vim 常用快捷键

1、vim ~/.vimrc 进入配置文件

如果不知道vimrc文件在哪，可使用 :scriptnames 来查看

set nu #行号

set tabstop=4 #一个tab为4个空格长度

set ai #设置自动缩进

syntax on #高亮

2、基本

空格键 向右移动一格

x 删除后面的字符 X 删除前一个字符 删除3个字符就是3x

dd:删除一行 D 删除到行尾

caw:改写单词 c 相当于 d 变为编辑模式

J:删除换行符，使下一行并上来。 nJ:连接后面的n行

u:撤销上一次操作 U:撤销当前行的所有修改

ctrl+r:对撤消的撤消

i 在光标前插入

I 在行首插入

a 在光标后插入

A 在行末插入

o:在当前行的下面另起一行，并变为插入模式

O：在当前行上面另起一行，变为插入模式

Ctrl+o:临时变成命令模式（一次而已）

：e!:放弃更改，然后相当于重新打开

：help:帮助，可用ZZ退出帮助窗口

vim中Nyy可以复制光标后的N行。有时我们不容易得出行数，这时可以用做标记的方法来制定复制范围：

1. 在开始行上输入ma作一个标记a
2. 移动到结束行，输入y'a会复制当前行到标记a之间的文本。d'a会删除。

或者是v进入可视模式，再13G跳转到相应行，y即可。

:10,20y　回车即可，相应的删除也是如此　:10,20d　（此方法比上面两种方法更简单）

:10,20 m 30　把10行到20行的内容，剪切到30行之后

:10,20 co 30　把10行到20行的内容，复制到30行之后

将光标放在 { 处，然后输入v%就可以把大括号中内容选定

如果光标放在第一个s上，想删除到“(”为止，则输入dt(就可以了，t(的作用是跳到下一个"("前。

ctrl +n 自动补全 ctrl + p 也一样

:ab hw hello world 用一个缩写字符串代替一个长的字符串，此处用 hw 代替 hello world

%: 移动到与制匹配的括号上去（ ），{}，[]，<>等

' 移动到上一次的修改行

fx 向右跳到本行字符x处（x可以是任何字符）
Fx 向左跳到本行字符x处（x可以是任何字符）
tx 和fx相同，区别是跳到字符x前
Tx 和Fx相同，区别是跳到字符x后

到与当前单词相同的上一个单词上， * 到与当前单词相同的下一个单词上

如果你要重复键入一个短语或一个句子, 也有一种快捷的方法。Vim有一种记录宏的机制。你键入"qa"开始把一段宏记录入寄存器变量`a'中。
按下来你可以象平常一样键入你要的操作, 只是这些操作都会被Vim记录进它命名为`a'的宏中, 再次再下"q"键, 就结束了宏`a'的录制。当你要重复执行你刚才记录的那些操作时只要使用"@a"命令。共有26个可用的寄存器供你记录宏。 使用宏你可以重复多个不同的操作。而不仅仅是插入文本了。如果你要进行某种重复的操作, 记着要用这一招呀。

:abbr hte the

:abbr hw Hello World

输入先面的单词时，自动用后面的替换。

3、移动：

b、3b、w、3w:向前\后移动几个单词，标点也算一个单词。相应的大写状态为不含标点，即只把空格和换行符作为单词间隔符。

$：移动到行尾 3$：移动到3行后的行尾

^:移动到行首，0也是

+：移到下一行的行首

-： 移到上一行的行首

f:搜索命令，小写时向后搜索（用来定位）如 fx：定位到下一个x上。 Fx：定位到上一个x上，重复时，可用;或, 不过，表示反方向

%：跳到相对应的括号上，编程时常用

33G：跳转到33行 此时按``可以返回到原来行

gg:文件头 G： 文件尾

30%：跳转到文件的30%处

"H"意为Home, "M"为Middle, "L"为Last. 当前屏幕的上中下位置，大小写皆可

Ctrl+G:显示当前位置

set number:设置显示行号，set nonumber:关闭显示

:set ruler 设置在窗口右下角显示行号，与上面的好处是，节省空间

ctrl+u\d 向上\下滚动半屏

ctrl+e\y 向上\下滚动一行

ctrl+b\f 向上\下滚动一屏 这个比较实用，记住。

zz:将当前行滚动于屏幕中间，方便查看上下文 zt置顶，zb置尾

/string 查找string，回车后，按n键可以跳到下一个，N上一个，另外按/键后，按上下键可以找到以前查找的记录，同样的 ：也有记录

?/string 同上，默认向上查找

:set ignorecase 大小写无关

:set noignorecase 大小写敏感

* ：查找下一个光标所在单词 #是查找上一个

:set hlsearch 高亮显示查找结果

:set nohlsearch 取消高亮

:nohlsearch 去掉当前显示的高完（一次性）

`` 上次光标停靠的行

% 匹配到相应括号处

>> 向右移动本行一段距离 << 向左移动本行一段距离 3<< 把下面3行（包括本行），向左移动一段距离 :20,30>> 把20行到30行向右移动一段距离

4、小幅改动：

:%s/str1/str2/g 替换每一行的 str1为 str2

:10,20s/str1/str2/g 替换从行10到行20之间的 str1为 str2

:10,$s/str1/str2/g 替换从行10到最后一行之间的 str1为 str2

:s/str1/str2/g 替换当前行的 str1为 str2

. 重复执行命令

:10,$ w test2.cpp 取行10到最后一行内容，保存到test2.cpp

:r class/User.hpp 读取文件中的内容，插入到当前行的后面

dw:删除一个单词（光标后部分） 不如：daw实用 d4w：删除4个单词 d$:删除当前光标到行尾 d^：删除当前光标至行首 d换成c效果是一样的，只是操作完会变成insert模式 dnw： 删除N个单词 dnj: 向下删除n行 dnk: 向上删除n行

X：删除左边的字符，相当于<-键，x删除当前字符 D:相当于d$ C:相当于c$ s:相当于c1 S:相当于cc

r:替换当前字符，但不会进入insert模式 3r:把后面3个字符替换掉 R：替换模式

.:重复上一次操作

v：进入Visual模式 V:进入可视行模式，比如 Vjjd 删除3行 Ctrl+v：可视块模式

P：粘贴至光标前 p：粘贴至光标后 3P：粘贴3次 "2p 粘贴最后第二次的删除的内容

yy:复制一行 yaw： 复制一个单词,光标在单词任意位置 ynw： 复制N个单词 ynj: 向下复制n行 ynk: 向上复制n行

自动缩进：

:set cindent(所有的set都可以简写为se,虽然只节省了一个字符, 译者注) 需要注意的是cindent控制缩进量是通过shiftwidth选项的值, 而不是通过tabstop 的值, shiftwidth的默认值是8(也就是说, 一个缩进为8个空格, 译者注), 要改变默认的设置, 可以使用":set shiftwidth=x"命令, 其中x是你希望一个缩进量代表的空格的数目.

{ = 到前一个空行上
} = 到下一个空行上

5、VIM的一些插件：

c.vim :如果是用root账号的，把文件复制到/usr/share/vim/vim70中解压没有用的，不存在 $HOME/.vim 这个目录，没办法，只能新建个目录，然后把压缩包cp到这个目录，再unzip即可。在 ~/.vimrc 中 写入 filetype plugin on

:e! 返回上次保存后的状态

Ctrl+z:暂停vi,回到Unix提示符，再输入fg即可回到vi。

^回到行首，光标位于行首字母处；0回到行首，光标位于行首字母前。$回到行尾，光标位于行尾字母处。

nb:向前移动n个单词，nw：向后移动n个单词。光标位于单词的第一个字母处。 nw这个操作很慢，不知何故。 以空格、标点符号与单词的分界为分隔符。（几个连续的标点视为一个单词）

同样的，也可以使用nB,nW，只是这里只使用空格做为分隔符。

相换两个相邻字母的位置：x、p

s:删除一个字符，并进入编辑模式。 S：删除一整行，进入编辑模式，相当于cc。 ns:删除后面n个字符，并进入编辑模式。

~:更改字母的大小写，同时光标进入到下一个字符。 n~:把后面n个字母的大小写状态改变。

dw:删除单词后面部分　db：删除单词前面部分。 如果要删除整个单词(光标位于单词中间的话)，可以dbw\wdb

de:类似于dw,删除单词后面的部分（只删除到本单词结尾，dw会删除掉单词后面的空格） dE:删除的范围包括标号在内的单词结尾。

e:相当于w，向后移动一个单词。不同的是，w移动到单词第一个字符上，e移动到单词最后一个字符上。所以ea,可以给本单词追加内容。

D：d$ 的简写，同样的，C：c$的简写。

U：会恢复一整行原先的面貌，即最原始的样子。

Y：相当于yy，不同于D与C的操作方法。

.：重复上一个命令。

除了O/o，插入命令(A,a,I,i)接受数值参数，如：5Ihello，然后按ESE键。会在行首输入5个连接的hello

nr:替换后面n个字符。

nJ:合并下面的n行（从本行算起）。

ynl:向后复制n个字符。

e/E:到单词的结尾。

表2-1：编辑命令

文本对象	更改	删除	复制
一个单词	cw	dw	yw
两个单词，不包括标点符号	2cW或c2W	2dW或d2W	2yW或y2W
后退三个单词	3cb或c3b	3db 或 d3b	3yb或y3b
一整行	cc	dd	yy或Y
到一行的结尾	c$或C	d$或D	y$
到一行的开头	c0	d0	y0
单个字符	r	x或X	yl或yh
五个字符	5s	5x	5yl

滚动整屏：

^f：向前（下）一整屏

^b：向后（上）一整屏

^d：向前（下）一半屏

^u：向后（上）一半屏

z,Enter：将光标所在行移动到屏幕顶部（同于zz)

z.：将光标所在行移动到屏幕中间

z-：将光标所在行移动到屏幕尾部

nz,Enter：将第n行移动到屏幕顶部，同样的，z. 与 z- 前也可以加数字。

在屏幕中移动：

H、M、L分别移动到屏幕的顶部、中间和尾部。

nH、nL 移动到距离屏幕顶部和顶部n行的位置。

Enter：到下一行的第一个字符。

+：到下一行的第一个字符。

-：到上一行的第一个字符。

n|：移动到当前行的第n列

e：移到单词的结尾

E：移到单词的结尾（忽略标点符号）

( ) { } [[ ]] 这几个对编程作用不大，可忽略。

d/it：向后删除到it之前的位置（不删it）。 d?it：向前删除到it之前的位置（删除it）。

fx:本行中向右搜索x，光标置于x上。 Fx:向左搜索。; 重复上一个搜索命令，方向相同。 ,重复上一个搜索命令，方向相反。

tx:同fx，只是光标置于x之前。 Tx类似。

dfi：向右删除第一个i的位置（包括i) dti：同dfi，只是不包括i

Ctrl+G 查看当前行信息

nG跳转后，可使用``回到上一次的位置，''功能一样，不过只是回到前次位置所在行的开头，而不是确定的位置上。

删除包含keyword字符串的行： :g/keyword/d

删除空行：:%s/^\n$//g
