

vi 编辑器

正则表达式

描述字符串匹配规则的表达式称为正则表达式。它常用于 shell 命令和其他工具中，例如 grep 和 Makefile。以下字符是具有特定意义的正则表达式的算符：

- 代表任何一个字符；
- [] 方括号内的字符表示仅选其中之一；
- 连字符用于方括号内，表示字符范围；
- \ 反斜杠消除其后字符的算符意义；
- * 表示前面字符的零次或多次重复；
- + 表示前面字符的一次或多次重复；
- ? 表示前面字符的零次或一次重复；
- | 表示或者是前一字符或者后一字符；
- ^ 表示行首，在方括号内表示排除；
- \$ 表示行尾；
- () 优先运算符；

正则表达式就是表示字符匹配模式。在编辑器的字符串替换命令中，利用它就可以完成复杂的字符串匹配操作。以下是一些正则表达式的例子：

- abc 匹配字符串 “abc”；
- ab*c 匹配以字符 ‘a’ 开始、以字符 ‘c’ 结束、中间有零个或多个 ‘b’ 出现的字符串；
- a.*b 匹配以字符 ‘a’ 开始、以字符 ‘b’ 结束任何字符串；
- a\.b 匹配字符串 “a.b”；
- ab?c 匹配字符串 “ac” 或者字符串 “abc”；
- ^a 匹配以字符 ‘a’ 开始的行；
- a\$ 匹配以字符 ‘a’ 结束的行；
- ^\$ 匹配空行；
- [0-9] 匹配 ‘0’ 到 ‘9’ 之间的任一字符；
- [^a] 匹配除 ‘a’ 之外的任一字符；

vi 编辑器

1. 进入 vi

进入的方法很简单，键入如下命令便可。

```
$ vi filename
```

2. 退出 vi

在命令方式下使用下面几种退出命令：

- :q 退出。当文件已被修改，vi 将在屏幕的底行显示提示信息。
- :q! 强行退出。
- :w 回写文件但不退出；
- :wq 回写文件并退出；

:x 与 wq 相同，回写文件并退出；

3. 进入命令方式

进入 vi 后，首先处于命令方式；或在输入方式，通过键入 Esc 键进入命令方式。在命令方式下，可以进行修改、删除、移动或复制正文等编辑操作。

4. 进入输入方式

使用任何一种正文输入命令进入输入方式。在输入方式下可以输入或添加文本。

5. 正文输入命令

vi 提供几条基本的命令进入输入方式：

- a 在光标的后面开始插入正文。
- A 在光标所在行的行首插入正文。
- i 在光标的前面开始插入正文。
- I 在光标所在行的行末插入正文。
- o 在光标所在行（当前行）的下一行从其行首开始插入正文。
- O 在光标所在行（当前行）的上一行从其行首开始插入正文。

在输入方式下可以把文本输入到文件中，一行结束时按 RETURN 键，开始新的一行；如果输入时有错，则可以用 Backspace 键删除输错的字符。

6. 移动光标

为了编辑文本，必须把光标移到所编辑的地方，在 vi 编辑器中可以用键盘上的 ↓ ↑ ← → 键和四个排列在一起的键 h, j, k, l：

- | | |
|----------|-----------------------|
| ↑ 键或 k 键 | 把当前光标向上移动一行，保持光标的列位置； |
| ↓ 键或 j 键 | 把当前光标向下移动一行，保持光标的列位置； |
| → 或 l 键 | 把当前光标向右移动一个字符； |
| ← 或 h 键 | 把当前光标向左移动一个字符； |
| \$ 键 | 把当前光标移动到该行行末； |
| ^ 键 | 把当前光标移动到该行行首； |
| w 键 | 把当前光标移动到该行的下一个字的首字符上； |
| b 键 | 把当前光标移动到该行的上一个字的首字符上； |
| e 键 | 把当前光标移动到该行的该字的末尾字符上； |

在用 k, j, l, h 四个键时，可以在它们的前面加一个数字，这样在需要多次移动光标时不必多次按移动命令键。

7. 正文删除命令

在命令方式下删除正文，使用以下几条命令：

- x 删除当前光标所在的字符；
- nx 删除从当前光标开始的 n 个字符，n 为要删除的字符数；
- dw 删除当前光标所在的字；
- ndw 删除从当前光标开始的 n 个字，n 为要删除的字的个数；
- dd 删除当前光标所在行；
- ndd 删除从当前光标开始的 n 行，n 为要删除的行数。

8. 正文替换命令

在命令方式下，可以用下面一些命令替换正文。

rx 用 x 替代当前光标所在的字符。

nrx 用 x 替换 n 个字符，在替换完第 n 个字符后该命令自动停止。

注意，替换命令执行后不进入输入方式。

9. 恢复命令

恢复被修改或删除的内容。vi 的恢复命令是 u 和 U。

u 废除最近的命令，恢复被修改或删除的内容；

U 把当前行恢复到修改完它之前的状态。

10. 查找正文命令

vi 允许在文件中查找字符串，查找命令有四种：

/pattern 在缓冲区中向下查找指定的字符串 pattern；

?pattern 在缓冲区中向上查找指定的字符串 pattern；

n 重复上一次查找命令；

N 以相反的查找方向重复上一次查找命令。

注意：命令/或?和要查找的字符串 pattern 一起显示于屏幕底部的。

11. 行编辑命令

:0 定位到文件头；

:n 定位到第 n 行；

:\$ 定位到最后一行。

:set number 显示正文的行号

:set nonumber 将行号取消

:d 删去当前行；

:n d 删去第 n 行；

:n1,n2 d 删去第 n1 到第 n2 行的内容；

:n1,n2 s /old_text/new_text/[g]

在第 n1 到第 n2 行的范围内，用 new_text 替换 old_text。如果尾部有 g，替换行中每个出现的 old_text。否则，每行中只替换一次。

:n1,n2 m n3 将第 n1 到第 n2 行的内容移动到第 n3 行之后。

:n1,n2 t n3 将第 n1 到第 n2 行的内容拷贝到第 n3 行之后。

:[n] r filename 将文件 filename 的内容加入到第 n 行后。如果没给出 n，则文件内容将被加在缓冲区的末尾。

:[n1,n2] w filename 将第 n1 到第 n2 行的内容写到文件 filename 中。如果没给出 n1 和 n2，则将整个缓冲区内容写入文件中。

12. 屏幕滚动命令

<code>^f (control-f)</code>	向下滚动一整屏正文;
<code>^d (control-d)</code>	向下滚动半个屏幕的内容;
<code>^b (control-b)</code>	向上滚动一整屏正文;
<code>^u (control-u)</code>	向上滚动半个屏幕内容;