

Tools Solutions

陈磊

April 13, 2017

Contents

1	持续集成	1
1.1	travis-ci	1
1.1.1	travis-ci 使用流程	1
1.1.2	添加编译状态图标	1
1.1.3	travis 配置文件样本	1
2	文档处理工具	2
2.1	Pandoc	2
2.1.1	markdown 多个文档转换	2
2.1.2	markdown 转 LaTeX, 用 listings 包替换 verbatim	2
2.2	LaTeX	2
2.2.1	Windows 上编译缓慢, 经常卡在 eu1lmr.fd	2
2.2.2	listings 代码换行	3
3	编辑器及插件	3
3.1	Visual Studio Code	3
4	工具安装	3
4.1	nodejs	3
4.2	git	4
4.2.1	Ubuntu	4
4.2.2	CentOS 从源码安装 git	5

1 持续集成

1.1 travis-ci

1.1.1 travis-ci 使用流程

travis-ci 结合 github 使用, 每次提交自动执行编译或测试任务.

1. <travis-ci.org> 用 github 账户授权登录
 2. travis-ci 上添加需要持续集成的 github repo
 3. github repo 中添加 .travis.yml 配置, 用以配置 travis 所要执行的任务和环境
 4. repo 更改完成, 提交, 可在 travis-ci 上实时查看任务执行过程
- 备注:

1.1.2 添加编译状态图标

```
1 [![Build Status](https://travis-ci.org/HereChen/Tools-  
  ↳ Solutions.svg?branch=master)](https://travis-ci.org  
  ↳ /HereChen/Tools-Solutions)
```

1.1.3 travis 配置文件样本

此库的 .travis.yml 样本

```
1 sudo: required  
2 dist: trusty      # ubuntu 14.04  
3  
4 # install  
5 before_install:  
6   - sudo apt-get update  
7   - sudo apt-get -y install texlive-full  
8   - sudo apt-get -y install pandoc  
9  
10 before_script:  
11   - chmod +x build.sh  
12  
13 script:  
14   - ./build.sh
```

2 文档处理工具

2.1 Pandoc

Pandoc 主页 <http://pandoc.org>. Pandoc 可以实现不同格式文档的相互转换, 支持的格式包括 markdown、HTML、LaTeX、docx、EPUB ODT 等等.

2.1.1 markdown 多个文档转换

```
1 pandoc src/markdown/*.md -o build/latex/content.tex
```

2.1.2 markdown 转 LaTeX, 用 listings 包替换 verbatim

```
pandoc -listings src/markdown/*.md -o build/latex/content.tex
```

2.2 LaTeX

2.2.1 Windows 上编译缓慢, 经常卡在 eu1mr.fd

删除 `texlive\2016\texmf-var\fonts\cache` 文件夹下字体缓存, 重新生成字体缓存, 执行 `texlive\2016\bin\win32\fc-cache.exe`¹.

2.2.2 listings 代码换行

添加 `breaklines=true` 后, 还是存在没有空格的 url 无法换行, 需设置 `breakatwhitespace=false`. 另外, 可以在代码换行的地方添加箭头标识换行².

```
1 \lstset{
2   breaklines=true,
3   % 箭头标识换行
4   postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\color
5     {red}\hookrightarrow\space}},
6   breakatwhitespace=false
7 }
```

¹LaTeX 编译卡顿怎么办?

²Pandoc: Markdown to PDF, without cutting off code block lines that are too long, Code not wrapping in listings even though `breaklines=true`, `lstlisting` line wrapping

3 编辑器及插件

3.1 Visual Studio Code

插件

- **Markdown TOC**, markdown 文本的目录生成, 可应用于 wiki 或博客目录生成.
- **REST Client**, 在文本编辑器中测试 API
- **Project Manager**, 类似于 Sublime Text 快捷切换项目目录

4 工具安装

4.1 nodejs

安装的方式包括: 从源码编译安装, 直接下载二进制, 依靠工具直接从库下载安装.

从源码安装

```
1 curl -o node-v6.10.0.tar.gz https://nodejs.org/dist/v6
   ↳ .10.0/node-v6.10.0.tar.gz
2 tar -xvzf node-v6.10.0.tar.gz
3 cd node-v6.10.0
4 ./configure
5 make
6 make install
```

CentOS 下载二进制安装

```
1 # 64bit node install
2 curl -o node-v6.10.0-linux-x64.tar.xz https://nodejs.org/
   ↳ dist/v6.10.0/node-v6.10.0-linux-x64.tar.xz
3 mv /opt/
4 cd /opt/
5 tar -xvJf node-v6.10.0-linux-x64.tar.xz
6 mv node-v6.10.0-linux-x64 node
7
8 vi ~/.bash_profile
9 # 添加以下内容到 .bash_profile
10 # export NODE_HOME=/opt/node
```

```
11 # export PATH=$NODE_HOME/bin:$PATH
12 source ~/.bash_profile
13
14 node -v
```

Ubuntu 从库中下载安装

```
1 curl -sL https://deb.nodesource.com/setup_6.x | sudo -E
  ↪ bash -
2 sudo apt-get install -y nodejs
```

4.2 git

4.2.1 Ubuntu

安装最新版的 git³.

```
1 sudo add-apt-repository ppa:git-core/ppa
2 sudo apt-get update
3 sudo apt-get install git
```

如果上面执行报错 Errors were encountered while processing: runit
↪ git-daemon-run, 尝试执行下面命令.

```
1 sudo apt-get purge runit
2 sudo apt-get purge git-all
3 sudo apt-get purge git
4 sudo apt-get autoremove
5 sudo apt update
6 sudo apt install git
```

4.2.2 CentOS 从源码安装 git

执行 `yum install git` 安装的版本比较老, 安装最新版可从源码安装⁴.

```
1 # 依赖工具安装
2 yum install curl-devel expat-devel gettext-devel openssl-
  ↪ devel zlib-devel
3 yum install gcc perl-ExtUtils-MakeMaker
```

³Installing Latest version of git in ubuntu, How to fix processing with runit and git-daemon-run [duplicate]

⁴How to install latest version of git on CentOS 6.x/7.x , 起步 - 安装 Git

```
4
5 # git 源码下载
6 # https://www.kernel.org/pub/software/scm/git/
7 curl -o git-2.9.3.tar.xz https://www.kernel.org/pub/
    ↪ software/scm/git/git-2.9.3.tar.xz
8 tar -xvJf git-2.9.3.tar.xz
9
10 # 编译安装 git
11 cd git-2.9.3
12 make prefix=/usr/local all
13 sudo make prefix=/usr/local install
```