

第二章 VIM 编辑器

2.1 VIM 的两种状态

VIM(vimsual)是Linux/UNIX系列OS中通用的全屏编辑器。

vimm分为两种状态，即命令状态和编辑状态，在命令状态下，所键入的字符系统均作命令来处理，如:q代表退出，而编辑状态则是用来编辑文本资料的。当你进入vim时，会首先进入命令状态。在命令状态下，按”i”(插入)或”a”(添加)可以进入编辑状态，在编辑状态，按ESC键进入命令状态。

在命令状态下，有如下一些常用命令：

新增：

- a 从光标后面开始添加文本
- A 从光标所在行的末尾开始添加文本

插入：

- i 从光标前面开始插入文本
- I 从光标所在行的开始处插入文本

2.2 VIM 内常用的命令操作

2.2.1 删除与修改

- x 删除光标处的字符
- dd 删除光标所在的整行

- 3dd 删除光标所在行以及下面的两行（删除 3 行）
- D 删除光标到行尾的文本，常用语删除注释语句（d\$）
- yy 复制光标所在的整行
- [n]yy 从光标开始往下复制 n 行,[n]表示一个整数
- p 将复制后的文本粘贴到光标处
- u 撤销上次操作

2.2.2 光标移动

- ^ 光标移动到行首
- \$ 光标移动到行尾
- Ctrl+d 向下翻半页
- Ctrl+f 向下翻一页
- Ctrl+u 向上翻半页
- Ctrl+b 向上翻一页
- gg 光标定位到文档头
- G 光标定位到文档尾
- H 光标定位到当前页首
- L 光标定位到当前页的最后一行的行首
- w 光标往后移一个字
- b 光标往前移一个字
- [n]+ 光标向后移动 n 行,[n]表示一个整数 10+

[n]- 光标向前移动 n 行,[n]表示一个整数 10-

[n]G 光标定位到第 n 行行首, [n]表示一个整数 20G

2.2.3 查找与替换

/[str] 查找字符串 str, [str]表示要查找的字符串

回车后会加亮显示所有找到的字符串, 接着

命令 n 移动到下一个找到的字符串, 命令 N 移动到

上一个找到的字符串 eg /hello

部分替换 (只能替换光标之所在的行)

:s/[src]/[dst] /i 忽略大小写 /g 全部匹配 eg :s/hello/world/ig 替换一行

:3,6 s/[src]/[dst]/ig (3-6 行中找) eg :3,6 s/hello/world

全部替换

:%s/[src]/[dst]/g 将文档中所有 src 的字符串替换为 dst 字符串

:%s/^ //g 将文档每一行的行首的空格去掉

2.2.4 块操作

v 可视化块选择状态, 选中块之后, 可以对块进行删除(d), 复制(y), 剪切(x)

Ctrl +v 竖向选择模式, 主要用于批量注释代码, 输入步骤如下:

1、首先按 ctrl+v, 竖选选中要注释的行

2、输入 I（注意是大写的 I），然后输入//

3、再输入 esc，就会看到选中的行被注释了。

2.2.5 文档保存及退出

结束编辑：

:q 在未修改文档的情况下退出

:q! 放弃文档的修改，强行退出

:w 文档存盘

:wq 文档存盘退出

其他：

:help 命令 查看该命令的帮助提示（不常用，当不小心按 F1 时，通过:q 进行退出）

:%!xxd 十六进制模式

:%!xxd -r 返回文本模式 中间有一个空格的

如果在编辑过程中不小心按了 Ctrl+s,vim 会处于僵死状态，按 Ctrl+q 可以恢复。

执行 vim +3 main.c //表示定位到 main.c 的第 3 行

执行 vim +/printf main.c //表示定位到第一个 printf 处

在命令模式下输入:new 2.c //表示再打开一个 vim,是横向的 用 vnew 2.c 表示纵向

也可以通过:split vsplit sp vsp, 两个窗口之间进行切换的方式: Ctrl+w,w

在命令模式下输入 gg=G 可以自动对齐，可以通过 gg=10gg, 只自

动对齐第一行到第十行

2.3 VIM 外使用到的命令

sed: 管道查找替换程序

`cat a.txt | sed 's/aa/bb/' > b.txt` //将 a.txt 中的 aa 替换成 bb 并重定向输出到 b.txt 中。

Vimdiff 命令讲解