

一份简短的关于 L^AT_EX 安装的介绍¹

嘯行²

2020 年 5 月 1 日

¹<https://github.com/OsbertWang/install-latex>

²ranwang.osbert@outlook.com

摘要

在 QQ 群 91940767、478023327 和 640633524 中, 经常有群友咨询如何安装 \LaTeX . 实际上, 用户安装的是 \LaTeX 的**发行版**和相关的**编辑器**. 本文将介绍在系统中**不存在其他 \LaTeX 发行版 (如 \CTEX 套装) 的前提下**, 在 Windows 10、Ubuntu 18.04 和 macOS 系统中安装 \TeX Live (macOS 中介绍 \MacTeX)、升级宏包、编译简易文档等相关操作, 并多以介绍命令行操作为主. 有关 \MiKTeX 的安装, 可以参考 [\$\text{\MiKTeX}\$ 的基本使用](#).

本文还将简要介绍几款常见编辑器的使用方法, 其他编辑器如 [VS code](#) 和 [Sublime Text](#), 用户可自行了解它们的使用方法.

除在本地安装 \LaTeX 发行版和编辑器之外, 本文还将额外补充使用 [Overleaf](#) 的相关内容.

本文所涉及到的代码需结合上下文说明, 不能简单地复制粘贴. 红色文字都是可点的超链接, 可直接跳转. [菜单](#) 表示软件菜单. [k](#) 表示键盘按键. 建议用户阅读 [\$\text{\LaTeX}\$ 2_ε 安装 & 新手指点 FAQ](#) 和 [lshort-zh-cn](#) 以更全面地了解基础内容.

本文大部分内容是个人最近一段时间的使用总结, 其中难免有不甚合理或晦涩难懂的部分. 若用户在阅读本文档的过程中有任何意见和建议, 请发[邮件](#)或在 [GitHub](#) 中提 issue.

本手册发布后, [Dongsheng Deng](#), [muzimuzhi](#), [Xiangdong Zeng](#) 对本手册提出了很好的建议, 并提供了帮助, 其中, 有关 macOS 的内容最初由 Xiangdong Zeng 草拟完成. 在此一并感谢.

目录

1	Windows 10 系统	3
1.1	安装 T _E X Live	3
1.1.1	下载镜像一直不成功的解决办法	6
1.2	卸载 T _E X Live	6
1.2.1	手动卸载	6
1.3	跨版本升级 T _E X Live	7
1.4	升级宏包	7
1.4.1	tlmgr 本身无法成功升级	7
1.4.2	升级到一半停止	7
1.5	安装宏包	8
1.6	编译文档	8
2	Ubuntu 18.04 系统	10
2.1	安装 T _E X Live	10
2.2	卸载 T _E X Live	12
2.3	跨版本升级 T _E X Live	13
2.4	升级宏包	13
2.5	安装宏包	13
2.6	编译文件	14
2.6.1	无法使用 xelatex 命令	14
3	macOS	16
3.1	安装 Homebrew	16
3.1.1	Xcode	16
3.2	安装 MacT _E X	16
3.3	卸载 MacT _E X	17
3.4	跨版本升级 MacT _E X	17
3.5	升级宏包	17
3.6	安装宏包	17
3.7	编译文件	17

4	Windows Subsystem for Linux	19
4.1	安装 T _E X Live	19
4.1.1	不懂 vim 的处理方法	21
4.2	卸载 T _E X Live	22
4.3	跨版本升级 T _E X Live	22
4.4	升级宏包	22
4.5	安装宏包	22
4.6	编译文件	22
4.6.1	L ^A T _E X 2 _ε 版本不匹配导致 xelatex 失败	23
4.7	尚未圆满解决的问题	23
5	使用编辑器	25
5.1	Windows 10 系统	25
5.1.1	T _E Xworks	25
5.1.2	T _E Xstudio	26
5.2	Ubuntu 18.04 系统	29
5.3	Windows Subsystem for Linux	30
5.4	macOS	30
5.4.1	T _E XShop	31
5.4.2	T _E Xstudio	31
5.5	多款编辑器对比参考	31
6	Overleaf	33
6.1	注册 Overleaf	33
6.2	使用 Overleaf	33
6.3	升级项目	34
6.4	学习与帮助	34
附录 A	有关 WSL 的一点补充内容	35
附录 B	离线安装宏包	38
附录 C	版本信息	39

1 Windows 10 系统

1.1 安装 T_EX Live

在安装 T_EX Live 之前, 在文件夹菜单栏 **查看**》**显示/隐藏** 中选择 **文件扩展名**.

用户可以从最近的镜像¹下载 T_EX Live 的 **iso 镜像文件**, 也可以找国内的源自行下载²: **清华大学**, **北京理工大学**, **北京交通大学**, **上海交通大学**, **中国科学技术大学**, **南京大学**, **华中科技大学**, **重庆大学**, **兰州大学**, **腾讯云**, **华为云**, **阿里云**. 假设镜像文件被下载到本地后存放路径为 **X:\Y**³. 下载完毕后, 用户打开 cmd 窗口⁴, 执行

```
1 echo %path:;=&echo.%
```

查看环境变量. 若 **C:\Windows\system32** 不在结果中⁵, 则关闭 cmd 窗口, 将 **C:\Windows\system32** 添加到环境变量中⁶. 根据以往经验, 安装过 **CT_EX** 套装的用户往往会有 **system32 丢失**的问题. 另外, 如果环境变量中有 **mingw** 或 **jdk** 相关的内容, 也请暂时删除⁷. 同时国内常见的 2345 好压也会对安装构成影响, 建议删除它并在安装 T_EX Live 后更换其他同类软件⁸. 继续打开 cmd 窗口依次执行以下代码

```
1 cd /d X:\Y
2 certutil -hashfile texlive2020.iso md5
```

若系统显示

```
1 MD5 的 texlive2020.iso 哈希:
2 a31c500c7830d901363c5fa0e65cc18d
3 CertUtil: -hashfile 命令成功完成
```

则表明下载的镜像文件正常.

接下来, 将镜像文件加载至虚拟光驱⁹. 假设加载后的路径为 **Z:**. 在卸载了国内第三方安全软件的前提下, 以管理员身份打开 cmd 窗口并执行¹⁰

¹这里所谓的最近, 其实是由系统判定的, 实际上系统可能会误判

²新版本发布时, 各镜像网站同步的进度会有不同

³这里, **X** 表示盘符, 如 **C**、**D**、**E** 等; **Y** 表示文件夹; 默认路径名称为不带空格的英文

⁴点击 **[Win]**, 输入 **cmd**, 点击 **[Enter]**; 或在任意文件夹中按 **[Ctrl]+[L]**, 输入 **cmd**, 点击 **[Enter]**

⁵这里默认系统盘为 **C** 盘

⁶在 **[桌面]》[此电脑]》[属性]》[高级系统设置]》[环境变量]》[系统变量]》[Path]** 中添加

⁷具体原因参见 [stackexchange](#) 的解释, 待安装结束后, 再将 **mingw** 或 **jdk** 环境变量添加至 T_EX Live 的环境变量之后

⁸推荐 **WinRAR** 或 **7-Zip**

⁹Windows 10 默认双击镜像文件便可加载; 受某些压缩软件影响双击失效时, 可通过文件夹菜单栏 **[主页]》[打开]》[Windows 资源管理器]** 加载; 第三方虚拟光驱软件可考虑 **WinCDEmu** 或 **UltraISO**; 不鼓励解压缩镜像文件

¹⁰点击 **[Win]**, 输入 **cmd**, 点击 **[Ctrl]+[↑]+[Enter]**

```
1 cd /d Z:
```

切换路径. 接下来执行

```
1 install-tl-windows.bat --no-gui
```

即可看到如下结果

```
1 =====> TeX Live installation procedure <=====
2
3 =====> Letters/digits in <angle brackets> indicate <=====
4 =====> menu items for actions or customizations <=====
5
6 Detected platform: Windows
7
8 <B> set binary platforms: 1 out of 6
9
10 <S> set installation scheme: scheme-full
11
12 <C> set installation collections:
13     41 collections out of 41, disk space required: 6702 MB
14
15 <D> set directories:
16     TEXDIR (the main TeX directory):
17     C:/texlive/2020
18     TEXMFLOCAL (directory for site-wide local files):
19     C:/texlive/texmf-local
20     TEXMFSYSVAR (directory for variable and automatically generated data):
21     C:/texlive/2020/texmf-var
22     TEXMFSYSCONFIG (directory for local config):
23     C:/texlive/2020/texmf-config
24     TEXMFVAR (personal directory for variable and automatically generated data):
25     ~/.texlive2020/texmf-var
26     TEXMFCONFIG (personal directory for local config):
27     ~/.texlive2020/texmf-config
28     TEXMFHOME (directory for user-specific files):
29     ~/.texmf
30
31 <O> options:
32     [ ] use letter size instead of A4 by default
33     [X] allow execution of restricted list of programs via \write18
34     [X] create all format files
35     [X] install macro/font doc tree
36     [X] install macro/font source tree
37     [X] adjust search path
38     [1] add menu items, shortcuts, etc.
39     [1] update file associations
40     [X] install TeXworks front end
```

```

41 [X] after install, set CTAN as source for package updates
42
43 <V> set up for portable installation
44
45 Actions:
46 <I> start installation to hard disk
47 <P> save installation profile to 'texlive.profile' and exit
48 <H> help
49 <Q> quit
50
51 Enter command:

```

这时, 用户可按键盘 **[D]** 来更改安装路径, 也可直接按键盘 **[I]** 在默认路径中直接安装 T_EX Live. 按 **[D]** 后, 用户可看到

```

1 =====
2 Directories customization:
3
4 <1> TEXDIR: C:/texlive/2020
5     main tree: C:/texlive/2020/texmf-dist
6
7 <2> TEXMFLOCAL: C:/texlive/texmf-local
8 <3> TEXMFSYSVAR: C:/texlive/2020/texmf-var
9 <4> TEXMFSYSCONFIG: C:/texlive/2020/texmf-config
10
11 <5> TEXMFVAR: ~/.texlive2020/texmf-var
12 <6> TEXMFCONFIG: ~/.texlive2020/texmf-config
13 <7> TEXMFHOME: ~/texmf
14
15 Note: ~ will expand to %USERPROFILE%
16
17 Actions:
18 <R> return to main menu
19 <Q> quit
20
21 Enter command:

```

这时按键盘 **[1]**, 将看到

```

1 New value for TEXDIR [C:/texlive/2020]:

```

即可更改路径, 如 D:/texlive/2020, 更改后的路径名称要求是不带空格的英文. 接下来, 按键盘 **[R]** 即可回到初始的安装界面. 用户可通过阅读英文执行其他操作, 这里不再赘述. 至此, 用户只需耐心等待安装完成, 并且不要点击 cmd 窗口¹¹. 安装完成后, 关闭再打开 cmd 窗口¹², 执行

```

1 tex -v

```

¹¹ Windows 10 默认采取快速编辑模式, 一旦点击 cmd 窗口, 将显示“选择命令提示符”而导致进程暂停

¹² 正常安装 T_EX Live 后, 环境变量会变化, 关闭原 cmd 窗口再重新打开才能获取新的环境变量

显示

```
1 TeX 3.14159265 (TeX Live 2020/W32TeX)
2 kpathsea version 6.3.2
3 Copyright 2020 D.E. Knuth.
4 There is NO warranty. Redistribution of this software is
5 covered by the terms of both the TeX copyright and
6 the Lesser GNU General Public License.
7 For more information about these matters, see the file
8 named COPYING and the TeX source.
9 Primary author of TeX: D.E. Knuth.
```

即可认为安装顺利完成¹³。

1.1.1 下载镜像一直不成功的解决办法

目前发现有些用户使用 edge 浏览器时无法正常下载镜像文件。这时, 用户可以选择使用 `wget` 对镜像进行下载。具体用法是: 将 `wget` 所在目录添加到环境变量中。打开 `cmd` 窗口执行以下代码

```
1 wget http://mirrors.ctan.org/systems/texlive/Images/texlive2020.iso
```

即可下载最近的镜像。类似地, 也可以指定国内的源进行下载, 例如选择清华大学镜像

```
1 wget https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/texlive2020.iso
```

其他国内镜像只需要将链接替换即可, 链接在 1.1 节都已给出。

1.2 卸载 T_EX Live

在跨版本升级 T_EX Live 时, 通常需要卸载旧版 T_EX Live。在命令行执行

```
1 kpsewhich -var-value TEXMFROOT
```

可以查看 TEXMFROOT 的值, 该值即为 T_EX Live 的安装路径¹⁴。接下来, 用户只需执行¹⁵安装路径中的卸载批处理文件即可实现卸载, 如

```
1 C:\texlive\2020\tlpkg\installer\uninst.bat
```

结束以上步骤后, 手动删除用户文件夹中的 `.texlive2020` 子文件夹。

1.2.1 手动卸载

如果执行批处理文件 `uninst.bat` 出错, 用户也可手动删除安装文件夹, 之后再清理 T_EX Live 的环境变量, 并且清理注册表

¹³ 若未能正确显示结果, 则可能是因为环境变量未改变, 用户可手动添加 `<TEXDIR>\bin\win32` 至环境变量, `<TEXDIR>` 需与你方才的安装地址一致

¹⁴ 除这里提供的 TEXMFROOT, 还有很多其他变量, 可在 `texlive-zh-cn` 第 2.3 节处找到

¹⁵ 双击或在 `cmd` 中执行


```
1 HKEY_CURRENT_USER\Software\Classes\TL.bitmap2eps.convert.2020
2 HKEY_CURRENT_USER\Software\Classes\TL.DVIOUT.view.2020
3 HKEY_CURRENT_USER\Software\Classes\TL.TeXworks.edit.2020
```

1.3 跨版本升级 T_EX Live

目前在 tug.org 上面提供的说法是: Windows 没有类似 Unix 的升级程序, 需要进行新安装¹⁶.

1.4 升级宏包

安装完成后, 用户可以升级宏包以获得更好的使用体验. 下面将介绍使用命令行升级宏包的方法. 打开 cmd 窗口, 首先执行下面命令指定升级使用的镜像源. `ctan` 表示系统在升级时将自动寻求最近的源进行下载.

```
1 tlmgr option repository ctan
```

用户同样可以指定其他的镜像源, 方法是将上述的 `ctan` 改为以下网址: [清华大学](#), [北京理工大学](#), [北京交通大学](#), [上海交通大学](#), [中国科学技术大学](#), [南京大学](#), [华中科技大学](#), [重庆大学](#), [兰州大学](#), [腾讯云](#), [华为云](#), [阿里云](#). 接下来, 用户执行命令

```
1 tlmgr update --list
```

可查看目前源上可升级的宏包都有哪些. 高级用户可以根据自己的需求选择升级特定宏包. 建议初级用户直接升级全部宏包. 用户只需执行

```
1 tlmgr update --self --all
```

同时升级 `tlmgr` 本身和全部宏包.

部分用户会碰到升级不成功的情况, 这些情况我们将一一给出处理办法.

1.4.1 tlmgr 本身无法成功升级

遇到这种情况时, 用户需自行下载 [update-tlmgr-latest.exe](#), 然后再执行升级命令即可¹⁷.

1.4.2 升级到一半停止

这种情况下, 用户需要执行另外一个命令继续升级

```
1 tlmgr update --reinstall-forcibly-removed --all
```

¹⁶原文是: There is no comparable upgrade procedure for Windows. Doing a new installation is necessary.

¹⁷这里依旧让系统自动选择镜像源, 用户可以自行选择国内镜像以加快下载速度

1.5 安装宏包

在默认状态下, 用户将完整安装 $\text{T}_{\text{E}}\text{X}$ Live, 因此用户极少碰到需要手动安装宏包的情形. 同时, 在 [lshort-zh-cn](#) 中也明确提到, **如非万不得已, 尽量不要手动安装宏包**. 因此在这里我只介绍从源处安装宏包的命令. 假设用户想安装 `mcmthesis` 宏包, 只需在 `cmd` 执行

```
1 tlmgr install mcmthesis
```

需要注意的是, 用户一定要清楚所要安装的宏包名称, 并且在安装宏包前先确保镜像源设置正确.

1.6 编译文档

升级宏包完成后, 用户可以编译文档. 这里依旧使用命令行来完成这一过程.

首先, 用户需要在指定位置¹⁸建立一个 `tex` 文件:

```
1 mkdir E:\work_latex
2 cd /d E:\work_latex
3 notepad main.tex
```

第 1 行表示创建一个工作路径 `E:\work_latex`, 第 2 行表示进入工作路径, 第 3 行表示用记事本打开 `main.tex` 文件, 若文件不存在, 系统将询问用户是否创建该文件¹⁹. 在打开的记事本中编写一个最小示例

```
1 \documentclass{article}
2 \begin{document}
3   Hello \LaTeX{} World!
4 \end{document}
```

保存并退出. 接下来执行

```
1 pdflatex main
```

等待系统完成编译过程. 待编译完成后, 我们即可看到在 `E:\work_latex` 中出现了 `main.pdf` 文件和其他同名的辅助文件 `main.aux` 与 `main.log`.

对于中文文档, 可以在记事本中编写以下最小示例²⁰

```
1 \documentclass[UTF8]{ctexart}
2 \begin{document}
3   你好 \LaTeX{} 世界!
4 \end{document}
```

保存并退出. 接下来执行

```
1 xelatex main
```

等待系统完成编译过程. `xelatex` 编译命令配合 UTF-8 编码和 `ctex` 宏集已经是目前主流的中文处理手段. `xelatex` 可以使用系统内安装的字体, 用户在安装字体时需右键字体文件选择**为所有用户安装**.

¹⁸以下称工作路径

¹⁹Windows 10 较新版本可能对中文名友好一些, 但这里不建议用户使用中文命名工作路径和文件名

²⁰自 Windows 10 1903 版本开始, 记事本默认使用 UTF-8 编码; 若是之前的 Windows 10 版本, 请使用其他编辑器并规定文档编码为 UTF-8

总结一下, 在使用 L^AT_EX 时, 用户首先要在工作路径²¹中建立一个完整的 `tex` 文件²². 当 `tex` 文件内容确定后, 再保存文件并使用编译命令²³, 将 `tex` 文件编译成 `pdf` 文件.

编译命令有很多可选参数, 如用户能够从 `pdf` 文件跳回 `tex` 文件, 便是因为执行编译时添加了参数

```
1 pdflatex -synctex=1 main
```

此外还有很多其他参数, 用户若感兴趣, 可自行阅读手册.

在很多时候, 我个人更倾向于使用 `latexmk` 来编译 `tex` 文档, 如执行

```
1 latexmk -pdf -synctex=1 -interaction=nonstopmode main
```

意味着使用

```
1 pdflatex -synctex=1 -interaction=nonstopmode main
```

来编译文档 `main.tex`, 并在有需要时完成其他步骤²⁴.

²¹ 建议不同的工程放入不同的工作路径

²² 该文件必须包含 `\documentclass`, `\begin{document}` 和 `\end{document}`, 这里不讨论包含子文件的情况, 文件名需为不含空格的英文

²³ 目前常用的编译命令为 `pdflatex` 和 `xelatex`

²⁴ 如使用 `bibtex` 或 `biblatex` 处理参考文献时需要多次编译, 详情见相关文档

2 Ubuntu 18.04 系统

2.1 安装 T_EX Live

这里只阐述如何用镜像安装. 为使用户顺利使用 T_EX Live 2020, 建议用户首先卸载从源内安装的 T_EX Live 的相关包, 具体方法见 2.2 节.

下载 [iso 镜像文件](#) 的方法如前所述, 可选择国内源以加快下载速度, 见 1.1 节. 下载完毕后, 打开 Terminal 窗口, 执行以下命令

```
1 cd ~/Downloads
2 md5sum texlive2020.iso
```

若显示

```
1 a31c500c7830d901363c5fa0e65cc18d texlive2020.iso
```

则镜像文件下载正确.

接下来, 使用如下代码加载光盘镜像至 ~/Downloads/texlive 文件夹

```
1 mkdir ~/Downloads/texlive
2 sudo mount ./texlive2020.iso ~/Downloads/texlive
```

接下来执行

```
1 sudo ~/Downloads/texlive/install-tl
```

进行安装. 在屏幕上应该能见到以下内容

```
1 =====> TeX Live installation procedure <=====
2
3 =====> Letters/digits in <angle brackets> indicate <=====
4 =====> menu items for actions or customizations <=====
5
6 Detected platform: GNU/Linux on x86_64
7
8 <B> set binary platforms: 1 out of 6
9
10 <S> set installation scheme: scheme-full
11
12 <C> set installation collections:
13    40 collections out of 41, disk space required: 6516 MB
```

```

14
15 <D> set directories:
16   TEXDIR (the main TeX directory):
17     /usr/local/texlive/2020
18   TEXMFLOCAL (directory for site-wide local files):
19     /usr/local/texlive/texmf-local
20   TEXMFSYSVAR (directory for variable and automatically generated data):
21     /usr/local/texlive/2020/texmf-var
22   TEXMFSYSCONFIG (directory for local config):
23     /usr/local/texlive/2020/texmf-config
24   TEXMFVAR (personal directory for variable and automatically generated data):
25     ~/.texlive2020/texmf-var
26   TEXMFCONFIG (personal directory for local config):
27     ~/.texlive2020/texmf-config
28   TEXMFHOME (directory for user-specific files):
29     ~/texmf
30
31 <O> options:
32   [ ] use letter size instead of A4 by default
33   [X] allow execution of restricted list of programs via \write18
34   [X] create all format files
35   [X] install macro/font doc tree
36   [X] install macro/font source tree
37   [ ] create symlinks to standard directories
38   [X] after install, set CTAN as source for package updates
39
40 <V> set up for portable installation
41
42 Actions:
43   <I> start installation to hard disk
44   <P> save installation profile to 'texlive.profile' and exit
45   <H> help
46   <Q> quit
47
48 Enter command:

```

这里建议用户点击键盘 **I** 使用默认配置安装。如果用户对于 Ubuntu 系统比较了解, 可以根据提示, 更改安装设置。安装完毕后, 将加载的光盘镜像弹出

```
1 sudo umount ~/Downloads/texlive
```

默认安装完成后, 用户需要设置环境变量。继续在 Terminal 中执行

```
1 sudo gedit ~/.bashrc
```

在打开的文件末尾添加

```

1 # Add TeX Live to the PATH, MANPATH, INFOPATH
2 export PATH=/usr/local/texlive/2020/bin/x86_64-linux:$PATH

```

```
3 export MANPATH=/usr/local/texlive/2020/texmf-dist/doc/man:$MANPATH
4 export INFOPATH=/usr/local/texlive/2020/texmf-dist/doc/info:$INFOPATH
```

并保存退出. 这时再打开 Terminal 执行

```
1 tex -v
```

若显示

```
1 TeX 3.14159265 (TeX Live 2020)
2 kpathsea version 6.3.2
3 Copyright 2020 D.E. Knuth.
4 There is NO warranty. Redistribution of this software is
5 covered by the terms of both the TeX copyright and
6 the Lesser GNU General Public License.
7 For more information about these matters, see the file
8 named COPYING and the TeX source.
9 Primary author of TeX: D.E. Knuth.
```

即为安装成功.

接下来处理字体. 首先将配置文件复制到系统, 在 Terminal 中执行

```
1 sudo cp /usr/local/texlive/2020/texmf-var/fonts/conf/texlive-fontconfig.conf /etc/fonts/conf.d/09-
  texlive.conf
```

接下来刷新字体缓存, 继续在 Terminal 执行

```
1 sudo fc-cache -fsv
```

这样一来, TeX Live 中的字体才能够被正确调用. 更多内容请参考 [TeX live 指南](#).

2.2 卸载 TeX Live

如果要卸载从源内安装的 TeX Live, 个人比较推荐使用 synaptic package manager. terminal 中执行

```
1 sudo apt-get install synaptic
```

即可安装. 安装后打开, 搜索 texlive 即可看到与之相关的包, 右键标记以删除即可. 当然, 用户也可以直接使用

```
1 dpkg -l | grep 'TeX Live'
```

找到相应的包名 <package name>, 之后使用类似

```
1 sudo apt-get remove --purge <package name>
```

的命令删除各个包. 删除时系统会提示使用

```
1 sudo apt-get autoremove <package name>
```

来删除相应的依赖包.

如果是从光盘镜像安装, 只需直接删除文件夹即可. 可先在 terminal 中执行

```
1 kpsewhich -var-value TEXMFROOT
```

来查询安装路径, 进而通过 `sudo rm -rf` 命令进行删除. 默认安装的用户直接运行

```
1 sudo rm -rf /usr/local/texlive/2020
2 rm -rf ~/.texlive2020
```

当然删除后还要清理掉环境变量.

2.3 跨版本升级 T_EX Live

在 tug.org 网站上提供了相应的方法. 但网站也声明: 默认情况下, 请通过执行新安装来获取新版本的 T_EX Live¹.

2.4 升级宏包

首次升级前, 在 terminal 中执行

```
1 sudo visudo
```

将

```
1 /usr/local/texlive/2020/bin/x86_64-linux:
```

添加在 `secure_path` 中. 然后依次 `[Ctrl]+[X]`, `[Y]`, `[Enter]` 保存退出.

接下来在 terminal 中执行

```
1 sudo tlmgr option repository ctan
```

更改源, `ctan` 也可以改成其他地址, 详情见 1.4 节. 接下来, 用户执行命令

```
1 sudo tlmgr update --list
```

可查看目前源上可升级的宏包都有哪些. 高级用户可以根据自己的需求选择升级特定宏包. 建议初级用户直接升级全部宏包, 只需执行

```
1 sudo tlmgr update --self --all
```

同时升级 `tlmgr` 本身和全部宏包.

2.5 安装宏包

Ubuntu 18.04 下安装宏包的要求与 Windows 10 下没有多少区别, 只需注意权限, 例如

```
1 sudo tlmgr install mcmthesis
```

即安装了 `mcmthesis`.

¹原文是: By default, please get the new TL by doing a new installation instead of proceeding here

2.6 编译文件

首先, 用户需要在工作路径建立一个 `tex` 文件. 在 `Terminal` 中执行

```
1 mkdir ~/Documents/work_latex
2 cd ~/Documents/work_latex/
3 gedit main.tex
```

在打开的文件输入一个最小示例

```
1 \documentclass{article}
2 \begin{document}
3   Hello \LaTeX{} World!
4 \end{document}
```

保存并退出. 接下来执行

```
1 pdflatex main
```

等待系统完成编译过程. 待编译完成后, 我们即可看到在 `~/Documents/work_latex` 中出现了 `main.pdf` 文件和其他同名的辅助文件 `main.aux` 与 `main.log`. 执行

```
1 evince main.pdf
```

即可打开 pdf 文件.

对于中文文档, 可以在 `gedit` 中编写以下最小示例²

```
1 \documentclass[UTF8]{ctexart}
2 \begin{document}
3   你好 \LaTeX{} 世界!
4 \end{document}
```

保存并退出. 接下来执行

```
1 xelatex main
```

等待系统完成编译过程. `xelatex` 可以使用系统内安装的字体, 安装字体的方法可参考附录 A.

编译命令可添加参数, 这里与 1.6 节中的情形一致.

2.6.1 无法使用 `xelatex` 命令

有些用户反映安装完毕后无法使用 `xelatex` 命令. 这里或许是因为 `libfontconfig` 缺失, 用户可在命令行中执行

```
1 sudo apt-get install libfontconfig1
```

进行安装.

为提升安装速度, 可考虑更换 Ubuntu 源, 例如更换至清华大学. 在 `Terminal` 中执行

```
1 sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
```

² `gedit` 默认使用 UTF-8 编码

备份 `sources.list` 文件. 接下来执行

```
1 sudo gedit /etc/apt/sources.list
```

将文件替换为以下内容³

```
1 # 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
2 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
3 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted universe multiverse
4 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe
   multiverse
5 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main restricted universe
   multiverse
6 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe
   multiverse
7 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main restricted universe
   multiverse
8 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe
   multiverse
9 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main restricted universe
   multiverse
10
11 # 预发布软件源，不建议启用
12 # deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main restricted universe
   multiverse
13 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-proposed main restricted universe
   multiverse
```

然后执行

```
1 sudo apt-get update && sudo apt-get upgrade
```

至此换源完毕.

³本段文字可至[清华大学镜像网站](#)获取

3 macOS

3.1 安装 Homebrew

强烈建议用户使用 [Homebrew](#)。Homebrew 是一个包管理, 类似 Ubuntu 上面的 `apt-get`。安装教程可以在其网站找到, 这里简单列出来:

```
1 /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

将以上命令在**终端**¹执行。脚本会在执行前暂停, 并说明它将做什么。依据屏幕指令执行即可。

中国大陆用户可以更改镜像以提高访问速度。以中国科学技术大学镜像源为例:

```
1 cd "$(brew --repo)/Library/Taps/homebrew/homebrew-core"
2 git remote set-url origin https://mirrors.ustc.edu.cn/homebrew-core.git
3 cd "$(brew --repo)/Library/Taps/homebrew/homebrew-cask
4 git remote set-url origin https://mirrors.ustc.edu.cn/homebrew-cask.git
5 echo 'export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles' >> ~/.
  bash_profile
6 source ~/.bash_profile
```

如果是 `zsh` 用户, 最后两行请替换为

```
1 echo 'export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles' >> ~/.zshrc
2 source ~/.zshrc
```

3.1.1 Xcode

根据 [Homebrew 网站](#)的提示², 推荐用户在安装 Homebrew 前先安装 Xcode。在**终端**执行以下命令即可:

```
1 xcode-select --install
```

3.2 安装 MacTeX

MacTeX 是 TeX Live 在 macOS 下的再打包版本, 额外加入了一些辅助程序。如果已经安装了 Homebrew, 只需在**终端**执行以下命令即可完成安装:

¹打开方法为:  + , 输入 `terminal` 并打开 **终端** 应用

²原文是 Homebrew supports and recommends the latest Xcode and/or Command Line Tools available for your platform

```
1 brew cask install mactex
```

如有输入密码等提示, 请根据屏幕指示操作. 至于环境变量等繁琐细节, Homebrew 会自动进行处理, 无须用户干预.

完整的 MacTeX 会比较大. 如果磁盘空间实在紧张, 也可以考虑安装 BasicTeX:

```
1 brew cask install basictex
```

安装完成后 BasicTeX 依然会缺很多包, 手动安装会比较麻烦, 所以不推荐没有经验的用户尝试.

3.3 卸载 MacTeX

如果用户借助 Homebrew cask 安装了 MacTeX, 那么卸载工作可能会稍显麻烦. 这里引用 [Github](#) 上的讨论. 用户可以根据这里的内容卸载 MacTeX.

3.4 跨版本升级 MacTeX

跨版本升级 (MacTeX 的版本与 TeX Live 保持一致), 可在终端借助 Homebrew 完成:

```
1 brew update
2 brew cask upgrade mactex
```

3.5 升级宏包

升级宏包依旧可以使用 tlmgr. 使用方法与 2.4 节类似, 这里不再重复. 一般来说, 也需要使用 sudo 获取管理员权限后才能完成安装.

3.6 安装宏包

安装 CTAN 中的宏包方法与 2.5 节一致.

3.7 编译文件

假设已经用 TextEdit.app 或其他文本编辑器编写以下示例 main.tex³, 内容为

```
1 \documentclass{article}
2 \begin{document}
3   Hello \LaTeX{} World!
4 \end{document}
```

接下来在终端中执行

³注意建立最小示例前先确定工作路径

```
1 pdflatex main
```

等待系统完成编译过程. 待编译完成后, 可看到在工作路径中生成了 `main.pdf` 文件和其他同名的辅助文件 `main.aux` 与 `main.log`. 可以打开 `main.pdf` 查看内容.

对于中文文档, 可以编写以下最小示例⁴

```
1 \documentclass[UTF8]{ctexart}
2 \begin{document}
3   你好 \LaTeX{} 世界!
4 \end{document}
```

保存并退出. 接下来在**终端**中进入工作路径, 执行

```
1 xelatex main
```

等待系统完成编译过程. `xelatex` 可调用系统字体, 为系统安装字体的方法请参考 [How to install and remove fonts on your Mac](#). 安装完成后, 刷新字体缓存. 注意到此时 `xelatex` 只能通过文件名来调用发行版预装的字体, 解决此问题的方法请参考[慕子的文章](#).

编译命令的相关参数, 这里不再赘述.

⁴注意使用 UTF-8 编码

4 Windows Subsystem for Linux

目前微软推出了 Windows Subsystem for Linux (WSL) 供开发人员使用. 在这里简要介绍如何在 WSL 中安装 TeX Live. 我选择的是微软商店中的 Ubuntu. 此篇安装教程仅供参考. 这里称 WSL 中使用的命令行为 `bash`. **不建议**对操作系统了解不多的用户阅读本章内容.

4.1 安装 TeX Live

在主系统¹中下载 **iso 镜像文件**, 可选择国内源以加快下载速度. 下载完毕后, 在 `cmd` 中验证文件是否正常. 换源下载及文件验证方法见 1.1 节.

在正式安装前, 用户需要在 `bash` 中执行

```
1 sudo apt-get install libfontconfig1
2 sudo apt-get install ttf-mscorefonts-installer
3 sudo apt-get install fontconfig
```

这些命令是为了处理日后使用中可能出现的字体问题. 为避免安装速度过慢, 可仿照 2.6.1 节更改文件 `sources.list`. 注意 WSL 中无法启动 `gedit`, 因此需要将其替换为 `vim`, 即

```
1 sudo vim /etc/apt/sources.list
```

不熟悉 `vim` 的用户, 可参考 4.1.1 节.

接下来, 在主系统中将镜像挂载, 例如挂载到 `X:\`, 而后进入 `bash` 并执行如下命令

```
1 sudo mkdir /mnt/x
2 sudo mount -t drvfs X: /mnt/x
```

如此便可在 `bash` 中找到挂载的光盘镜像. 之后执行

```
1 sudo /mnt/x/install-tl
```

进行安装. 在屏幕上应该能见到以下内容

```
1 =====> TeX Live installation procedure <=====
2
3 =====> Letters/digits in <angle brackets> indicate <=====
4 =====> menu items for actions or customizations <=====
5
6 Detected platform: GNU/Linux on x86_64
```

¹在 Windows 10 中直接进行的操作即为在主系统中的操作

```

7
8 <B> set binary platforms: 1 out of 6
9
10 <S> set installation scheme: scheme-full
11
12 <C> set installation collections:
13     40 collections out of 41, disk space required: 6516 MB
14
15 <D> set directories:
16     TEXDIR (the main TeX directory):
17     /usr/local/texlive/2020
18     TEXMFLOCAL (directory for site-wide local files):
19     /usr/local/texlive/texmf-local
20     TEXMFSYSVAR (directory for variable and automatically generated data):
21     /usr/local/texlive/2020/texmf-var
22     TEXMFSYSCONFIG (directory for local config):
23     /usr/local/texlive/2020/texmf-config
24     TEXMFVAR (personal directory for variable and automatically generated data):
25     ~/.texlive2020/texmf-var
26     TEXMFCONFIG (personal directory for local config):
27     ~/.texlive2020/texmf-config
28     TEXMFHOME (directory for user-specific files):
29     ~/.texmf
30
31 <O> options:
32     [ ] use letter size instead of A4 by default
33     [X] allow execution of restricted list of programs via \write18
34     [X] create all format files
35     [X] install macro/font doc tree
36     [X] install macro/font source tree
37     [ ] create symlinks to standard directories
38     [X] after install, set CTAN as source for package updates
39
40 <V> set up for portable installation
41
42 Actions:
43 <I> start installation to hard disk
44 <P> save installation profile to 'texlive.profile' and exit
45 <H> help
46 <Q> quit
47
48 Enter command:

```

用户直接点击 ☐ 使用默认配置安装。如果用户对于 WSL 比较了解, 可以根据提示, 更改安装设置。安装完毕后, 需继续在 bash 中执行

```
1 sudo umount /mnt/x
```

```
2 sudo rmdir /mnt/x
```

弹出已加载的光盘镜像。

默认安装完成后, 用户需要设置环境变量. 继续在 `bash` 中执行

```
1 sudo vim ~/.bashrc
```

在打开的文件末尾添加

```
1 # Add TeX Live to the PATH, MANPATH, INFOPATH
2 export PATH=/usr/local/texlive/2020/bin/x86_64-linux:$PATH
3 export MANPATH=/usr/local/texlive/2020/texmf-dist/doc/man:$MANPATH
4 export INFOPATH=/usr/local/texlive/2020/texmf-dist/doc/info:$INFOPATH
```

并保存退出. 同样, 不熟悉 `vim` 的用户可参考 4.1.1 节. 退出 WSL 再进入, 执行

```
1 tex -v
```

将显示

```
1 TeX 3.14159265 (TeX Live 2020)
2 kpathsea version 6.3.2
3 Copyright 2020 D.E. Knuth.
4 There is NO warranty. Redistribution of this software is
5 covered by the terms of both the TeX copyright and
6 the Lesser GNU General Public License.
7 For more information about these matters, see the file
8 named COPYING and the TeX source.
9 Primary author of TeX: D.E. Knuth.
```

即为安装成功.

接下来仿照 Ubuntu 18.04 处理字体. 首先将配置文件复制到系统, 在 Terminal 执行

```
1 sudo cp /usr/local/texlive/2020/texmf-var/fonts/conf/texlive-fontconfig.conf /etc/fonts/conf.d/09-
  texlive.conf
```

接下来刷新字体缓存, 继续在 Terminal 执行

```
1 sudo fc-cache -fsv
```

4.1.1 不懂 `vim` 的处理方法

`vim` 的使用方法与目前很多流行的编辑器不同, 因此这里给出不用 `vim` 的处理方法.

首先更改 `sources.list`. 假设已经将 `sources.list` 备份, 而后在主系统中通过 `everything` 搜索文件 `sources.list`, 找到路径在 `...etcapt` 的文件, 省略号表示 `etc` 前那段很长的路径. 使用常用的编辑器打开 `sources.list` 并进行更改即可, 更改内容同 2.6.1 节.

接下来更改 `.bashrc`. 同样也是在主系统中通过 `everything` 搜索文件 `.bashrc`, 找到路径在 `...homeusername` 的文件, 使用常用的编辑器打开并进行更改.

4.2 卸载 T_EX Live

直接删除文件夹即可. 先在 `bash` 中执行

```
1 kpsewhich -var-value TEXMFROOT
```

来查询安装路径, 进而通过 `sudo rm -rf` 进行删除. 默认安装的用户直接运行

```
1 sudo rm -rf /usr/local/texlive/2020
2 rm -rf ~/.texlive2020
```

当然删除后还要清理掉环境变量.

4.3 跨版本升级 T_EX Live

WSL 中的方法同 2.3 节.

4.4 升级宏包

WSL 中的方法同 2.4 节.

4.5 安装宏包

WSL 中的方法同 2.5 节.

4.6 编译文件

在这里, 我们假设已经有了一个最小示例 `main.tex`², 内容为

```
1 \documentclass{article}
2 \begin{document}
3   Hello \LaTeX{} World!
4 \end{document}
```

接下来在 `bash` 中进入工作路径, 执行

```
1 pdflatex main
```

等待系统完成编译过程. 当然也可以在 `cmd` 中进入工作路径并执行

```
1 bash -i -c "pdflatex main"
```

实现编译. 待编译完成后, 可看到在工作路径中生成了 `main.pdf` 文件和其他同名的辅助文件 `main.aux` 与 `main.log`. 在主系统中可以打开 `main.pdf` 查看内容.

对于中文文档, 可以在主系统编写以下最小示例³

²可以在主系统中建立文件, 也可通过 `bash` 调用 `vim` 建立, 建立文件前需要确定工作路径

³注意使用 UTF-8 编码


```

1 \documentclass[UTF8]{ctexart}
2 \begin{document}
3   你好 \LaTeX{} 世界!
4 \end{document}

```

保存并退出. 接下来在 `bash` 中进入工作路径, 执行

```
1 xelatex main
```

等待系统完成编译过程. 同样可以在 `cmd` 中进入工作路径并执行

```
1 bash -i -c "xelatex main"
```

实现编译. `xelatex` 同样可以使用 WSL 内安装的字体⁴, 在 WSL 中安装字体的方法见附录 A.

编译命令可添加参数, 这里与 1.6 节中的情形一致, 不再赘述.

4.6.1 L^AT_EX 2_ε 版本不匹配导致 `xelatex` 失败

在 WSL 中将 T_EX Live 安装至默认路径时, 使用 `xelatex` 编译文档, 系统有时会提示目前 L^AT_EX 2_ε 的版本低于某宏包的要求版本. 这种情况是由于 `fmt` 文件未能更新所导致. 此时可考虑在命令行运行

```
1 fmtutil-user --byfmt xelatex
```

待结束后再用 `xelatex` 编译文档. 更多内容参考 [github 上的讨论](#).

另外也可考虑重装 `xetex`. 在命令行执行

```
1 sudo tlmgr install --reinstall xetex
```

4.7 尚未圆满解决的问题

将 T_EX Live 安装至 WSL 仍有悬而未决的问题. 由于 `bash` 默认是无窗口化的纯命令行, 因此用户无法直接通过命令 `texdoc` 打开相应的手册. 这里给出几种方法.

explorer 浏览

先找到手册所在路径, 例如寻找 `lshort-zh-cn`

```
1 texdoc -l lshort-zh-cn
```

系统会显示

```

1 1 /usr/local/texlive/2020/texmf-dist/doc/latex/lshort-chinese/lshort-zh-cn.pdf
2 = [zh] The document itself
3 Enter number of file to view, RET to view 1, anything else to skip:

```

这时按其他键 (如) 退出. 然后分别执行

⁴注意主系统中的字体默认不能被调用, 如果打算使用主系统中的字体, 见 [github 上的讨论](#)

```
1 cd /usr/local/texlive/2020/texmf-dist/doc/latex/lshort-chinese/  
2 explorer.exe .
```

主系统的资源管理器便会打开. 这时用户便可以在主系统中使用 pdf 阅读器打开手册进行阅读.

everything 搜索

使用 [everything](#). 通过 `texdoc -l` 找到相应文件名, 而后直接搜索. 具体方法这里不再赘述.

借助其他工具

目前有很多开源的 WSL 辅助程序, 相关讨论见本手册的 [issue](#).

5 使用编辑器

5.1 Windows 10 系统

在实际操作中, 用户会发觉使用记事本编写 `tex` 文件十分不便, 因此很多用户都将其他编辑器作为自己的首选. 这里向大家介绍随 \TeX Live 一同发行的轻量级编辑器 \TeX works 和功能更加丰富的 \TeX studio (github 上有[源码](#)和[可执行文件](#)). 其他编辑器无法一一介绍, 还请自己在网上寻求更多帮助.

5.1.1 \TeX works

\TeX works 是一款轻量级的 \LaTeX 编辑器, 个人认为非常适合入门级用户使用¹.

和大多数国产软件不同, \TeX works 不会自动在桌面生成快捷方式, 新人往往不知道如何打开它. 实际上 \TeX works 在 \TeX Live 安装路径的 `\bin\win32` 中, 可在 `cmd` 中执行 `texworks` 打开, 也可直接在 Windows 搜索栏里搜 \TeX works 打开.

刚开始使用 \TeX works 的用户不必过多配置, 在默认状态下打开软件、编写代码、保存文件后, 用户可使用 \TeX works 的“排版”按钮进行文档编译而不必回到命令行². 按钮旁边是编译命令菜单, 用户可根据需要自行选择.

下面介绍几个常用的功能.

显示行号

打开 **编辑** **》** **首选项** **》** **编辑器**, 选择 **行号** 即可.

调整字体

\TeX works 有时会擅自更改编辑器所使用的字体, 很多字体“长相”诡异, 让用户误以为是乱码. 用户可在 **编辑** **》** **首选项** **》** **编辑器默认配置** 和 **格式** **》** **字体** 处进行更改. Windows 10 中字体列表见[这里](#).

自动补全

自动补全主要是 **tab** 键的功能, 使用前需确定 **首选项** **》** **编辑器** 中是否勾选了 **启动自动补全**. 有关自动补全的更多内容用户可参考 [\$\text{\TeX}\$ works 的自动补全功能](#).

¹很多用户在编写 `tex` 文件时喜欢借助其他工具, 这不是个好习惯, 这里鼓励用户多手动敲代码

²实际上编辑器的按钮也是调用命令行来编译文档, \TeX works 默认排版快捷键是 `ctrl + T`

使用模板

TeXworks 支持从模板新建文档, 在 **文件** 菜单中可见.

拼写检查

默认情况, TeXworks 没有搭载拼写检查字典, 需要用户自己配置. 首先, 访问 [openoffice](#) 下载词典. 接下来将词典安装到 `<resources>\dictionaries`³. 安装完成后, 在 **编辑** **拼写** 中选择即可.

魔法注释

魔法注释可以直接规定主文档、文件编码和编译命令等. 例如用户写学位论文时, 往往有一个主文档和若干子文档. 在子文档的开头写上魔法注释

```
1 % !TeX root = ../mainfile.tex
```

即可告诉系统上一级目录中的 `main.tex` 是主文档⁴.

```
1 % !TeX encoding = UTF8
```

表示该文档使用 “UTF-8” 编码⁵.

```
1 % !TeX program = xelatex
```

表示该文档使用 `xelatex` 编译命令进行编译.

```
1 % !TeX spellcheck = <language_code>
```

表示该文档使用 `<language_code>` 进行拼写检查, 它的具体名称请用户到 **编辑** **拼写** 中查看.

转编码

实际上转编码是很多编辑器都具备的功能. 在 TeXworks 右下角有三个按钮, 左边按钮控制换行符类型, 中间按钮控制文档编码, 右边按钮控制行跳转. 目前大部分中文用户主要面临的是 “UTF-8” 和 “GBK” 之间的转换. 如果文档是 “GBK” 编码, 使用 TeXworks 打开文件后, 文档会出现乱码. 这时, 点中间按钮, 选择编码类型 “GBK”, 再点击按钮, 选择 **使用所选编码重载文档**, 若文档中乱码消失, 则再次点击按钮, 选择 “UTF-8”, 最后保存文档, 完成转码工作.

除以上内容外, TeXworks 还支持正则表达式、自定义快捷键等. 这些内容都写在 TeXworks 自带的手册中. 手册不长, 用户可以通读一遍以了解更多内容. 很多功能可以通过**菜单栏**找到, 因此建议用户稍微了解一下它的菜单栏.

5.1.2 TeXstudio

相较于 TeXworks, TeXstudio 功能更丰富, 用法更多. 在使用 TeXstudio 前, 用户一定要查询 `system32` 是否在**环境变量**中. 具体方法在前面已有说明, 此处不再赘述. 以下有关 TeXstudio 的用法介绍, 还请用户提前提前在 **Options** **configure TeXstudio** 中勾选 **Show Advanced Options**.

³`<resources>` 的具体位置可以在 **帮助** **TeXworks 配置与资源** 中找到

⁴主文档和子文档的更多内容请参考 `\include` 和 `\input` 的用法

⁵目前建议用户多用 “UTF-8” 编码, 尤其是中文文档

更改默认编译命令

在 **Options** **»** **configure TeXstudio** **»** **build** **»** **default compiler** 中选择默认编译命令。

另外用 `latexmk` 和 `bibtex` 做编译时, 需要在 **Options** **»** **Configure TeXstudio** **»** **build** **»** **Build Options** 中取消 **Check and update bibliography before compiling**.

显示行号

在 **Editor** **»** **Show Line Numbers** 选择行号如何显示。

数学符号表

在 TeXstudio 左下角有两个按钮: **Side Panel** 和 **Messages / Log file**. 打开 **Side Panel**, 点击软件左侧边栏第 3 个图标 **Symbols**.

插入、改写调整

与很多其他编辑器一样, TeXstudio 也可以通过键盘 **insert** 改变插入、改写状态. 分别处于插入、改写状态时, 编辑器光标会呈现不同的样子.

调整缩进

在 **Options** **»** **Configure TeXstudio** **»** **Editor** **»** **Indentation Mode** 处可以选择是否保持缩进. 如果需要让缩进完全变为空格, 勾选 **Replace Indentation Tab by Spaces**.

在 **Adv. Editor** **»** **Appearance** 中可以更改 **Tab Widths**.

块选择模式

同时按 **Ctrl** + **Alt** 便可通过鼠标左键进行块选择.

调整图标大小

有些用户屏幕分辨率高, 因此软件上图标会显得很很小. 在 **Options** **»** **Configure TeXstudio** **»** **Gui Scaling** 处可以调整三处的显示比例.

自动补全和未识别代码

TeXstudio 自动补全功能会先通过用户引用的宏包来判断代码是否需要自动补全, 同时它也可以以此判定用户输入的命令是否正确. 若系统判断命令无法识别, 则该命令会出现红色背景.

TeXstudio 能够识别的命令全部被写入了 **cwl 文件** 中. 用户可以根据自己的需要更改 **cwl 文件**, 然后将其放入 **<settings directory>\completion\user 文件夹**⁶.

自动补全后有可能生成文本框, 使用 **Ctrl** + **→** 可以跳转至下一个文本框, **Ctrl** + **←** 则可以跳转至上一个.

⁶ 点击 **Help** **»** **Check LaTeX Installation**, 在生成的 **System Report.txt** 中找到 **texstudio.ini** 文件, 其所在文件夹即为 **<settings directory>**

拼写检查

在 **Options** **Configure TeXstudio** **Language Checking** **Default Language** 处选择 **en_US** 即可默认进行英文拼写检查。

魔法注释

除了像 TeXworks 中使用魔法注释外, TeXstudio 还有很多其他的魔法注释, 例如

```
1 % !TeX TKS-program:compile = tks:///latexmk/{ }[-xelatex -synctex=1 -interaction=nonstopmode ?me"]
```

表示使用 **latexmk** 编译命令, **{ }** 表示无视编辑器赋予该命令的一切参数, 而 **[]** 表示添加其中的参数进行编译, 本例中添加 **-xelatex -synctex=1 -interaction=nonstopmode** 作为参数. 如果用户不清楚魔法注释都可以写哪些类型, 可以点击 **LaTeX** **Add magic comments** 查看五种魔法注释类型. 更多内容需要阅读软件手册.

渲染方式

默认的渲染方式对中文括号的支持不够好. 用户可以在 **Options** **Configure TeXstudio** **Adv. Editor** **Hacks/Workarounds** 中取消 **Try to automatically choose best display options**, 选择 **Disable cache of character width** 和 **Disable fixed pitch mdoe**. 这样能够比较好地渲染中文括号.

转编码

TeXstudio 右下角有一处显示文档编码 (Encoding). 依旧以 “UTF-8” 和 “GBK” 之间的转换为例. 如果文档是 “GBK” 编码, 使用 TeXstudio 打开文件后, 点击 **Encoding** 处, 选择 **More Encodings**. 打开窗口后, 选择 “GBK / ...”, 点击 **Reload With**. 若这时文档没有乱码, 再点击 **Encoding** **More Encodings**, 选择 “UTF-8”, 点击 **Change To**, 保存文件, 完成转码. 在 **Edit** **Setup Encoding ...** 处同样可以打开编码窗口.

自定义命令并生成按钮

TeXstudio 允许用户自定义命令, 并将命令做成按钮放置于面板上. 如在 **Options** **Configure TeXstudio** **Build** **User commands** 中, 填写名称 **User1:Build-xe-view** 和对应的功能

```
1 latexmk -pdfxe -silent -synctex=1 -interaction=nonstopmode ?me" | tks:///view-pdf-internal --
   embedded
```

点击 **OK**. 接下来我们在 **Tools** **User** 中即可看到自定义的命令和相应快捷键.

接下来设置按钮. 打开 **Options** **Configure TeXstudio** **Toolbars**, 在两个下拉菜单中分别选择 **Tools** 和 **All menus**, 在右边找到 **Tools** **User** **Build-xe-view** 并将其添加至左端, 点击 **OK**. 这时, 在面板中将添加新的按钮.

生成宏指令

TeXstudio 允许用户生成宏指令. 在 **Macros** **Edit Macros** 中, 用户可以根据自己的需要, 给出宏指令的名称、快捷键、内容等等. 例如我们将 **latexmk-pdf** 填写在 **name** 中, 将

```
1 % !TeX Txs-program:compile = txs:///latexmk/{ }[-pdf -synctex=1 -interaction=nonstopmode ?me"]
```

填写在 **LaTeX Content** 中, 再将 **Shift+F1** 填写在 **Shortcut**. 设置完毕后, 点击 **OK**. 这时, 我们可以看到 **Macros** **» latexmk-pdf** 出现. 在文档特定位置点击它或者直接使用 **⇧**+**F1**, 即可看到魔法注释出现.

更改颜色方案

有一些用户喜欢深色的颜色方案. 手动修改颜色不方便的话, 我们可以从网上找到一些现成的颜色方案, 例如 [stackexchange 网站](#) 和 [某些用户的博客](#). 更改方案只需要打开 **ini** 文件, 将这些颜色方案复制到 **[formats]** 部分. 在 **Help** **» Check LaTeX Installation** 中有 **ini** 文件的具体位置.

调用外部 PDF 阅读器

T_EXstudio 已经内嵌了 PDF 阅读器供用户预览编译结果, 但一些用户喜欢使用外部 PDF 阅读器. 这里以 **SumatraPDF** 为例, 用户直接安装或下载便携版均可, 假定它在本地的位置为 **<SumatraPDFROOT>**, 而 **T_EXstudio** 在本地的位置是 **<TeXstudioROOT>**. 打开 **Options** **» Configure T_EXstudio** 窗口, 在 **Build** **» User commands** 中添加

```
1 "<SumatraPDFROOT>\SumatraPDF.exe" -reuse-instance -forward-search ?c:rme" @ -inverse-search "<
  TeXstudioROOT>\texstudio.exe %f -line %l" "?m.pdf"
```

将其命名为 **user0:sumatrapdf**. 接下来, 在 **Build** **» Build & View** 中将 **txs:///compile** | **txs:///view** 改为 **txs:///compile** | **txs:///user0**. 最后, 在 **Menus** 中将 **&View** 的命令由 **txs:///view** 改为 **txs:///user0**. 完成以上设置后, 关闭窗口. 这时, 用户使用快捷键 **F5** 和 **F7** 均可打开 **SumatraPDF** 并且实现了正向搜索. 如果有人喜欢在编译时添加参数 **--outdir=temp**, 那么可以将 **user0:sumatrapdf** 改为

```
1 "<SumatraPDFROOT>\SumatraPDF.exe" -reuse-instance -forward-search ?c:rme" @ -inverse-search "<
  TeXstudioROOT>\texstudio.exe %f -line %l" "?r)temp\?m.pdf"
```

如果以上设置没能正确实现正反向搜索, 那么可以将 **user0:sumatrapdf** 改为

```
1 dde:/// "<SumatraPDFROOT>\SumatraPDF.exe":SUMATRA/control/[ForwardSearch("?m.pdf",?c:rme",@,0,0,1)]
```

其他不变. 而后用 **SumatraPDF** 打开生成的 PDF 文件, 在 **Settings** **» Options** **» Set inverse search command line** 中输入

```
1 "<TeXstudioROOT>\texstudio.exe" "%f" -line %l
```

至此完成了逆向搜索, 双击 PDF 文件便可回到 **T_EXstudio** 中对应代码的行首.

5.2 Ubuntu 18.04 系统

简化起见, 这里只介绍如何使用 **T_EXstudio**. 根据官网推荐, 我们安装源内的 **T_EXstudio**. 安装前可参照 [2.6.1 节](#) 的介绍换源以增速.

在 **Terminal** 中执行

```
1 sudo apt install texstudio
```

即可安装 T_EXstudio. 注意安装过程中会产生一些依赖, 它们包括了源内的 T_EX live 包, 如 `tex-common`, `texlive-base`, `texlive-binaries`, `texlive-latex-base` 和 `texlive-latex-recommended`. 用户需要卸载它们.

当然用户也可以使用 PPA 来安装 T_EXstudio. 以此在 Terminal 中执行

```
1 sudo add-apt-repository ppa:sunderme/texstudio
2 sudo apt-get update
3 sudo apt-get install texstudio
```

安装完毕后, 用户可以在 Terminal 中执行

```
1 texstudio main.tex
```

使用 T_EXstudio 编辑文档. 也可直接双击 `main.tex` 文件.

注意, 在双击打开时, 用户需要在 `Options` `Configure TeXstudio` 中点击 `Show Advanced Options`, 接下来在 `Build` `Build Options` `Commands ($PATH)` 中添加路径, 默认安装时为

```
1 /usr/local/texlive/2020/bin/x86_64-linux
```

其他设置可参考前文, 这里不再赘述.

5.3 Windows Subsystem for Linux

使用 WSL 时, 推荐用户在主系统中使用 VS Code 的 Remote WSL 插件来配合 L^AT_EX Workshop 编辑文件. 参考配置文件见附录 A.

若需要使用 T_EXstudio, 则需要将 `bash.exe` 及其绝对路径写入编译命令中. 使用 `everything` 搜索时会发现多个 `bash.exe`, 这里我们选择在路径 `C:\Windows\WinSxS...` 下的 `bash.exe`. 若该路径下仍有多个 `bash.exe`, 则挑选与 `C:\Windows\System32` 同等大小的.

下面给出一个例子. 仿照前述建立用户自定义命令的方法, 在 `Options` `Configure TeXstudio` `Build` `User commands` 中, 填写名称 `User4: bash` 和功能⁷

```
1 "C:\Windows\WinSxS\...\bash.exe" -i -c "latexmk -xelatex -synctex=1 -interaction=nonstopmode -halt
-on-error -file-line-error ?me)"
```

点击 `OK`. 接下来打开 `Options` `Configure TeXstudio` `Toolbars`, 在两个下拉菜单中分别选择 `Tools` 和 `All menus`, 在右边找到 `Tools` `User` `bash` 并将其添加至左端, 点击 `OK`. 这时, 在面板中将添加新的按钮.

遗憾的是, 由于路径表达方式不同, T_EXstudio 无法正常使用正反搜索. 这个问题的一个可选择解决方案是在 WSL 中安装图形界面并使用 X Server, 这样可直接将 T_EXstudio 安装至 WSL, 这里不讨论此方案的优劣.

5.4 macOS

macOS 下的编辑器, 这里介绍两款: T_EXShop 和 T_EXstudio.

⁷ 这里省略了中间路径, 用户根据自己的情形进行补充即可

5.4.1 T_EXShop

MacT_EX 自带 T_EXShop 编辑器 (注意不要与其他自带的程序混淆), 通常在安装完毕后便可立即使用. 通过编辑器打开 `main.tex`, 将排版程序 (工具栏上面左数第二个下拉菜单) 由默认的 `LaTeX` 切换到 `pdflatexmk`, 再点击旁边的排版 `排版` (`Typeset`), 待编译完成后会弹出 PDF 预览界面.

编译中文文档, 通常需要将排版程序切换到 `XeLaTeX`. 注意可能需要多次编译以生成正确的交叉引用等内容.

5.4.2 T_EXstudio

如果希望使用 T_EXstudio, 也可以通过 Homebrew 安装:

```
1 brew cask install texstudio
```

注意因为网络问题可能会下载失败. 这时可能需要从其他途径获得安装包并手工安装, 此处不再赘述. 其他设置与 Windows 系统中大致相仿.

无法显示文字

T_EXstudio 在 macOS 下有可能无法显示文字, 这是其他系统下目前没有发现的问题. 具体解决方法见 [李阿玲在知乎的文章](#).

5.5 多款编辑器对比参考

显然, 能够支持 L^AT_EX 的编辑器并不局限于本手册所介绍的几款. 表 5.1 是 [ddswhu](#) 在 L^AT_EX 小圈子内部对 L^AT_EX 编辑器所进行的用户体验调查的结果, 比较主观, 仅供参考. 其中 Sublime Text 和 Visual Studio Code 为文本编辑器, 它们需要借助插件才能支持 L^AT_EX 编译. 对自己网络非常自信的用户也可以参考 [wikipedia](#) 上的对比结果.

表 5.1: L^AT_EX 编辑器对比

	WinEdt	T _E Xstudio	T _E Xworks	Sublime Text	VS Code
插件依赖				L ^A T _E XTools	L ^A T _E X Workshop
主流系统	Win	全平台	Linux/Win	全平台	全平台
软件类型	商业软件	开源软件	开源软件	商业软件	商业软件
软件价格	219 元	0	0	80 美元	0
授权方式	终身/教育			终身/个人	
代码高亮	★★★★☆☆	★★★★☆☆	★☆☆☆☆	★★★★★☆☆	★★★★★☆☆
颜色主题	★★★★☆☆	★★★★☆☆	★☆☆☆☆	★★★★★☆☆	★★★★★☆☆
自动补全	★★★★☆☆	★★★★☆☆	★☆☆☆☆	★★★★★☆☆	★★★★★☆☆
代码片段	★★★★☆☆	★★★★☆☆	☆☆☆☆☆	★★★★★☆☆	★★★★★☆☆
辅助输入	★★★★★☆☆	★★★★★☆☆	☆☆☆☆☆	★★★★★☆☆	★★★★★☆☆
开发完成	★★★★★☆☆	★★★★★☆☆	★★★★★☆☆	★★★★★☆☆	★★★★★☆☆
推荐指数	★★★★☆☆	★★★★☆☆	★☆☆☆☆	★★★★☆☆	★★★★☆☆

6 Overleaf

在特定场合, 有些用户并不需要也没条件在本地安装发行版, 因此这里额外补充 [Overleaf](#) 的相关内容.

6.1 注册 Overleaf

Overleaf 是全球范围内首屈一指的在线 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 编辑平台. 它为每位用户提供了 Ubuntu 系统下的 $\text{T}_{\text{E}}\text{X}$ Live. 它优秀的协作功能、丰富的模板仓库已吸引全球科研工作者成为它的用户. Overleaf 提供了包括中文在内的多种语言供用户使用. 2020 年 1 月 23 日, 它将后台的 $\text{T}_{\text{E}}\text{X}$ Live 升级为 2019 版本, 同时, Overleaf 还允许用户自主选择项目中的 $\text{T}_{\text{E}}\text{X}$ Live 版本.

目前 Overleaf 允许用户使用中文, 并为用户预先准备了一些中文字体. 这些中文字体可通过 $\text{C}_{\text{T}}\text{E}_{\text{X}}$ 宏集调用. 特别推荐用户使用思源宋体和思源黑体这两种开源中文字体.

遗憾的是, 国内网络环境会对 Overleaf 所使用的 reCaptcha 造成影响. 这也使得很多用户在直接注册 Overleaf 时就遇到了问题.

目前, 比较好的替代方案是借助 [ORCID](#) 来进行注册. 目前使用国内网络访问 ORCID 还比较流畅. 用户, 尤其是科研工作者, 可以先注册一个 ORCID 账号. 未来投稿时, 可将 ORCID 账号与自己的期刊网站账号进行绑定. 同时, 用户也可逐步将自己所发表论文列在 ORCID 网站以便管理.

6.2 使用 Overleaf

用户通过 ORCID 注册 Overleaf 后便可进入自己的项目列表页面进行使用.

新建项目是用户首先使用的功能. Overleaf 提供了多种渠道为用户新建项目: 可以通过 Overleaf 中的模板, 也可以通过上传本地的 zip 压缩文件包.

新建项目后, 用户便可进入编辑界面. 在编辑界面用户需要先在左上角 **Menu** 中选择合适的编译命令. 由于默认字体和文件编码等原因, 强烈建议用户在处理中文文档时使用 XeLaTeX . 编写文档后, 用户可通过鼠标点击按钮进行编译, 也可使用快捷键 **ctrl** + **enter**.

用户编写的文件会保存在网站. 编写完成后, 用户只需点击 **Menu** 旁的箭头回到项目列表. 这时可以看到新增项目右侧有四个图标, 它们分别是 **Copy**、**Download**、**Archive** 和 **Trash**. 用户可根据自己的需求点击合适的图标.

6.3 升级项目

Overleaf 将后台 \TeX Live 升级为 2019 版本后, 用户新建项目默认使用 \TeX Live 2019, 而老项目还是使用 \TeX Live 的老版本. 如果用户打算使用 \TeX Live 2019, 只需将项目复制一个新的副本, 即点击 Copy. 这样创建的副本便可以使用 \TeX Live 2019. 如果用户是从 Github 上面导入的项目, 建议再次导入.

6.4 学习与帮助

Overleaf 的模板和文档对全网公开, 用户可以自行学习. 另外 Overleaf 有着专业的技术支援团队, 用户可发送邮件至 support@overleaf.com 咨询使用过程中遇到的问题, 在邮件中请注意文明用语. 部分问题会超出免费服务的范畴, 用户需谨记这点.

附录 A 有关 WSL 的一点补充内容

有用户指出, WSL (本质上是 Ubuntu) 中安装字体比较麻烦. 这里引用 [Ubuntu 系统字体命令和字体的安装](#) 一文. 大致步骤如下.

首先获取需要安装的字体文件, 假设文件保存在 `~/fonts/`. 然后在 `/usr/share/fonts/` 文件夹中创建新的文件夹, 例如 `myfonts`

```
1 cd /usr/share/fonts/  
2 sudo mkdir myfonts
```

接下来将获取的字体文件复制到 `myfonts` 中

```
1 sudo cp ~/fonts/* /usr/share/fonts/myfonts/
```

然后修改字体文件的权限

```
1 sudo chmod -R 755 myfonts
```

最后建立字体缓存

```
1 mkfontscale  
2 mkfontdir  
3 fc-cache -fv
```

在目前的版本中, 本文在 WSL 中进行编译. 目前感受是该方法较之 Windows 10 系统中的编译更快. 其他方面并未进行比较. 由于 `TeXstudio` 在 WSL 中无法正常正反搜索, 因此我使用了 `VS Code` 配合 [Remote WSL](#) 和 [`LaTeX Workshop`](#) 来编写文档. 这里将个人使用的相关配置附上, 其中注释的部分是调用外部 `SumatraPDF` 阅读器的配置, 用户可参考 [SumatraPDF 阅读器该怎么设置——以 `TeXstudio` 和 `VS Code` 为例](#) 来了解更多内容.

```
1 "latex-workshop.latex.tools": [  
2   {  
3     "name": "latexmkpdf",  
4     "command": "latexmk",  
5     "args": [  
6       "-synctex=1",  
7       "-interaction=nonstopmode",  
8       "-halt-on-error",  
9       "-file-line-error",  
10      "-pdf",  
11      "%DOCFILE%"
```

```
12 ]
13 },
14 {
15     "name": "latexmkxe",
16     "command": "latexmk",
17     "args": [
18         "-synctex=1",
19         "-interaction=nonstopmode",
20         "-halt-on-error",
21         "-file-line-error",
22         "-pdfxe",
23         "%DOCFILE%"
24     ]
25 },
26 ],
27 "latex-workshop.latex.recipes": [
28     {
29         "name": "latexmkpdf",
30         "tools": [
31             "latexmkpdf"
32         ]
33     },
34     {
35         "name": "latexmkxe",
36         "tools": [
37             "latexmkxe"
38         ]
39     },
40 ],
41 "latex-workshop.latex.autoBuild.run": "never",
42 "latex-workshop.view.pdf.viewer": "tab"
43 // "latex-workshop.view.pdf.viewer": "external",
44 // "latex-workshop.view.pdf.ref.viewer": "external",
45 // "latex-workshop.view.pdf.external.viewer.command": "<SumatraPDFROOT>/SumatraPDF.exe",
46 // "latex-workshop.view.pdf.external.viewer.args": [
47 //     "-inverse-search",
48 //     "\"<VSCodeROOT>/bin/code.cmd\" -r -g \"%f:%l\"",
49 //     "%PDF%"
50 // ],
51 // "latex-workshop.view.pdf.external.synctex.command": "<SumatraPDFROOT>/SumatraPDF.exe",
52 // "latex-workshop.view.pdf.external.synctex.args": [
53 //     "-forward-search",
54 //     "%TEX%",
55 //     "%LINE%",
56 //     "%PDF%",
57 // ],
```

实际配置时, 用户需要结合自身安装情况更换 `<SumatraPDFROOT>` 和 `<VSCodeROOT>`, 即 SumatraPDF 和 VS Code 各自的安装路径. 希望阅读本文的用户能够尽快上手使用 \LaTeX .

附录 B 离线安装宏包

由于一些原因,有些电脑不能联网,这对升级宏包不利.这里介绍离线升级标准 T_EX Live 的包 (pkg) 的方法,主要是用 `tlmgr install` 的 `--file` 参数来实现.

首先在能够联网的电脑上访问 [archive](#) 页面,下载 `<archivename>.tar.xz` 文件.文件列表很长,加载需花费些时间.注意 `<archivename>` 未必是 `pkg` 上的 `<pkgname>`,例如在 `pkg` 上的 `lshort-zh-cn` 对应着 `archive` 上的 `lshort-chinese.tar.xz` 和 `lshort-chinese.doc.tar.xz`.

这里多解释一点 `<archivename>`.前面已经看到一个 `<pkgname>` 可能会对应多个 `<archivename>`,但基本上 `<archivename>` 的命名规则是 `<xxx>`, `<xxx.source>` 和 `<xxx.doc>`. `<xxx>` 是必装的宏包文件; `<xxx.source>` 是选择安装的源码,如 `dtx` 文件; `<xxx.doc>` 是选择安装的文档,如 `pdf` 文件等.这三者未必同时存在,例如前面提到的 `lshort-chinese`.

另外,如果需要安装的是一个可执行文件,例如 `pdftex` 或 `xetex`,那么会涉及到根据操作系统进行下载的相关判断.如果要升级本地已安装的可执行文件,可以在命令行输入以下语句查询:

Windows 系统

```
1 tlmgr info --only-installed | findstr "pdftex"
```

Ubuntu 和 Mac 系统

```
1 tlmgr info --only-installed | grep 'pdftex'
```

在输出结果中可看到带系统信息的名称.如果是升级本地未安装的可执行文件,那么用户就需要根据经验自行判断.

下载正确的 `<archivename>.tar.xz` 后,可先检查一下该压缩文件中是否包含了 `tlpobj` 文件.之后将压缩文件拷贝到未联网电脑上,在命令行执行

```
1 tlmgr install --file <archivename>.tar.gz
```

系统便可自行安装.

附录 C 版本信息

v2020.5.1

1. 更新 MiKTeX 相关网址
2. 更新至 TeX Live 2020
3. 更新 Windows 10 手动卸载 TeX Live 的内容
4. 调整 TeXstudio 图标大小
5. 更新镜像网址
6. 更改 openoffice 网址
7. 更新 Windows 10 升级宏包描述
8. 更新 Overleaf 的介绍

v2020.4.1

1. 简要介绍 WSL 辅助程序
2. 添加 Overleaf 学习与帮助
3. 添加两处与 WSL 有关的内容
4. 增加几处软件菜单栏的讲解
5. 更清晰查看 Windows 10 环境变量的代码
6. Windows 10 直接以管理员身份打开命令行安装
7. 调整标点

v2020.3.1

1. 摘要中明确提到 CTeX 套装和发行版、编辑器
2. 更改 Windows 10 安装 TeX Live 的描述, 如路径名, 文件扩展名等

3. 将文件名要求写入注释
4. 添加 $\text{T}_{\text{E}}\text{X}$ studio 数学符号表和插入、改写调整
5. 解决 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ 版本不匹配导致 xelatex 失败的问题
6. 添加打开 `cmd` 的方法
7. 改写 $\text{T}_{\text{E}}\text{X}$ studio 显示行号的设置
8. 更新 $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ 相关网址
9. 添加调整 $\text{T}_{\text{E}}\text{X}$ works 字体的内容
10. 添加 Windows 中手动卸载 $\text{T}_{\text{E}}\text{X}$ Live 的方法
11. 添加 Overleaf 中文网址

v2020.2.1

1. 添加版本信息列表
2. 改变 Windows 10 的 $\text{T}_{\text{E}}\text{X}$ studio 的下载网址
3. 添加 macOS 中 xelatex 用字体名调用字体的方法
4. 详述 WSL 中不懂 `vim` 的处理方法和阅读宏包手册的方法
5. 强调路径为不带空格的英文
6. 订正若干拼写错误
7. 给出 `jdk` 环境变量的处理
8. 添加北京交通大学镜像
9. 注明安装需卸载国产安全软件
10. 更正 $\text{T}_{\text{E}}\text{X}$ works 自动补全的教程网址
11. 给出更换 2345 好压的建议
12. 补充 Overleaf 升级内容