

vi 编辑器

简介

vim

vim是vi的升级版，它不仅兼容vi的所有指令，而且还有一些新的特性，例如vim可以撤消无限次、支持关键词自动

完成、可以用不同的颜色来高亮你的代码。

网址：<http://www.vim.org/download.php>

ctags

ctags是个很强大的vim插件，有了它浏览代码时可以在函数, 变量之间跳来跳去等等。

网址：<http://ctags.sourceforge.net/>

安装

安装vim

```
sudo apt-get install vim
```

安装ctags

```
sudo apt-get install ctags
```

命令行打开文件

方法	功能
vi filename	打开或新建文件，并将光标置于第一行行首
vi +n filename	打开文件，并将光标置于第n行行首
vi -b filename	以二进制方式打开文件
vi filename +":cmd"	打开文件，并执行命令cmd

示例：

方法	功能
vi -b test.c +":1,2 d"	打开文件test.c，删除第1至第2行
vi -b test.c +":1,2 d" +":wq"	打开文件test.c，删除第1至第2行，保存退出

编辑器模式

vim编辑器有2个操作模式：

模式	功能
普通模式 (Normal mode)	控制光标的移动、进行文本的选择、复制、粘贴、剪切、删除、查找等工作
插入模式 (Insert mode)	vi将输入的字符作为正文内容放在正在编辑的文件中

模式切换

普通模式切换到插入模式

方法	功能
i	从光标当前位置开始插入
I	从光标所在行的开头开始插入正文
a	从光标当前位置的下一个字符开始插入
A	从光标所在行的末尾开始插入正文
o	在光标位置的下一行插入一个空行，再进行插入
O	在光标位置的上一行插入一个空行，再进行插入

插入模式切换到普通模式

方法	功能
ESC	返回到普通模式

编辑器操作

分割窗口

方法	功能
:sp file	分割出一个横向窗口
:vsp file	分割出一个纵向窗口

窗口跳转

方法	功能
ctrl-w w	跳转至下一个窗口
ctrl-w p	跳转至前一个窗口

自动完成(插入模式)

方法	功能
ctrl+p	向前搜索可匹配的单词
ctrl+n	向后搜索可匹配的单词
ctrl+f	搜索可匹配的文件名并完成

打开、查看、保存(普通模式)

方法	功能
:w	保存当前文件
:saveas file	把文件保存为一个新的文件，文件保存后编辑新的文件
:w file	把文件保存为一个新的文件，但仍然编辑老的文件
:q	退出正在编辑的文件
:q!	不保存文件并无条件退出
:wq	保存退出当前文件
:x	保存退出当前文件
ZZ	保存退出当前文件
ZQ	不保存文件并无条件退出
:edit file	不离开vim，打开文件file
:edit	重新加载文件(edit也可以简写为e)
:edit!	重新加载当前文档，并丢弃已做的改动
:hide edit foo.txt	既不想保存，也不想放弃修改，又想编辑其他的文件
:tabe file	不离开vim，打开文件file
:tabedit file	不离开vim，打开文件file
[n]gt	改变当前正在编辑的文件
:ls	列出当前vim打开的文件

方法	功能
:files	列出当前vim打开的文件
:buffers	列出当前vim打开的文件文件；编号后面有"#"的，代表最近编辑过的文件；文件编号后面有"%a"的，代表当前编辑的文件
:b[n]	改变当前正在编辑的文件（n代表文件编号或文件名）
:bn	改变当前正在编辑的文件
:bp	改变当前正在编辑的文件
:bd [n]	关闭打开的文件

文件比较工具

方法	功能
vimdiff file1 file2	比较两个文件
:set diff	比较多个文件，用 :vsp filename或:sp filename命令打开另一个文件, 然后在每个文件窗口中输入此命令
:diffsplit file	水平分割窗口比较文件
:vertical diffsplit file	竖直分割窗口比较文件
[c	跳到上一个差异点
]c	跳转到下一个差异点
dp	把一个差异点中当前文件的内容复制到另一个文件里 (diff "put")

文件跳转和Tags

方法	功能
:tags	显示所有匹配的标签
:tag [关键字]	跳转到与“关键字”匹配的标记处，有多个匹配时跳转至第一个匹配
Ctrl-]	跳转到与光标下的关键字匹配的标记处；除“关键字”直接从光标位置自动获得外，功能与“:tag”相同
Ctrl-T	跳转回上次使用以上命令跳转前的位置
:tselect [关键字]	显示与“关键字”匹配的标记列表，输入数字跳转到指定的标记（缩写：“:ts”）
g]	与“Ctrl-]”功能类似，但使用的命令是“:tselect”
:tjump [关键字]	类似于“:tselect”，但当匹配项只有一个时直接跳转至标记处而不再显示列表
g Ctrl-]	与“Ctrl-]”功能类似，但使用的命令是“:tjump”
:tnext	跳转到下一个匹配的标记处（缩写：“:tn”）
:tprevious	跳转到上一个匹配的标记处（缩写：“:tp”）
:tfirst	第一个标签匹配处
:tlast	最后一个标签匹配处
gd	跳到变量的定义处（对局部变量有效）
[[跳到函数定义开头 或者上一个函数定义开头
]]	跳到函数定义结尾， 或者下一个函数定义结尾
]]	跳到下一个函数开头
%	跳到 和() {}等匹配的对应位置
{ 或 }	跳到下/上一个空白行
`.	跳转到最近修改过的位置

方法	功能
`"	跳转到离开文件时光标所在的位置
ctrl+o	在历史浏览的各个文件中跳转
ctrl+i	在历史浏览的各个文件中跳转
ctrl+i/o	无论你是用tags的ctrl+]跳走了,还是通过gf跳走了,都可以在不同的文件之间跳转

移动光标(普通模式)

方法	功能
gg	将光标定位到文件开始处
G	将光标定位到文件结束处
[n]G	将光标定位到第n行开始处
H	光标定位到屏幕顶部
M	光标定位到屏幕中间
L	光标定位到屏幕底部
[n]+	光标移下移n行
[n]-	光标移上移n行
0	光标移至行首
^	光标移至行首字符处
\$	光标移至行尾
h或Backspace	光标向左移动一格
j或Ctrl + n或Enter	光标向下移动一格
k或Ctrl + p	光标向上移动一格
l或Space	光标向右移动一格
w或W	光标右移一个字至字首
b或B	光标左移一个字至字首
e或E	光标右移一个字至字尾

屏幕翻滚(普通模式)

方法	功能
Ctrl + e	屏幕上翻一行
Ctrl + y	屏幕下翻一行
Ctrl + u	屏幕上翻半屏
Ctrl + d	屏幕下翻半屏
Ctrl + b	屏幕上翻一屏
Ctrl + f	屏幕下翻一屏

选择(普通模式)

方法	功能
v	配合其他命令、方向键、鼠标选择内容
	鼠标左击 + 移动选择内容

复制(普通模式)

方法	功能
y	拷贝选择的内容到剪贴板
[n]yy	复制从当前行开始的n行到粘贴板

粘贴(普通模式)

方法	功能
p	把粘贴板上的内容插入到当前行

剪切(普通模式)

方法	功能
d	剪切选择的内容到剪贴板
D	剪切从光标至行末的内容
c	剪切选择的内容到剪贴板并且进入插入模式
[n]dd	剪切从当前行开始的n行到粘贴板

删除(普通模式)

方法	功能
[n]x	删除光标后n个字符
[n]X	删除光标前n个字符

查找(普通模式)

方法	功能
/string	从光标处向文件尾查找字符串
?string	从光标处向文件首查找字符串
n	同方向重复上一次查找命令
N	反方向重复上一次查找命令
:nohl	取消高亮
:set ignorecase	让VIM忽略大小写
:set noignorecase	不让VIM忽略大小写

替换(普通模式)

方法	功能
r	替换当前字符
R	替换当前及后续字符直至按下Esc键
cc	替换光标所在的行
: [range]substitute/from/to/[option]	在[range]指定范围，将“from”替换成“to”;substitute可简写为s
:s/模式/字符串/标志	进行替换，其中的“模式”是一个正则表达式

替换示例

方法	功能
:n1,n2s/p1/p2/g	将n1到n2行中所有p1均用p2替代
:s/p1/p2/g	将当前行中所有p1均用p2替代
:%s/p1/p2/g	将文件中所有p1均用p2替代
:g/p1/s//p2/g	将文件中所有p1均用p2替代
:%s/^s*//	删除行首空格
:%s\s*\$//	删除行尾空格
:g/^s*\$//d	删除空行
:%s/^M\$//g	去掉行尾的^M(这里的“^M”要使用“CTRL-V+CTRL-M”生成)
:%s/^M//g	去掉所有的^M
:%s/\r/[ctrl-v]+[enter]/g	将^M替换成回车
:%s/\r/\r/g	将^M替换成回车

附：正则表达

匹配符	功能
g	(global)全局的
%	指当前所编辑的文件
^	[反向选择] 或 [定位在行首],在[]内代表反向选择,在[]外代表定位在行首
[ABC]	匹配 [...] 中的所有字符
[^ABC]	匹配除了 [...] 中字符的所有字符
\$	行末，要匹配 \$ 字符本身，请使用 \\$
\w	匹配字母、数字、下划线。等价于 [A-Za-z0-9_]
\f	匹配一个换页符。值：0x0c，输入方法：CTRL-V CTRL-L
\n	匹配一个换行符。值：0x0a，输入方法：CTRL-V CTRL-J
\r	匹配一个回车符。值：0x0d，输入方法：CTRL-V CTRL-M
\t	匹配一个制表符。值：0x09，输入方法：CTRL-V CTRL-I
\v	匹配一个垂直制表符。值：0x0b，输入方法：CTRL-V CTRL-K
\s	匹配任何空白字符，包括空格、制表符、换页符等等。等价于 [\f\n\r\t\v]
\S	匹配任何非空白字符
.	匹配除换行符 \n 之外的任何单字符。要匹配 .，请使用 \.

匹配符	功能
*	匹配前面的子表达式零次或多次。要匹配 * 字符，请使用 *
+	匹配前面的子表达式一次或多次。要匹配 + 字符，请使用 \+

多行剪切、复制(普通模式)

方法	功能
:n1,n2 d	剪切n1到n2行之间的内容到粘贴板
:n1,n2 m n3	剪切n1行到n2行之间的内容到n3行下
:n1,n2 co n3	复制n1行到n2行之间的内容到n3行下

撤销(普通模式)

方法	功能
u	撤销前面多次修改
Ctrl + r	反撤销前面多次修改

排版(普通模式)

方法	功能
[n]>	所选内容向右移动n个tab
[n]<	所选内容向左移动n个tab
=	格式化代码(将凌乱不堪的代码对齐): 选中要排版的内容后按"="号键

大小写切换(普通模式)

方法	功能
U	选中的内容切换成大写
u	选中的内容切换成小写
~	选中的内容大写小写互换

编码转换工具

方法	功能
:set fileencoding=utf8	设置文件编码为utf8（缩写：":set fenc"）
:set fileencoding=gb2312	设置文件编码为gb2312
:set fileencodings= [编码列表]	解码的猜测列表， 例如:utf-8,usc-bom,gb2312,gb18030,gbk,euc-jp,cp936
:set fencs	:set fileencodings的缩写
iconv -f gbk -t utf8 file1 -o file2	将gbk编码的文件重新编码为utf-8， -o 指定新的文件名

杂项(普通模式)

方法	功能
.	执行上一次操作
:set compatible?	查询是否设置了兼容模式
:set filetype?	查询vim识别的当前的文件类型
:set number	显示行号（或set nu）
:set nonumber	不显示行号（或set nonu）
:set expandtab	用space替代tab的输入
:set noexpandtab	不用space替代tab的输入
:!command	执行shell命令commandxxd
:%!xxd	二进制显示和处理(打开文件时需要加上+b)
:%!xxd -r	还原二进制显示和处理

VIM保存临时数据

- OS级别-剪切板
- VIM级别-寄存器

VIM寄存器

寄存器是VI用于保存临时数据的地方，不同于传统的编辑器（于系统共享一个寄存器，也成为剪切板），VIM具有多个寄存器，分别保存不同的临时数据，活用多个寄存器可以显著提高数据的安全性和可操作性。

同时，为了与系统剪切板互通，VIM有一个专用的寄存器，与系统剪切板内容一致，既保证了VIM本身的统一性，也实现了与操作系统的对接。

查看寄存器值

- 查看所有寄存器值: `:reg`
- 查看指定寄存器值: `:reg "{register_name}`

调取寄存器值

- NORMAL Mode: `"{register_name}`
- COMMAND MODE: `<C-r>+"寄存器名称` (输入<CTL+r>后VIM会自动打出"寄存器引用符号")。
- INSERT MODE: `<C-r>+寄存器名称` (无需输入寄存器引用符号")

xterm_clipboard

要完成vim中的内容复制到系统剪切板, 需要vim支持

`+clipboard`, 检查的方法:

```
→ ~ vim --version | grep clipboard
-clipboard          +keymap          +printer
    +vertspllit
+emacs_tags         +mouse_gpm       -sun_workshop
    -xterm_clipboard
```

查看clipboard和xterm_clipboard前面的符号

- 加号 (+), 表示支持
- 减号 (-), 表示不支持

可以看到现在是不支持从vim中复制到系统剪切板中的。

怎么可以让它支持呢, 就需要安装一些插件来解决一下, 其实也很简单就是下面的这一条命令

```
sudo apt install -y vim-gtk
```

安装完成以后，同样的在终端中输入 `vim --version | grep clipboard`

```
→ ~ vim --version | grep clipboard

+clipboard          +keymap             +printer
  +vertsplit
+emacs_tags         +mouse_gpm          -sun_workshop
  +xterm_clipboard
```

共享系统剪切板和VIM寄存器

"将无名寄存器映射都系统剪切板，将复制、删除的内容保存到剪切板寄存器

`set clipboard=unnamedplus`

文档信息

- 版本:v2.0
- 时间:2020-11-01
- 作者:孔令春