

本文是关于通过 pandoc 转换 markdown 格式为 html 和 pdf 的，并由[travis-ci.org](https://travis-ci.org)自动编译转换生成到 Github 上。目的是因为我的主页目前没有富文本编辑器，使用 markdown 写文章是一个不错的选择，转换后再把 html 代码通过后台发布即可。因为都没有接触过类似的转换，所以花了很多时间，感觉必须要整理一下的。

## 必要的几个条件

- 符合规则的[markdown](#)文件
- [pandoc](#) ( 提供基本转换环境 )
- texlive & texlive-xetex & texlive-xetex-extra ( 转换为 pdf )
- [中文字体](#)
- 创建 Makefile 文件以实现批量转换
- travis-ci.org 的帮助 ( 提供了 Ubuntu12.04.5LTS 的虚拟化编译平台 )

## 安装基本的转换工具

针对 Ubuntu12.04.5 下安装 pandoc 如果使用 cabal 处理依赖关系总是出现版本不符合的问题，不知道如何解决，反而直接下载[deb](#) 包来安装却非常简单，适合 travis-ci 虚拟机环境：

```
wget https://github.com/jgm/pandoc/releases/download/1.17.1/pandoc-1.17.1-2-amd64.deb
sudo dpkg -i pandoc-1.17.1-2-amd64.deb
```

本机电脑使用 Gentoo 安装依赖关系处理非常好。

## 使 **pandoc** 支持中文 **pdf** 转换

在一知半解的情况下，我使用 xelatex 这个引擎为 Pandoc 提供 pdf 转换中文支持，Ubuntu12.04.5LTS 需要 texlive-xetex texlive-latex-extra 这两个包，安装这两个包需要添加 texlive-backports/ppa 这个库。

```
sudo apt-get install python-software-properties #这个在Travis下是不需要的
sudo add-apt-repository ppa:texlive-backports/ppa
sudo apt-get update
sudo apt-get install texlive-xetex texlive-latex-extra
```

默认情况下，转换中文时会出现不显示中文字体的情况，那是因为 pandoc 下的 XeLaTeX 模版没有默认设置中文，需要做如下的添加，[完整的模版](#)：

```
\usepackage{fontspec}          % 允许设置字体
\usepackage{xeCJK}             % 分别设置中英文
\setCJKmainfont{WenQuanYi Micro Hei} % 设置中文字体，当然前提是有这个字体存在
\setmainfont{DejaVu Sans}      % 设置英文字体
\setromanfont{DejaVu Sans}     % 字体
\setmonofont{DejaVu Sans Mono}
\linespread{1.2}\selectfont    % 行间距
\XeTeXlinebreaklocale "zh"     % 针对中文自动换行
\XeTeXlinebreakskip = 0pt plus 1pt % 字与字之间加入0-1pt，保证对齐
\parindent 0em                 % 段落缩进
\setlength{\parskip}{20pt}     % 段落之间距离
```

[这个](#)页面有关于 LaTeX 的一些语法结构。通过 hypersetup 可以设置转换之后 pdf 的一些属性，比如作者、标题、标签等：

```
\hypersetup{breaklinks=true,
             bookmarks=true,
             pdfauthor={${author-meta$},
             pdftitle={${title-meta$},
             pdfsubject={${subject-meta$},
             pdfkeywords={${keywords-meta$},
             colorlinks=true,
             urlcolor=${if(urlcolor)$$urlcolor$$else$blue$endif$,
             linkcolor=${if(linkcolor)$$linkcolor$$else$magenta$endif$,
             pdfborder={0 0 0}}
```

## 如何使用 **Makefile**

可以参考的一个说明是[这个](#)，用到了里面所写的静态模式、（自动化）变量和伪目标。

```
SOURCES := $(wildcard *.md)
OBJECTS := $(patsubst %.md, %.html, $(wildcard *.md))
OBJECTS_PDF := $(patsubst %.md, %.pdf, $(wildcard *.md))
```

```
all: html pdf
```

```
html: $(OBJECTS)
```

```
pdf: $(OBJECTS_PDF)
```

```
$(OBJECTS): %.html: %.md
```

```
    pandoc $< -o $@ && mv *.html git/
```

```
$(OBJECTS_PDF): %.pdf: %.md
```

```
    pandoc --template=./template.tex --latex-engine=xelatex $< -o $@ && mv *.pdf git/
```

## 配置 **Travis** 以实现自动化转换

目前即使配置到了自动换转换，也还是一个很繁琐的过程，至少还需要复制到个人主页后台再粘贴的一个过程，先用着吧。Travis 可以在获取了 Github 相应权限后监视 Github Repository 动态，每当有 commit 时，便会克隆对应 Repository 到一个虚拟环境，根据预先设置好的.travis.yml 文件下的内容进行配置和脚本运行。如果在 Github 下生成一个支持公共资源的 Token，并在 Travis 后台设置对应的环境变量，就可以很方便地将转换完成的 html 和 pdf 文档 push 到另外一个 Repository 下，不重复 push 到原先的 Repository 的原因是避免无限循环。一个可用的.travis.yml：

```
language: ruby
```

```
rvm:
```

```
  - 2.2
```

```
cache:
```

```
  directories:
```

```
    - cache
```

```
before_install:
```

```
  - sudo add-apt-repository ppa:texlive-backports/ppa -y
  - sudo apt-get update
```

```
install:
```

```
  - wget https://github.com/jgm/pandoc/releases/download/1.17.1/pandoc-1.17.1-2-amd64.deb
  - sudo dpkg -i pandoc-1.17.1-2-amd64.deb
```

```
- sudo apt-get install ttf-wqy-microhei texlive-xetex texlive-latex-extra
```

script:

```
- git config --global user.email "$GIT_EMAIL"  
- git config --global user.name "$GIT_NAME"  
- git clone --depth 1 https://$GIT_TOKEN@github.com/Bekcpear/bekcpear.github.io git  
- pwd  
- ls -a  
- make all  
- cd git  
- ls -a  
- git config --global push.default matching  
- git add -A .  
- git commit -m "update from travis"  
- git push --quiet
```

after\_success:

```
- echo 'Welcome to my home page: https://bekcpear.io'
```

参考:

- [farseerfc/farseerfc/.travis.yml](#)
- [如何在 Linux 下使用 Markdown 进行文档工作](#)
- [LaTeX/Colors](#)
- [Markdown 语法中文](#)
- [关于一个 xelatex 的 tightlist 的报错](#)