Wiki

发布 *1.0*

欧阳松

2019 年 12 月 22 日

善假于物

CHAPTER 1

开发工具

1.1 搜索 RG

最快的搜索工具

Header row, column 1 (header rows optional)	Header 2	Header 3	Header 4
body row 1, column 1	column 2	column 3	column 4
body row 2	…	…	

1.2 分屏 Tmux

1.2.1 安装

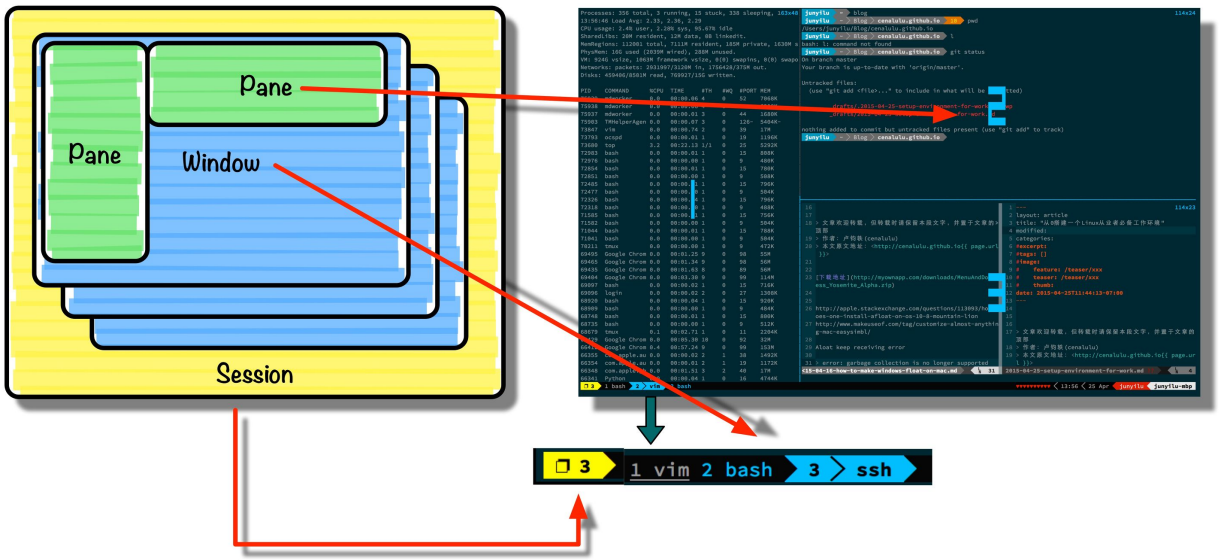
一般都是安装在服务端，所以主要记录 Linux 版本的安装过程。这里主要记录没有 ROOT 的时候安装最新版 Tmux，因为有 ROOT 的话有很多种办法安装到最新版的软件。主要需要注意的地方就是 configure 和 make 的时候要添加寻找头文件的地方。

```
$ mkdir $HOME/.local
$ curl -LO https://github.com/tmux/tmux/releases/download/3.0a/tmux-3.0a.tar.gz
$ curl -LO https://github.com/libevent/libevent/releases/download/release-2.1.11-
↪stable/libevent-2.1.11-stable.tar.gz
$ curl -LO https://ftp.gnu.org/gnu/ncurses/ncurses-5.9.tar.gz
```

(下页继续)

(续上页)

```
$ tar zvxf libevent-2.1.11-stable.tar.gz && cd libevent-2.1.11-stable
$ ./configure --prefix=$HOME/.local --disable-shared
$ make && make install && cd ..
$ tar zvxf ncurses-5.9.tar.gz && cd ncurses-5.9/
$ ./configure --prefix=$HOME/.local
$ make && make install && cd ..
$ tar zvxf tmux-3.0a.tar.gz && cd tmux-3.0a/
$ ./configure CFLAGS="-I$HOME/.local/include -I$HOME/.local/include/ncurses" LDFLAGS=
↪"-L$HOME/.local/lib -L$HOME/.local/include/ncurses -L$HOME/.local/include" --prefix=
↪$HOME/.local
$ CPPFLAGS="-I$HOME/.local/include -I$HOME/.local/include/ncurses" LDFLAGS="-static -L
↪$HOME/.local/include -L$HOME/.local/include/ncurses -L$HOME/.local/lib" make
$ make install && cd ..
$ export PATH=$HOME/.local/bin:$PATH
$ tmux -V
tmux 3.0a
```



1.2.2 配置

基于网上很热门的配置 gpakosz/.tmux 进行修改。主要修改的地方是：

```
# 使用 vi 键位
set -g status-keys vi
set -g mode-keys vi

# PREFIX-Q 显示编号保留时间，单位 ms
```

(下页继续)

(续上页)

```
set -g display-panes-time 5000

# 判断终端是否支持真彩色决定 Tumx 开启真彩色
tmux_conf_theme_24b_colour=true
```

1.2.3 Tmux Plugin Manager

```
$ git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm
$ bash ~/.tmux/plugins/tpm/bin/install_plugins
```

然后修改 .tmux.conf.local 文件。

```
setenv -g TMUX_PLUGIN_MANAGER_PATH '~/.tmux/plugins'
# List of plugins
set -g @tpm_plugins '          \
tmux-plugins/tpm             \
tmux-plugins/tmux-resurrect  \
tmux-plugins/tmux-continuum  \
'
# tmux-resurrect
set -g @resurrect-dir '~/.tmux/resurrect'

# 初始化 TPM 插件管理器 (放在配置文件的最后)
run '~/.tmux/plugins/tpm/tpm'
```

1.2.4 参考链接

https://gist.github.com/ryin/3106801

1.3 编辑器 Vim

为了使用更加现代化的 Vim 插件以及异步等特性，推荐使用 Vim8。

1.3.1 安装

macOS

```
$ brew install vim
```

Linux

```
$ curl -LO https://github.com/vim/vim/archive/v8.1.2418.tar.gz
$ tar zvxf v8.1.2418.tar.gz && cd vim-8.1.2418
$ LDFLAGS=-L$HOME/.local/lib ./configure --with-features=huge \
> --enable-multibyte \
> --enable-cscope \
> --prefix=$HOME/.local \
> --with-compiledby="ouyangsong" \
> --enable-pythoninterp=yes \
> --with-python-config-dir=/usr/lib/python2.7/config-x86_64-linux-gnu \
> --enable-fail-if-missing
```

如果想直接通过 APT 安装带有 Python 支持的 Vim，可以参考下面的命令。

```
$ sudo apt-get remove --purge vim vim-runtime vim-gnome vim-tiny vim-common vim-gui-
↪common
$ sudo apt install vim-nox
```

如果使用 Linuxbrew 管理软件的话，参考上面 macOS 的安装方法。

1.3.2 依赖

必须的依赖

- Node.js & npm
- ccls
- universal-ctags

可选的依赖

- ripgrep 强大的搜索软件
- protoc-gen-lint proto 文件的检测工具
- flake8 Python 代码风格检测

1.3.3 插件

网上很多 Vim 配置由于年久失修，其中推荐的插件很多没有使用 Vim8 的新特性或者性能存在问题。下面介绍日常编程中使用频率高的插件。

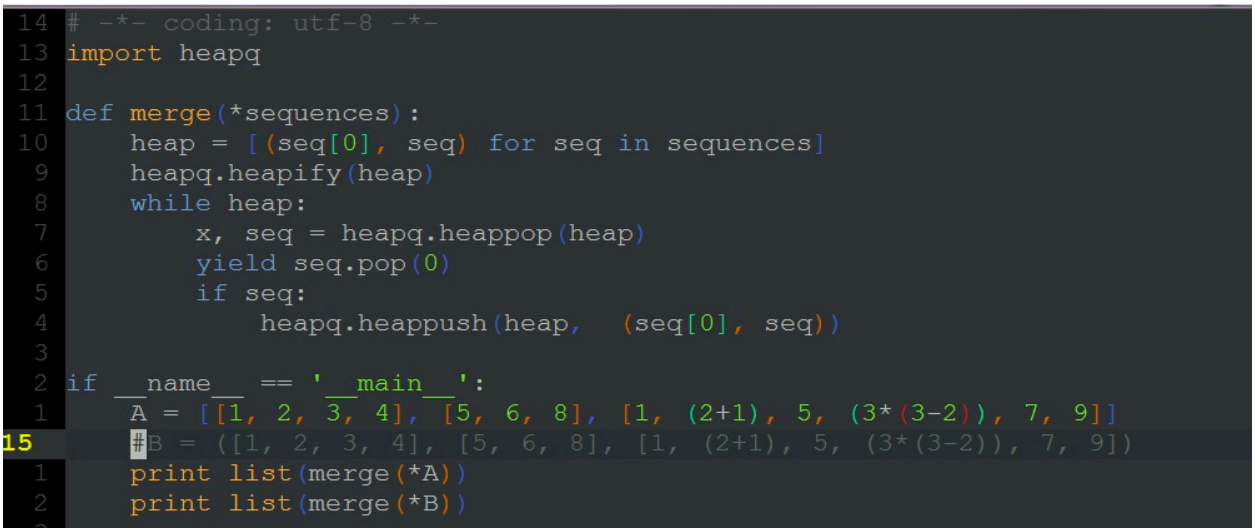
外观

- gruvbox-community/gruvbox

gruvbox 项目之前的作者没有继续维护，社区版本还有热心志愿者继续维护该项目。所以如果想用 gruvbox 主题的话，推荐使用社区版。

- luochen1990/rainbow

通过将不同层次的括号高亮为不同的颜色，帮助你阅读世界上最复杂的代码。



- Yggdroot/indentLine

显示代码缩进的对齐线，对于写 Python 这类依靠缩进判断语法块的语言还是很有用的。建议只针对需要的文件类型开启，否则在部分文件类型，比如 JSON 上会导致显示有问题。

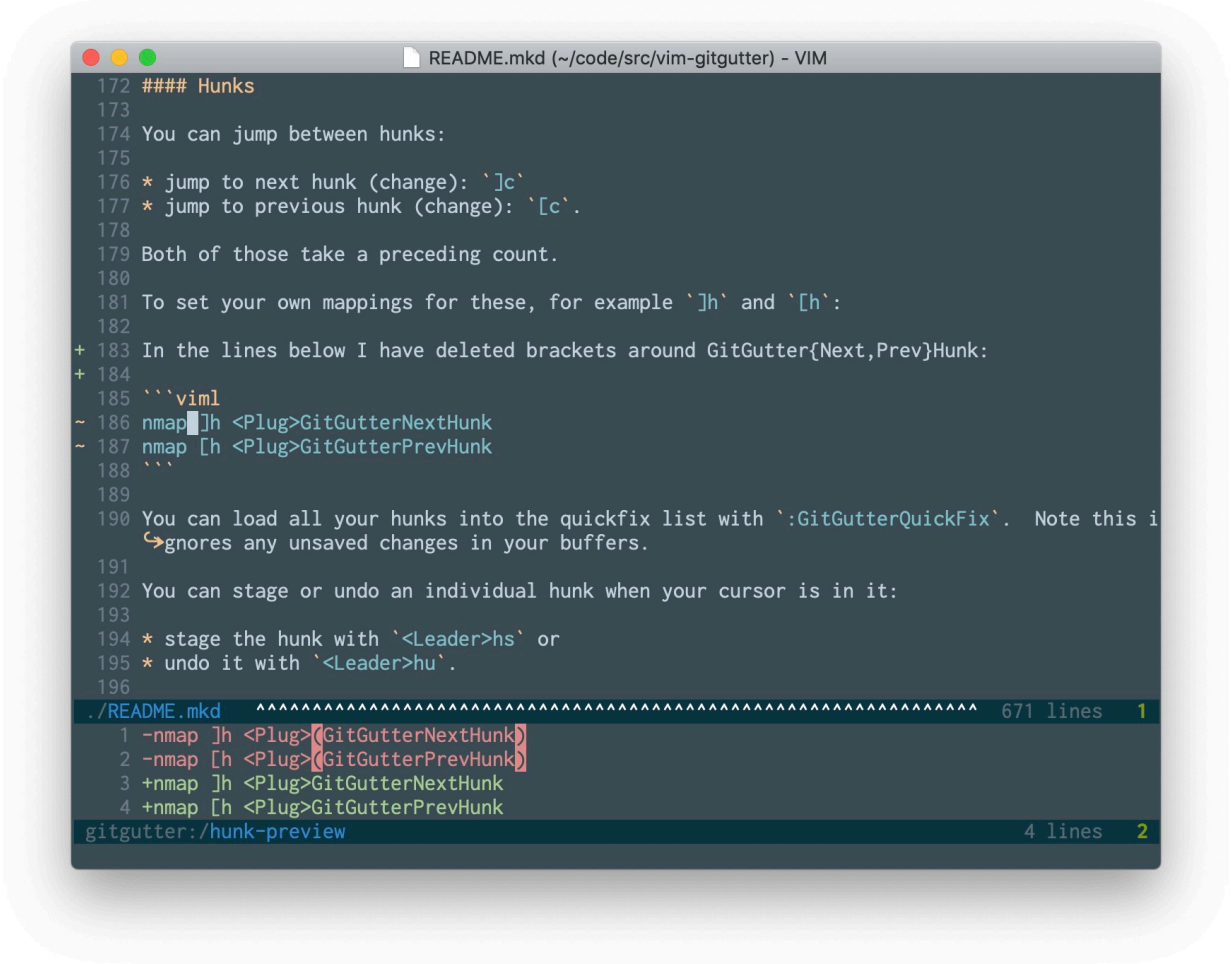
```
function indentLineGuide () {
    // Show off the more subtle glyph for indentLine in this codeblock
    while (indentLine !== shownOff) {
        if (!indentedEnough) {
            indentSomeMore();
        }
        else {
            switch(indentedAmount) {
                case 'A lot':
                    indentLess();
                    break;

                case 'Not much':
                    indentMoreStill();
                    break;

                case 'Just right':
                default:
                    stopIndenting();
            }
        }
    }
}
```

- vim-gitgutter

在左边显示 Git 中文件中每一行被修改的状态。



- itchyny/lightline.vim

轻量级但是功能强大的状态栏插件，可以配合显示其他插件的一些插件。

增强

- dense-analysis/ale

使用 Language Server Protocol 对代码进行错误检查，以及支持对应的修复。需要安装对应的代码检测软件，比如对 C/C++ 代码检查可以安装 ccls。

- scrooloose/nerdcommenter

针对不同的代码，自动选择相对应的注释符号来实现快速注释与取消注释某些代码。

- tpope/vim-surround

对 Vim 中本身就有的范围词性进行扩充，支持更多范围选择。

- tpope/vim-repeat

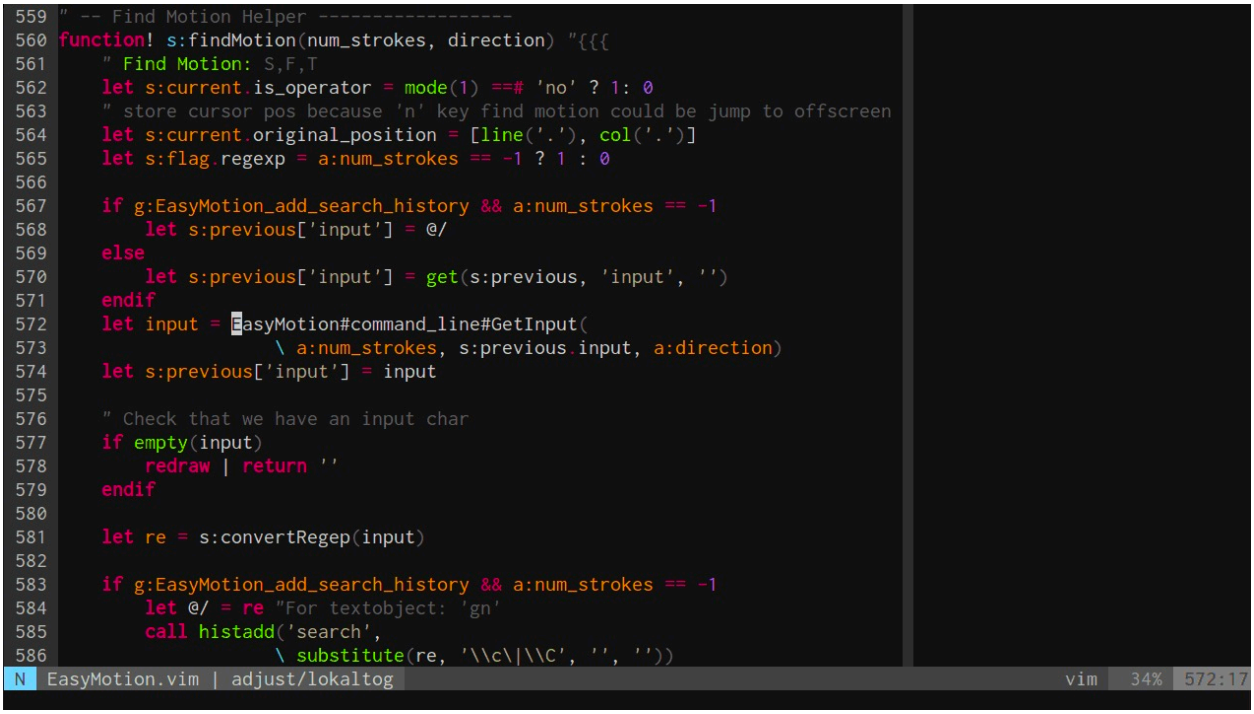
将 Vim 中某些本身只能重复一次的语句扩充至可以无限重复。

- jiangmiao/auto-pairs

自动补全括号等其他需要闭合的符号。

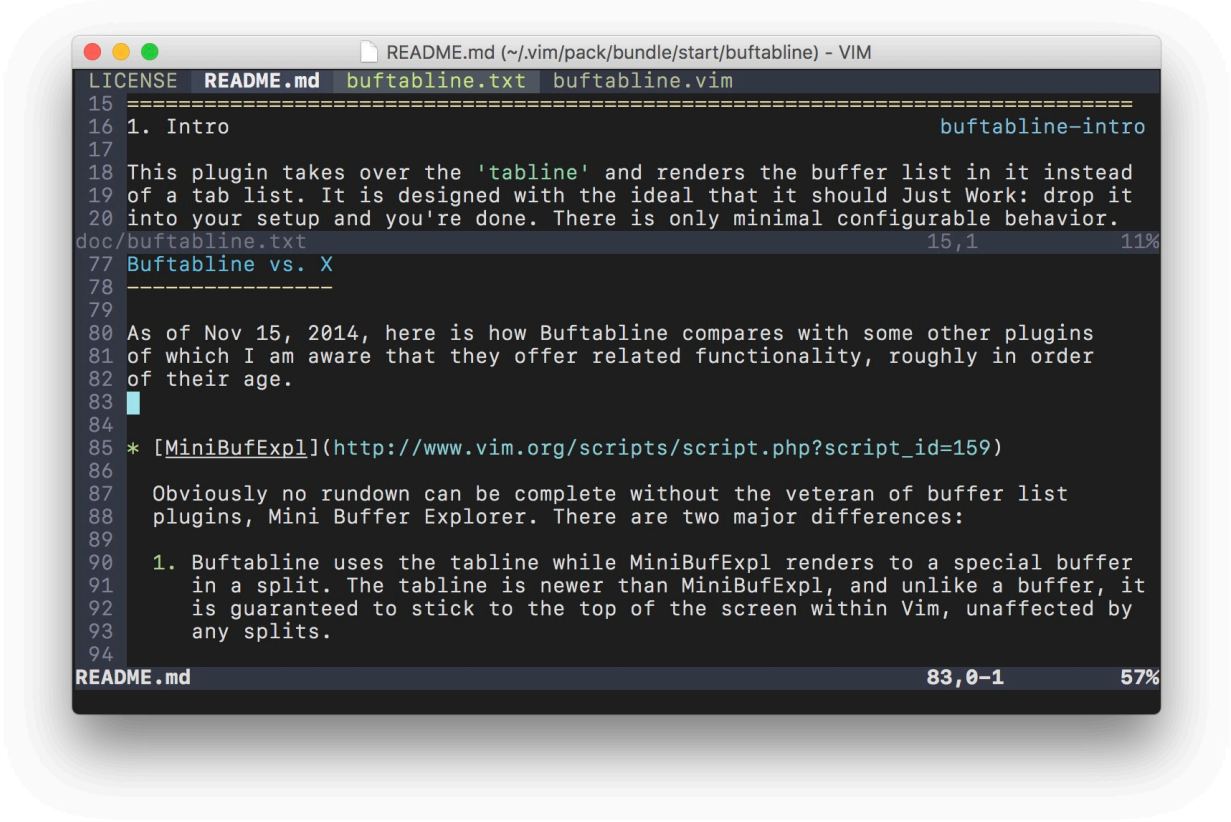
- easymotion/vim-easymotion

实现快速移动，启动快速移动后，可以通过两个字符就可以移动到任何位置。



- ap/vim-buftabline

在顶部用 tab 的方式显示已经存在的 buffer。这个功能其实 vim-airline 已经自带，但是如果使用 lightline.vim 就需要安装本插件。



- bronson/vim-trailing-whitespace

自动移除多余的空格，包括尾部的空格和含有空格的行。

- rhysd/accelerated-jk

当长按 J 或者 K 键进行上下移动时会不断加速移动。

补全

- neoclide/coc.nvim

异步的代码补全引擎，可以管理和安装各个语言的插件。强力推荐!

- honza/vim-snippets

提供一些常用的代码片段。

1.3.4 参考链接

https://askubuntu.com/a/602249

https://github.com/ycm-core/YouCompleteMe/wiki/Building-Vim-from-source

https://www.cnblogs.com/freeweb/p/9321520.html