

6.1 vim

2021年9月12日 15:53

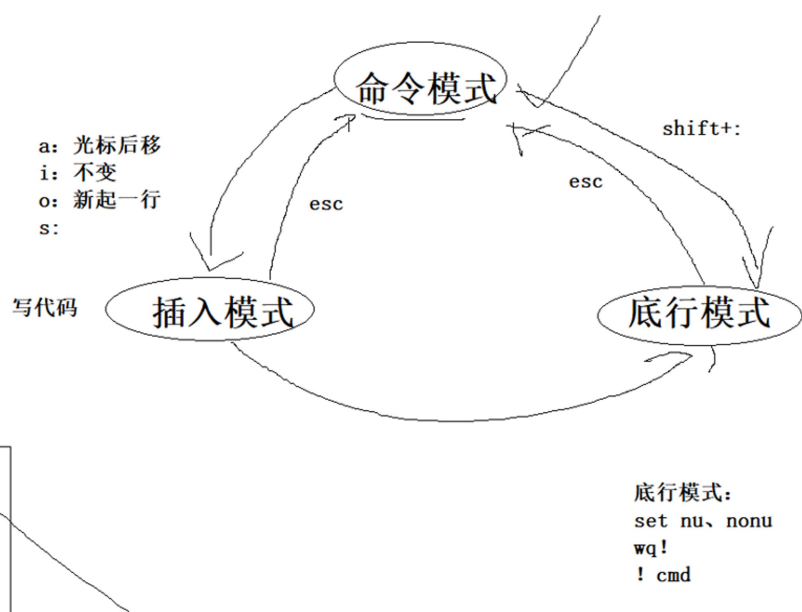
vim: 就是一款编辑器, 只负责写代码 ✓

gcc: 就是一款编译器, 只负责程序的翻译

gdb: 就是一款调试器, 只负责程序的调试

vim是一款多模式的编辑器

1. 命令模式 (默认) (输入的任何内容, 都被当做命令了)
2. 插入模式
3. 底行模式



- [正常模式]切换至[末行模式]

- 「shift + ;」, 其实就是输入「:」

- 退出vim及保存文件,在[正常模式]下, 按一下「:」冒号键进入「Last line mode」,例如:

- :w (保存当前文件)
- :wq (输入「wq」,存盘并退出vim)
- :q! (输入q!,不存盘强制退出vim)

任何模式互相切换, 都可以先切换到命令模式
无脑 `esc`

3. vim正常模式命令集 [重要点进行讲解]

• 插入模式

- 按「i」切换进入插入模式「insert mode」，按「I」进入插入模式后是从光标当前位置开始输入文件；[重要]
- 按「a」进入插入模式后，是从目前光标所在位置的下一个位置开始输入文字；
- 按「o」进入插入模式后，是插入新的一行，从行首开始输入文字。

• 从插入模式切换为命令模式

- 按「ESC」键。[重要]

• 移动光标

- vim可以直接用键盘上的光标来上下左右移动，但正规的vim是用小写英文字母「h」、「j」、「k」、「l」，分别控制光标左、下、上、右移一格
- 按「G」：移动到文章的最后[重要]
- 按「\$」：移动到光标所在行的“行尾”[重要]
- 按「^」：移动到光标所在行的“行首”[重要]
- 按「w」：光标跳到下个字的开头[重要]
- 按「e」：光标跳到下个字的字尾
- 按「b」：光标回到上个字的开头[重要]
- 按「#l」：光标移到该行的第#个位置，如：5l,56l
- 按「gg」：进入到文本开始[重要]
- 按「shift+g」：进入文本末端
- 按「ctrl」+「b」：屏幕往“后”移动一页
- 按「ctrl」+「f」：屏幕往“前”移动一页
- 按「ctrl」+「u」：屏幕往“后”移动半页
- 按「ctrl」+「d」：屏幕往“前”移动半页

• 删除文字

- 「x」：每按一次，删除光标所在位置的一个字符[重要]
- 「#x」：例如，「6x」表示删除光标所在位置的“后面（包含自己在内）”6个字符
- 「X」：大写的X，每按一次，删除光标所在位置的“前面”一个字符
- 「#X」：例如，「20X」表示删除光标所在位置的“前面”20个字符
- 「dd」：删除光标所在行[重要]
- 「#dd」：从光标所在行开始删除#行[重要]

• 复制

- 「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。[重要]
- 「#yy」：例如，「6yy」表示拷贝从光标所在的该行“往下数”6行文字。
- 「p」：将缓冲区内的字符贴到光标所在位置。注意：所有与“y”有关的复制命令都必须与“p”配合才能完成复制与粘贴功能。[重要]

• 替换

- 「r」：替换光标所在处的字符。

- 替换

- 「r」：替换光标所在处的字符。
- 「R」：替换光标所到之处的字符，直到按下「ESC」键为止。[重要]

- 撤销上一次操作

- 「u」：如果您误执行一个命令，可以马上按下「u」，回到上一个操作。按多次“u”可以执行多次回复。[重要]
- 「ctrl + r」：撤销的恢复[重要]

- 更改

- 「cw」：更改光标所在处的字到字尾处
- 「c#w」：例如，「c3w」表示更改3个字

- 跳至指定的行

- 「ctrl」+「g」列出光标所在行的行号。[重要]
- 「#G」：例如，「15G」，表示移动光标至文章的第15行行首。

4. vim末行模式命令集

在使用末行模式之前，请记住先按「ESC」键确定您已经处于正常模式，再按「:」冒号即可进入末行模式。

- 列出行号

- 「set nu」：输入「set nu」后，会在文件中的每一行前面列出行号。[重要]

- 跳到文件中的某一行

- 「#」：「#」号表示一个数字，在冒号后输入一个数字，再按回车键就会跳到该行了，如输入数字15，再回车，就会跳到文章的第15行。

- 查找字符

- 「/关键字」：先按「/」键，再输入您想寻找的字符，如果第一次找的关键字不是您想要的，可以一直按「n」会往后寻找到您要的关键字为止。

5. vim操作总结

- 三种模式

- 正常模式
- 插入模式
- 底行模式

- 我们一共有12种总模式，大家下来可以研究一下

- vim操作

- 打开，关闭，查看，查询，插入，删除，替换，撤销，复制等等操作。

命令模式，编辑操作
shift + 4 : 定位行尾
shift + 6 : 定位行首

锚点

写代码

插入模式

shift + g: 定位文本结尾
gg: 直接定位文本开始
n+shift+g: 光标定位带任意行
b、w: 自行验证
h j k l: 左, 下, 上, 右 jump: 下跳 k: king

n yy: 光标所在行进行复制
n p: 粘贴, 当前行的下一行
u: 撤销

n dd: 剪贴, 删除 (批量化)
p: 粘贴
n x: 删除光标所在的位置之后的字符
n X: 删除光标所在的位置之前的字符
shif+~: 大小写切换

在一个桌面中, 光标只有一个, 光标在哪里, 我们就操作那个屏幕
ctrl + w (double): 屏幕间进行互相切换

命令模式

底行模式:

set nu、nonu

wq!

! cmd

! man 可以搜索库函数是什么函数

配置linux

基本配置

插件式配置

```
[temp@VM-0-3-centos ~]$ pwd
/home/temp
[temp@VM-0-3-centos ~]$ ls -al
total 44
drwx----- 4 temp temp 4096 Mar 27 20:56 .
drwxr-xr-x. 23 root root 4096 Mar 27 20:54 ..
-rw----- 1 temp temp 535 Mar 27 20:56 .bash_history
-rw-r--r-- 1 temp temp 18 Apr 1 2020 .bash_logout
-rw-r--r-- 1 temp temp 193 Apr 1 2020 .bash_profile
-rw-r--r-- 1 temp temp 231 Apr 1 2020 .bashrc
drwxrwxr-x 3 temp temp 4096 Mar 27 20:54 .cache
drwxrwxr-x 3 temp temp 4096 Mar 27 20:54 .config
-rw-rw-r-- 1 temp temp 36 Mar 27 20:55 test.c
-rw----- 1 temp temp 741 Mar 27 20:56 .viminfo
-rw-rw-r-- 1 temp temp 7 Mar 27 20:56 .vimrc
[temp@VM-0-3-centos ~]$ cat .vimrc
set nu
[temp@VM-0-3-centos ~]$ vim test.c
```

Linux中有很多用户: root, 各种普通用户
/home/XXX, 默认, 一个人一个vim配置
/root

#所谓的vim基本配置, 本质是将我们的配置项放入.vimrc

设置行号：

家目录创建一个隐藏文件 .vimrc

写入set nu即可

其他的网上自己去找

搜索vim简单配置

比如

1 set nu

2 syntax on

3 set showmode

4 set showcmd

5 set mouse=a

6 set autoindent

7 set cursorline

最好的方法就是找

问问题 13127

<https://github.com/askunix/VimForCpp>

linux安装