Vim 之奇技淫巧

@黄庚根

2017.09

长安汽车
CHANGAN

议程

- 介绍
- Vim 入门篇
- Vim 进阶篇
- Vim高级篇
- 常用命令

Vim 适合哪些人？

Linux常客

程序猿

攻城狮

技术狂

vim感兴趣的人

为什么要使用Vim?

操作简洁

高效

跨平台

功能强大

Vi 、Vim有什么关系？

Vi：记事本

Vim：IDE

vim = Vi IMproved

需要准备什么？

学习曲线陡峭

需要时间

不断练习和磨练

入门篇

- 工作模式
  - Normal模式 ESC
  - 插入模式 i
  - 命令模式 :
  - 可视化模式 v, V, Ctrl + v
- 基本命令
  - 移动光标：

还在玩方向键？？

```
      ^
      k                Hint:  h 键在左边，左移.
< h       l >                 l 键在右边，右移.
      j                       j 键看上去像一个向下的箭头，下移.
      v
```

入门篇

- 基本命令

  编辑： i

  撤销与重做：u, Ctrl+R

  保存与退出：:w, :q, :x, ZZ

Vim 进阶篇

- 快速移动光标
  - 以下操作在Normal模式下执行：
  - 行内移动光标：
    - ✓ ^, $, 0
    - ✓ w, W
    - ✓ b, B
    - ✓ e, E
    - ✓ f, F
    - ✓ 2f{X}, {n}F{X}：
  - 行间移动光标:
    - ✓ {n}G, G
    - ✓ gg → =1G

Vim 进阶篇

- 快速移动光标
  - 按匹配括号移动光标，只支持() [] {}:
    %
  - 屏内移动光标:
    H, M, L
  - 按句移动光标：
    (,)
    {, }
  - 按代码块移动光标，要求 { 或 } 独占一行
    [[,]]

Vim 进阶篇

- 快速移动光标
  - 移动光标至上次停留的位置：
    '' 两个单引号
  - 移动光标至上次 上一次的修改行:
    '.
  - 移动光标至上次 上一次的修改点：
    `.

Vim 进阶篇

- 如何进入编辑模式

  I，i
  A，a
  X，x
  D，d → D=(d$)
  C，c → C=(c$)
  R, r

- 快速删除

  dd, dw, de, d{n}w, {n}dd, dgg, d{n}gg, dG, df{char}, dt{char}

  cd, cw, ce, c{n}w, {n}cc, cgg, c{n}gg, cG , cf{char}, ct{char}

Vim 进阶篇

- 重复操作

  .

  {n}.

- 大小写转换

  ~ 或 gU, gu

- 更快

  <start position><command><end position>

  → command可以是d, y, gU, gu...

示例： 0y$ 或 ,y$

Vim 进阶篇

- 复制

  yy, yw, ye, y{n}w, {n}yy, ygg, y{n}gg, yG, yf{char}, yt{char}

  Y = yy

- 粘贴

  p, P

Vim 进阶篇

- 查找
  - 向后查找： /
  - 向前查找： ?
  - 查找下一个： n, N

Vim 进阶篇

- 替换
- :s/{old}/{new}　　→ 替换当前行第一个{old}为{new}
- :s/{old}/{new}/g　　→ 替换当前行所有的{old}为{new}
- :s/{old}/{new}/c　　→ 替换当前行所有的{old}为{new}, 每次提醒

- :%s/{old}/{new}　　→ 替换所有行的{old}为{new}
- :{start-row-num},{end-row-num}s/{old}/{new}
- → 替换{start-row-num}行至{end-row-num}行的{old}为{new}
- :{n},$s/{old}/{new}　→ 替换替换第{n}行开始到最后一行中每一行所有的{old}为{new}

- :s#{old}#{new}　　→ / 不会作为分隔符

Vim 高级篇

- 翻页
- 移动屏幕
- 寄存器
- 书签
- 宏
- Visual模式
- 列编辑
- 分屏

Vim 高级篇

翻页

- 向下翻页

  Ctrl + f → （forward）向下翻页

- 向上翻页

  Ctrl + b → （backward）向上翻页

- 向下翻半页

  Ctrl + d → （down）向下翻半页

- 向上翻半页

  Ctrl + u → （up）向上翻半页

Vim 高级篇

移动屏幕

- 使光标所在行位于屏幕中间

  zz

- 所光标所在行位于屏幕顶部

  zt

- 所光标所在行位于屏幕底部

  zb

Vim 高级篇

寄存器

提供无与伦比的寄存功能，提供10类共48个寄存器。最常见的 y 操作

- 寄存器分类
  1. **匿名寄存器** ""
  2. 编号寄存器 "0 到 "9
  3. 小删除寄存器 "-
  4. 26个命名寄存器 "a 到 "z
  5. 3个只读寄存器 ":, "., "%
  6. Buffer交替文件寄存器 "#
  7. 表达式寄存器 "=
  8. 选区和拖放寄存器 "*, "+, "~
  9. 黑洞寄存器 "_
  10. 搜索模式寄存器 "/

Vim 高级篇

书签：

文件书签是你标记文件中的不同位置，然后可以在文件内快速跳转到你想要的位置。而全局书签是标记不同文件中的位置。也就是说你可以在不同的文件中快速跳转

- 设置书签

  m{a-z} → 小写字母为文件书签，大写字母为全局书签

- 跳转书签

  `{a-z} 或 '{a-z}

- 查看书签

  :marks

- 删除书签

  :delm{marks} →删除一个书签

  :delm! → 删除所有书签

Vim 高级篇

宏：

指某种特定顺序的一系列操作，我们可以录制自己的操作序列，然后重复这个序列多次，以简化某种重复的操作。vim宏有录制和播放的过程。

- 设置宏

  开始录制：q{a-z}

  完成录制：q

- 使用宏

  @{a-z} 或 @@

- 示例

  qmxjq 100@m

Vim 高级篇

Visual可视化模式：

便于选取文本。

- 字符模式

  v

- 行模式

  V

- 块模式

  Ctrl + v

Vim 高级篇

分屏：

- 垂直分屏(N为分几屏): vim -ON file1 file2
- 水平分屏: vim -oN file1 file2
- 关闭分屏：Ctrl + w c 或 Ctrl+w q
- 水平分屏（针对打开的文件再打开一个新文件）

  :sp filename
- 垂直分屏（针对打开的文件再打开一个新文件）

  :vsp filename

Vim 高级篇

分屏之移动光标：

- 移动到右边的屏：ctrl +w l
- 移动至左边的屏：ctrl+w h
- 移动到上边的屏：ctrl+w k
- 移动到下边的屏：ctrl+w j

- 其他：调整屏幕尺寸 ctrl+w =, ctrl+w +, ctrl+w -

命令模式下其他技巧

- 跳转到指定的行 :{int-number}
- 设置显示行号

  :set nu[mber] set nonu[mber] :set nu[mber]!
- 语法高亮 :syn[tax] on :syn[tax] off
- 设置不可见字符 :set list :set nolist
- 搜索设置

  :set ignorecase → 忽略大小写

  :set noignorecase → 取消忽略大小定

  :set hlsearch → 高亮显示搜索的匹配结果

  :set nohlsearch → 取消高亮显示搜索的匹配结果

长安行天下

CHANGAN DRIVES THE WORLD