的一种，基于行编辑器ex，具有字处理程序的灵活性和简单易用的特性。
* vi有两种基本的工作模式：命令（编辑）模式和文本输入模式。命令模式下的所有输入都会被解释为命令，不会被显示在屏幕上但是会被执行；文本输入模式下，按键不被解释为命令执行，而是作为文本写入到用户文件中。启动vi默认会进入命令模式。
* 启动vi编辑器的命令：vi [文件名]
* 使用vi编辑器打开该文件，若没有此名称的文件，则vi会创建一个新文件。
* 移动光标的命令：

想移动光标必须使vi处在命令模式并保证大写锁定光比

| **命令键** | **功能** |
| --- | --- |
| h/← | 光标左移一个字符 |
| k/↑ | 光标上移一行 |
| j/↓ | 光标下移一行 |
| l/→ | 光标右移一个字符 |
| $ | 光标移动至当前行尾 |
| 0 | 光标移动至当前行首 |
| w | 光标右移一个字 |
| b | 光标左移一个字 |
| e | 光标移至字尾 |
| [Return] | 光标移至下一行首 |
| [Spacebar] | 光标向右移动一格 |
| [Backspace] | 光标向左移动一格 |

* 切换模式的命令

| **命令键** | **功能** |
| --- | --- |
| i | 命令模式→文本输入模式，并向光标的左侧输入正文 |
| I | 命令模式→文本输入模式，并从光标所在行开头输入 |
| a | 命令模式→文本输入模式，并向光标的右侧输入正文 |
| A | 命令模式→文本输入模式，并从光标所在行末尾输入 |
| o | 命令模式→文本输入模式，在光标下一行添加新行且光标位于新行的开头。 |
| O | 命令模式→文本输入模式，在光标上一行添加新行且光标位于新行的开头。 |
| Esc | 文本输入模式→命令模式 |

* 文本修改命令：

当vi处于命令模式下，可以替换、删除字符或者撤销操作。

| **命令键** | **功能** |
| --- | --- |
| x | 删除光标位置字符 |
| dd | 删除光标所在行 |
| u | 撤销最近修改 |
| U | 撤销对当前行的修改 |
| r | 替换光标位置字符 |
| R | 替换从光标开始的字符并切换为文本输入模式 |
| . | 重复上一次修改 |

多数修改命令可以配合数字使用，如5x删除5个连续字符，3dd删除三行

* 搜索命令：

vi编辑器提供了/和？分别在文件中向前、向后搜索指定的字符串。使用该搜索命令时，必须保证vi编辑器处于命令模式。例如向前搜索UNIX。命令为： /UNIX ；光标会移动到第一个搜索结果处，按n键可以移动到下一个搜索结果。

* 退出vi编辑器：

| **命令键** | **功能** |
| --- | --- |
| :wq | 保存文件，退出vi编辑器 |
| :w | 保存文件，不退出vi编辑器 |
| :q | 退出vi编辑器 |
| :q! | 不保存文件，退出vi编辑器 |
| ZZ | 保存文件，退出vi编辑器 |

* Linux下的vim编辑器是vi编辑器的增强版本，在vim编辑器中的命令模式下输入help [Return]，就会显示一个通用帮助命令。

**Vi 编辑器的高级用法**

**一、更多有关vi编辑器的知识**

**vi编辑器与ex编辑器的相互切换**

* 命令模式下:Q进入ex编辑器
* ex编辑器下，输入vi返回vi编辑器

**启动Vi编辑器**

1、基本命令的扩充

* shell下直接输入vi，启动vi编辑器，不提供文件名
* 保存，命令模式下:w保存文件
  + 此命令不退出vi，如若需退出，键入:wq
  + 如果是新建的文件，则vi警告没有当前文件名的文件，需要附加文件名参数如:w(q) filename
  + 如果上述方法保存文件存在文件名冲突，则需要使用:w(1)!

2、vi的启动选项

* 只读模式：vi -R filename，shell下亦可使用view命令通过vi查看文件
* 退出只读模式：命令模式下:q 或者:q!（强退）
* 命令选项: vi -c [command] filename

3、编辑多个文件

vi 启动的时候可以使用多个文件名作为参数，启动后按n(!)即可启动下一个编辑文件。

* 编辑另一个文件：在vi的命令模式下，:e filename 即打开新的一个文件（类比word）
* 读取另一个文件：在vi的命令模式下，:r filename 即读取一个新的文件，并将其引入到党建编辑的文件中
* 写入另一个文件：在vi的命令模式下，:[指定行号] w(!)写入的指定文件(覆盖性写入)

**二、重排文本**

| **按键** | **功能** |
| --- | --- |
| d | 删除指定位置的文本，并将其保存在缓冲区里 |
| y | 复制指定位置的文本到缓冲区 |
| P(大写) | 将缓冲区的内容放到当前光标的位置之上 |
| p | 将缓冲区的内容放到当前光标的位置之下 |
| c | 删除文件并使vi进入文本输入模式 |
| dd(p) | 删除当前行，(再按p，将前面被删除的行放到当前的行下) |
| dd(P) | (再按P，将前面被删除的行放在当前的行上) |
| (v)yy(p,P) | (v选择指定的行)复制当前的(被选定的)行 |

**三、vi操作符的域**

* 许多的vi命令只能在一个文本块(字符，字，行，句)上操作
* 命令的表示：命令=操作符+域
* *vi没有具体的域控制键表示整行，如果需要将一行作为一个命令的域，可以按两次操作符按键。*
* 部分的与控制键

| **域** | **功能** |
| --- | --- |
| $ | 标识域为光标位置开始到行尾 |
| 0 | 标识域为光标位置开始到行首 |
| e或w | 标识域为光标位置开始到当前字尾 |
| b | 标识域为光标位置到当前字首 |

**使用删除操作符和域控制键**

* 删除从光标位置开始到当前行尾的文本: 命令模式下按:d$
* 删除从光标位置开始到当前行首的文本: 命令模式下按:d0
* 删除从光标位置后的一个字: 命令模式下按:dw
* 删除从光标位置后的n个字: 命令模式下按:(n)d$
* 删除到字尾：命令模式下按:de
* 删除到前一个字符的字首: 命令模式下按:db

**使用复制操作符和域控制键**

* 复制当前光标位置开始到当前行尾的文本：命令模式下按:y$
* 复制当前光标位置开始到当前行首的文本：命令模式下按:y0

**四、在vi中使用缓冲区**

vi中有多个临时存储的缓冲区。用于保存用户副本的临时文件。

**缓冲区类型**

* 数字编号缓冲区
* 命名缓冲区

**数字编号缓冲区**

* vi编辑器使用的9个临时缓冲区
* 1-9从上至下，构成一个缓冲区栈
* 缓冲区可以保存任意大小的文本
* 缓冲区的访问："(缓冲区编号)p，将指定缓冲区的内容复制到光标处

**字母编号缓冲区**

* 使用26和字母命名的缓冲区
* vi不会默认使用，故可以提供给用户自由操作
* 使用方式同数字编号缓冲区
* "(缓冲区编号)来指定当前操作所关联的缓冲区(vi会默认使用数字缓冲区的1)

**五、光标定位键**

* vi的翻页操作符

| **按键** | **功能** |
| --- | --- |
| Ctrl + d | 将光标向下移动到文件尾，通常每次移动12行 |
| Ctrl + u | 将光标向上移动到文件头，通常每次移动12行 |
| Ctrl + f | 将光标向下移动到文件尾，通常每次移动24行 |
| Ctrl + b | 将光标向上移动到文件头，通常每次移动24行 |

* 显示当前行号：命令模式下 Crl + g

**六、定制vi编辑器**

vi 可以在命令模式下键入: set xxx 来设置vi 编辑器的一些参数

**选项的格式**

选项分为三类

* 布尔值：set X 或者 set no X ，设置某一些选项
* 数字: 如 set X=12 ，将12 值赋予X
* 串：set X = Y , 将Y的值赋值给X

**设置vi的环境**

set对环境的修改是临时的，仅对当前的会话有效

* 部分vi环境选项

| **选项** | **缩写** | **功能** |
| --- | --- | --- |
| autoindent | ai | 将新行与前一行的行首对齐 |
| ignorecase | ic | 在搜索选项中忽略大小写 |
| magic |  | 允许搜索时使用特殊字符 |
| number | nu | 显示行号 |
| report |  | 同通知用户上一个命令影响的行号 |
| scroll |  | 设定Ctrl + d 命令翻动的行数 |
| shiftwidth | sw | 设置缩进的空格数，和autoindent一起使用 |
| showmode | smd | 在屏幕的右下角显示vi的编辑模式 |
| terse |  | 缩短错误信息 |
| wrapmargin | wm | 设置有边界为指定的字符数 |

**缩写和宏**

* 缩写
  + 设置缩写，在命令模式下使用:ab 缩写名 命令名
  + 取消缩写，在命令模式下使用:unab 缩写名
  + 显示缩写，在命令模式下使用:ab
* 宏
  + 将按 键序列(指令)映射到 指定按键序列上
  + 创建宏，在命令模式下使用:map 宏名 按键序列
  + map 在unix 下的创建是临时的

**.exrc 文件**

* 用户在vi编辑器下的所有设置选项都是临时的，如果需要永久修改，在需要在.exrc文件中保存选项设置
* vi 在启动的时候会拣择当前目录下是否存在.exrc 如果存在，就将.exrc里面的内容设置为选项
* 直接在.exrc下输入 设置环境的指令即可

**七、其他的vi命令**

**运行shell命令**

* 在vi下使用!提示vi此命令为shell命令
* 常用的命令
  + :! ls , 列出当前目录下的文件
  + :! who , 显示当前登陆的用户
  + :! pwd, 显示当前工作的目录的路径
  + : r ! date , 显示当前的时间并将其放在光标后
  + : r ! cal x xxxx , 读取指定的日历到光标后
  + : ! vi filename , 启动vi的另一个副本来编辑指定文件

**行链接**

* 将光标移动到前一行的行尾，按j键会自动连接两行

**搜索和替换**

vi使用的较好的搜索命令是 / 和 ?

* 命令模式下，:/text 查找指定文本
* 命令模式下，输入newtext 将 text 替换
* 命令模式下，输入n查找下一个文本
* 命令模式下，按.重复上一次的修改
* 命令模式下，按: ? text 从当前文本向上搜索，寻找第一个text

**文件恢复选项**

当文件正在编辑的时候，vi突然崩溃可以进行恢复。

* vi -r filename 对某文件启动恢复选项
* vim下 文件启动后，在命令模式下 : recover