

[快捷键](#)

- [命令行快捷键](#)
- [向终端粘贴文本](#)
- [快速搜索](#)
- [清除屏幕](#)

[Alias](#)

- [ls相关宏](#)
- [cd相关宏](#)
- [自定义](#)
- [清除屏幕](#)
- [屏蔽gdb启动时提示信息](#)
- [Just Fun](#)
- [Alias definitions](#)
- [Git alias](#)
- [易错命令](#)

[Vim相关小技巧](#)

- [基本编辑](#)
 - [命令行模式->Insert模式](#)
 - [复制/剪切和粘贴](#)
 - [快速移动](#)
 - [退出](#)
- [提升速度](#)
 - [文本搜索与替换](#)
 - [窗口分屏](#)
 - [基本视觉模式](#)
 - [一些混杂的命令](#)
 - [等号对齐](#)
 - [ab命令](#)
 - [map映射](#)
 - [使用ctags和cscope](#)
- [另外的主题](#)
 - [mapleader](#)
 - [Vim加密和解密](#)
 - [设置光标高亮行颜色](#)
- [编写vim插件](#)
- [vim终极插件](#)

一个关于linux使用小技巧的文档，主要还是分享一下vim相关的tips。100在这里只是表明很多，具体的数目取决于您的参与和贡献。

快捷键

命令行快捷键

- CTRL + A 移动光标到行末
- CTRL + E 移动光标到行首
- CTRL + W 剪切一个单词
- CTRL + U 剪切光标至行首的内容
- CTRL + K 剪切光标至行末的内容
- CTRL + Y 粘贴

向终端粘贴文本

- Shift + Insert

快速搜索

- CTRL + R

用途：反向搜索执行过的命令，搜索上几个命令还是[上]键更有效率

清除屏幕

- CTRL + L

Alias

ls相关宏

- alias ll='ls -alF'
- alias la='ls -A'
- alias l='ls -CF'

cd相关宏

- alias ..='cd ..'
- alias ...='cd ../..'
- alias='cd ../../..'

自定义

- mcd() {mkdir -p "\$1"; cd "\$1";}
- cls() {cd "\$1"; ls;}

清除屏幕

- alias c="printf '\033c'"

屏蔽gdb启动时提示信息

- alias gdb='gdb -q'

Just Fun

- alias busy="cat /dev/urandom | hexdump -C | grep \"ca fe\""

Alias definitions

Put your alias into a separate file like .bash_aliases

```
if [ -f ~/.bash_aliases ]; then
    source ~/.bash_aliases
fi
```

Git alias

在git的配置文件.gitconfig中配置如下alias:

```
[alias]
    co = checkout
    ci = commit
    st = status
    br = branch -v
    l = log --pretty=format:'%h : %s' --topo-order --graph
    g = grep -n --color -E
    cp = cherry-pick -x
    nb = checkout -b
    who = shortlog -n -s --no-merges

    addall = !sh -c 'git add. && git add -u'
```

易错命令

- alias sl='ls'
- alias ptyhon='python'

Vim相关小技巧

这里分享的技巧全都是vim原生态的, 不需要另外安装插件

基本编辑

命令行模式->Insert模式

- i 光标所在字符前插入

- I 光标所在行的行首插入
- a 光标所在字符后插入
- A 光标所在行的行尾插入
- o 光标所在行的下面另起一行插入
- O 光标所在行的上面另起一行插入

复制/剪切和粘贴

- nyy 复制n行
- ndd 剪切n行
- p 粘贴在光标所在行的下面
- P 粘贴在光标所在行的上面

我们也可以使用move操作，将第5行和第10行之间内容剪切粘贴到第20行下面，操作如下：

- :5,10m20

快速移动

- hjkl vim方向键控制
- nj 向下移动n行
- nw 向右移动n个单词
- nb 向左移动n个单词

还有几个很常用的快速移动命令，如下：

- ctrl + f 向前查看分页
- ctrl + b 向后查看分页
- ngg或者nG 光标移动至第n行
- gg 光标移动到文件首部
- G 光标移动到文件尾部

退出

- :wq 保存并退出vim
- :wqa 保存并退出多屏
- ZZ 保存并退出vim
- q! 退出不保存（如果缓存有修改）

设置insert模式下的映射关系：jk ->

，这样我们可以在insert模式下执行j k shift z z保存并退出

提升速度

文本搜索与替换

vim打开文件的同时进行搜索,如需搜索file文件中的字符AAA

```
vim file +/"AAA"
```

打开文件，在命令行模式执行：

- / 向前查找字符串
- ? 向后查找字符串
- "*" + n 查找所选单词并向后查找
- "*" + N 查找所选单词并向后查找

连续行注释，g表示默认，交互询问是否执行使用c,如需注释第一行到第20行可执行

- :1,20/^#/g
- :1,20?^?//g

%s表示全文本替换

- :%s/AAA/BBB/g
- :%s/AAA/BBB/c

交互式替换的整个过程也可以如下操作：

1. gg切换到文件开头部分
2. / 向后查找AAA
3. cw替换掉之后，使用n找到下一处匹配，使用.号执行相同替换操作

窗口分屏

- :sp 水平分割，默认当前文件
- :Ve 垂直分割

同一个文件适合做水平分割；

同时操作.h和.cpp文件建议垂直切分；

对于多个分屏可使用ctrl_w + (hjkl) 做窗口间的跳转

当然你还可以：

```
vim -O file1 file2 垂直分屏
vim -o file1 file2 水平分屏
```

基本视觉模式

ctrl v 可进入视觉模式，主要用于整体块处理

比如cpp文件注释所选的行（行首插入//）可执行：

```
ctrl + v, [上][下]键选择处理的行, shift a, //, <esc>
```

一些混杂的命令

等号对齐

对齐光标所在的行

- =

对齐光标之后的n行

- `n==`

对齐整个文件

- `gg=G`

在打开\关闭文件的时候自动对齐整个文件

- `:autocmd BufWritePre,BufRead .cpp,.ch,sh,.java :normal! gg=G`

ab命令

ab命令可以将长串字符用缩写表示，用在insert模式当中

- `:ab jz jiangzheahu@126.com`

map映射

- `:map ggvG`

实现类似于Windows下的Ctrl+a全选

使用ctags和cscope

首先配置.vimrc,加入

```
set tags=tags;/
:noremap <leader>ss :cs f s <C-R>=expand("cword")<CR><CR>
```

自定义index程序放置在~/bin目录，如下：

```
ctags -R .
cscope -Rbq *
```

我们进入CPP工程目录，执行index程序后就生成了tags和cscope索引文件用于函数的跳转，使用查看函数定义，使用返回,使用,ss查看光标所在单词的全局匹配。

另外的主题

mapleader

vim 默认是\，可以设置为',，按,ev打开配置文件，按,sv重载配置文件

- `:let mapleader = ','`
- `:nnoremap ev :vsplit $MYVIMRC`
- `:nnoremap sv :so $MYVIMRC`

Vim加密和解密

- `:X` 输入密码

- :set key= 清除

设置光标高亮行颜色

这里简单的将其设置为前景红色，背景蓝色

- set cursorline
- highlight CursorLine ctermbg=Blue ctermfg=Red

编写vim插件

我们使用的vim插件都是根据vim语法编写出来的，详细内容可以查看 [《Learn Vim Script The Hardway》](#)，上边介绍的tips除了缺少智能补全功能，已经基本能满足日常工作需要。

任何知识的获取都有时间成本，而你则需要权衡是否值当，这里就不介绍vim插件的编写了。

vim终极插件

Vim插件首推Vundle和YouCompleteMe，当然Taglist也还不错。顺带提一下编辑器之神：**emacs**，和**Vim**被统称为文本编辑器中的绝代双骄：)

这里介绍vim tips的目的是使你在Linux终端情况下也能高效的处理编辑工作，如果你已经有了好用的集成开发环境。那么我强烈建议：**人生苦短，我用IDE。**