

C = Control
M = Meta = Alt | Esc
Del = Backspace

基本快捷键(Basic)

C-x C-f “find”文件, 即在缓冲区打开/
新建一个文件
C-x C-s 保存文件
C-x C-w 使用其他文件名另存为文件
C-x C-v 关闭当前缓冲区文件并打开
新文件

C-x i 在当前光标处插入文件

C-x b 新建/切换缓冲区

C-x C-b 显示缓冲区列表

C-x k 关闭当前缓冲区

C-z 挂起 emacs

C-x C-c 关闭 emacs

光标移动基本快捷键(Basic Movement)

C-f 后一个字符

C-b 前一个字符

C-p 上一行

C-n 下一行

M-f 后一个单词

M-b 前一个单词

C-a 行首

C-e 行尾

C-v 向下翻一页

M-v 向上翻一页

M-< 到文件开头

M-> 到文件末尾

编辑(Editint)

M-n 重复执行后一个命令 n 次

C-u 重复执行后一个命令 4 次

C-u n 重复执行后一个命令 n 次

C-d 删除(delete)后一个字符

M-d 删除后一个单词

Del 删除前一个字符

M-Del 删除前一个单词

C-k 移除(kill)一行

C-Space 设置开始标记 (例如标记区域)

C-@ 功能同上, 用于 C-Space 被操作系统拦截的情况

C-w 移除(kill)标记区域的内容

M-w 复制标记区域的内容

C-y 召回(yank)复制/移除的区域/行

M-y 召回更早的内容 (在 kill 缓冲区

内循环)

C-x C-x 交换光标和标记

C-t 交换两个字符的位置

M-t 交换两个单词的位置

C-x C-t 交换两行的位置

M-u 使从光标位置到单词结尾处的字母变成大写

M-l 与 M-u 相反

M-c 使从光标位置开始的单词的首字母变为大写

重要快捷键(Important)

C-g 停止当前运行/输入的命令

C-x u 撤销前一个命令

M-x revert-buffer RETURN (照着这个输入)撤销上次存盘后所有改动

M-x recover-file RETURN 从自动存盘文件恢复

M-x recover-session RETURN 如果你编辑了几个文件, 用这个恢复

在线帮助(Online-Help)

C-h c 显示快捷键绑定的命令

C-h k 显示快捷键绑定的命令和它的作用

C-h l 显示最后 100 个键入的内容

C-h w 显示命令被绑定到哪些快捷键上

C-h f 显示函数的功能

C-h v 显示变量的含义和值

C-h b 显示当前缓冲区所有可用的快捷键

C-h t 打开 emacs 教程

C-h i 打开 info 阅读器

C-h C-f 显示 emacs FAQ

C-h p 显示本机 Elisp 包的信息

搜索/替换(Search/Replace)

C-s 向后搜索

C-r 向前搜索

C-g 回到搜索开始前的位置(如果你仍然在搜索模式中)

M-% 询问并替换(query replace)

Space 或 y 替换当前匹配

Del 或 n 不要替换当前匹配

. 仅仅替换当前匹配并退出(替换)

, 替换并暂停(按 Space 或 y 继续)

! 替换以下所有匹配

^ 回到上一个匹配位置

RETURN 或 q 退出替换

使用正则表达式(Regular expression)搜索/替换

可在正则表达式中使用的符号:

^ 行首

\$ 行尾

. 单个字符

* 任意多个(包括没有)字符

\< 单词开头

\> 单词结尾

[] 括号中的任意一个字符(例如[a-z]表示所有的小写字母)

M C-s RETURN 使用正则表达式向后搜索

M C-r RETURN 使用正则表达式向前搜索

C-s 增量搜索

C-s 重复增量搜索

C-r 向前增量搜索

C-r 重复向前增量搜索

M-x query-replace-regexp 使用正则表达式搜索并替换

窗口命令(Window Commands)

C-x 2 水平分割窗格

C-x 3 垂直分割窗格

C-x o 切换至其他窗格

C-x 0 关闭窗口

C-x 1 关闭除了光标所在窗格外所有窗格

C-x ^ 扩大窗格

M-x shrink-window 缩小窗格

M C-v 滚动其他窗格内容

C-x 4 f 在其他窗格中打开文件

C-x 4 0 关闭当前缓冲区和窗格

C-x 5 2 新建窗口(frame)

C-x 5 f 在新窗口中打开文件

C-x 5 o 切换至其他窗口

C-x 5 0 关闭当前窗口

书签命令(Bookmark commands)

C-x r m 在光标当前位置创建书签

C-x r b 转到书签

M-x bookmark-rename 重命名书签

M-x bookmark-delete 删除书签

M-x bookmark-save 保存书签

C-x r l 列出书签清单

d 标记等待删除

Del 取消删除标记

x 删除被标记的书签

r 重命名

s 保存列表内所有书签

f 转到当前书签指向的位置

m 标记在多窗口中打开

v 显示被标记的书签(或者光标当前位置的书签)

t 切换是否显示路径列表

w 显示当前文件路径

q 退出书签列表

M-x bookmark-write 将所有书签导出至指定文件

M-x bookmark-load 从指定文件导入书签

Shell

M-x shell 打开 shell 模式

C-c C-c 类似 unix 里的 C-c(停止正在运行的程序)

C-d 删除光标后一个字符

C-c C-d 发送 EOF

C-c C-z 挂起程序(unix 下的 C-z)

M-p 显示前一条命令

M-n 显示后一条命令

DIRectory EDitor (dired)

C-x d 打开 dired

C(大写 C) 复制

d 标记等待删除

D 立即删除

e 或 f 打开文件或目录

g 刷新当前目录

G 改变文件所属组(chgrp)

k 从屏幕上的列表里删除一行(不是真的删除)

m 用*标记

n 光标移动到下一行

o 在另一个窗格打开文件并移动光标

C-o 在另一个窗格打开文件但不移动光标

P 打印文件

q 退出 dired

Q 在标记的文件中替换

R 重命名文件

u 移除标记

v 显示文件内容

x 删除有 D 标记的文件

Z 压缩/解压缩文件

M-Del 移除标记(默认为所有类型的标记)

~ 标记备份文件(文件名有~的文件)等待删除

标记自动保存文件(文件名形如

#name#)等待删除

* / 用*标记所有文件夹(用 C-u */n 移除标记)

= 将当前文件和标记文件(使用 C-@ 标记而不是 dired 的 m 标记)比较

M-= 将当前文件和它的备份比较

! 对当前文件应用 shell 命令

M-} 移动光标至下一个用*或 D 标记的文件

M-{ 移动光标至上一个用*或 D 标记的文件

% d 使用正则表达式标记文件等待删除

% m 使用正则表达式标记文件为*

+ 新建文件夹

> 移动光标至后一个文件夹

< 移动光标至前一个文件夹

s 切换排序模式(按文件名/日期)

或许把这个命令归入这一类也很合适:

M-x speedbar 打开一个独立的目录显示窗口

Telnet

M-x telnet 打开 telnet 模式

C-d 删除后一个字符或发送 EOF

C-c C-c 停止正在运行的程序(和 unix 下的 C-c 类似)

C-c C-d 发送 EOF

C-c C-o 清除最后一个命令的输出

C-c C-z 挂起正在运行的命令

C-c C-u 移除前一行

M-p 显示前一条命令

Text

只能在 text 模式里使用

M-s 使当前行居中

M-S 使当前段落居中

M-x center-region 使被选中的区域居中

宏命令(Macro-commands)

C-x (开始定义宏

C-x) 结束定义宏

C-x e 运行最近定义的宏

M-n C-x e 运行最近定义的宏 n 次

M-x name-last-kbd-macro 给最近定义的宏命名(用来保存)

M-x insert-kbd-macro 将已命名的宏保存到文件

M-x load-file 载入宏

编程(Programming)

M C-\ 自动缩进光标和标记间的区域

M-m 移动光标到行首第一个(非空格)字符

M-^ 将当前行接到上一行末尾处

M-; 添加缩进并格式化的注释

C, C++ 和 Java 模式

M-a 移动光标到声明的开始处

M-e 移动光标到声明的结尾处

M C-a 移动光标到函数的开始处

M C-e 移动光标到函数的结尾处

C-c RETURN 将光标移动到函数的开始处并标记到结尾处

C-c C-q 根据缩进风格缩进整个函数

C-c C-a 切换自动换行功能

C-c C-d 一次性删除光标后的一串空格(greedy delete)

为了实现下面的一些技术, 你需要在保存源代码的目录里运行“etags *.c *.h *.cpp”(或者源代码的其他的扩展名)

M-.(点) 搜索标签

M-x tags-search ENTER 在所有标签里搜索(使用正则表达式)

M-,(逗号) 在 tags-search 里跳至下一个匹配处

M-x tags-query-replace 在设置过标签的所有文件里替换文本

宏命令

:qx	开始记录宏, 存入寄存器x
q	退出记录模式
[n]@x	重复n次寄存器x中的宏命令

标签命令 minibuffer

:tabe fn	在一新标签中编辑文件 fn
gt与gT	切换到下/上一个标签页
:tabr	切换到第一个标签页
:tabl	切换到最后一个标签页
:tabm[N]	当前 tab 移到第 N 个 tab 后

窗口命令

ctrl+w s	水平分割窗口
ctrl+w w	切换窗口
ctrl+w q	退出当前窗口
ctrl+w v	垂直分割窗口

缓冲区, 待写入(:w)物理文件

:e file	把 file 加载到新的缓冲区中
:bn	跳转到下一个缓冲区
:bd	删除缓冲区(关闭文件)
:sp fn	分割窗口, 并加载 fn 到新窗口

退出编辑器

:w	保存修改
:wq与:x与ZZ	保存修改并退出
:q	退出, 若有修改, 会提示
:q!	强制退出, 放弃修改

查找替换

/pattern	向后搜索字符串pattern
?pattern	向前搜索字符串pattern
n	/下一个匹配?上一个匹配
N	/上一个匹配?下一个匹配
:%s/old/new/g	将所有old换为new
:%s/old/new/gc	old换new, 有提示

复制粘贴

dd	删除光标所在行
dw	删除一个字(word)
x	删除当前字符

X	删除前一个字符
D	删除到行末
[n]yy	复制一行, 加数字 n, 复制 n 行
yw	复制一个字
y\$	复制到行末
p	粘贴粘贴板内容到当前行下面
P	粘贴粘贴板内容到当前行上面
lp	有缩进的粘贴, vim会自动缩进
"a	将内容放入/存入 a 寄存器

移动光标

h,j,k,l	左, 下, 上, 右
ctrl+f与ctrl+b	上翻一页/下翻一页
ctrl+u与ctrl+d	上滚半屏/下滚半屏
%	跳到与当前括号匹配的括号处, 如当前在{, 则跳转到与之匹配的}处
w	跳到下一个字首, 按标点或单词分割
W	跳到下一个字首, 长跳, 如end-of-line 被认为是一个字
ctrl+e与ctrl+y	下卷一行/上卷一行

e与E	跳到下一个字尾, 短/长跳
b与B	跳到上一个字, 短/长跳
0	跳至行首, 不管有无缩进, 就是跳到第0个字符
^	跳至行首的第一个字符
\$	跳至行尾
gg	跳至文件的第一行
gd	跳至当前光标所在的变量的声明处
[N]G	跳到第N行, 如0G, 就等价于gg, 100G 就是第100行
fx	在当前行中找x字符, 找到了就跳转至
;与,	重复/逆重复上一个 f/F 命令, 而不用重复的输入 fx/Fx
tx 与 Tx	与 fx/Fx 类似, 但是只是跳转到 x 的前/后一个字符处
Fx	跟fx的方向相反

),(跳转到上/下一个语句
 	转到上一个位于第一列的“{”
 	转到下一个位于第一列的“{”
{	转到上一个空行
}	转到下一个空行
*	查找光标所在处的单词, 向下查找
#	查找光标所在处的单词, 向上查找
`.	跳转到上次编辑位置

在屏幕上移动

H	移动光标到当前屏幕上的最上边
M	移动光标到当前屏幕上的中间
L	移动光标到当前屏幕上的最下边

书签

ma	把当前位置存成标签a
`a	跳转到标签 a 处

编辑

:X	对所编辑的文件进行简单加密
[n]i=ESC与[n]a=ESC	重复输入n个=
ctrl+e与ctrl+y	逐字克隆下/上行内容
ctrl+n与ctrl+p	自动下/上查补齐
ctrl+r %	编辑状态下, 插入文件名
!asort	把标记a到光标处的文件排列
J	将下一行和当前行连接为一行
r与R	替换一个/连续替换
cc	删除当前行并进入编辑模式
cw	删除当前字, 并进入编辑模式
c\$	擦除从当前位置至行末的内容, 并进入编辑模式
s	删除当前字符并进入编辑模式
S	删除光标所在行进入编辑模式
xp	交换当前字符和下一个字符
ddp	交换当前行和下一行
u	撤销
ctrl+r	重做
.	重复上一个编辑命令
~	切换大小写, 当前字符
g~iw	切换当前字的大小写

gUiw	将当前字变成大写
guiw	将当前字变成小写
==	自动缩进当前行
[x]>>与[x]<<	将x行右/左移一个单位
zr,zm,zR,zM	打开关闭, 多层所有折叠
zn 与 zN,zi	禁止/恢复, 重禁/复折行
zf,zo,zc,zd	创建, 打开, 关闭, 删除折叠

插入模式

i与a	光标所在处前/后进入插入模式
I与A	光标所在行首/尾进入插入模式
o与O	光标所在行下/上插入一空行
Esc	退出插入模式

可视模式-标记文本

v与V	可视模式, 单字符/行模式
ctrl+v	可视模式, 列模式
o与O	跳转光标到选块对角/回行端
U	将选中块中的内容转成大写
aw	选中一个字
ab	选中括号中的内容, 包括括号
aB	选中{}括号中的所有内容
ib	选中括号中的内容, 不含括号
iB	选中{}中的内容, 不含{}

对标记进行动作

>与<	块右移, 块左移
y与d	复制块, 删除块
~	切换块中内容的大小写

寄存器

:reg	查看寄存器, :reg a 只查看a
"ayy	复制进寄存器a (52个)中
"0~"9	1~9删除寄存器, 0复制寄存器
"- "_	行内删除寄存器, 黑洞寄存器
"= "/"	表达式寄存器, 模式寄存器
". "%	最近插入文本, 当前文件名
": "#	最近命令, 当前交替文件名
"* "+ "~	选择及拖拽(剪切板开启)
:let @a=""	清除寄存器a中数据