

一份简短的关于 L^AT_EX 安装的介绍¹

王然²

2022 年 10 月 1 日

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

²ranwang.osbert@outlook.com

前言

以“嘯行”名义加入 QQ 群 91940767、478023327 和 640633524 后, 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**不存在其他 \LaTeX 发行版 (如 \CTeX 套装)**的前提下, 在 Windows 10、Ubuntu 20.04 和 macOS 系统中安装 \TeX Live (macOS 中介绍 $\text{Mac}\TeX$)、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 $\text{MiK}\TeX$ 的安装, 可以参考 [\$\text{MiK}\TeX\$ 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](#), [Vim](#), 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, 本文还将额外补充两款在线的 \LaTeX 平台, 即 [Overleaf](#) 和 [TeXPage](#) 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 [texlive-zh-cn](#) 和 [lshort-zh-cn](#) 以更全面地了解基础内容.

必须声明的是, 即便你已经通读本手册和上面提到的两本书, 有很大概率还会碰到很多问题. 这时我希望用户可以按照正确方式提问, 例如提供**最小工作示例**, 并在论坛上面留下你的问题和相应解答以方便后来者.

本文大部分内容是个人过去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. 本文另在[码云](#)有备份, 并于 2020 年 7 月提交至 CTAN.

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	5
1.1 安装 T _E X Live	5
1.1.1 下载镜像一直不成功的解决办法	8
1.1.2 需要管理员权限的解决办法	8
1.1.3 系统用中文作为用户名	9
1.2 卸载 T _E X Live	9
1.2.1 使用批处理文件	9
1.2.2 手动卸载	10
1.3 跨版本升级 T _E X Live	10
1.4 升级宏包	10
1.4.1 tlmgr 本身无法成功升级	10
1.4.2 升级到一半停止	10
1.4.3 升级宏包的必要性	11
1.5 安装宏包	11
1.6 调出宏包手册	11
1.7 编译文档	11
1.7.1 L ^A T _E X 2 _ε 版本不匹配导致 xelatex 失败	13
1.7.2 编译出现 l3backend-* 问题	13
2 Ubuntu 20.04 系统	14
2.1 安装 T _E X Live	14
2.1.1 在用户文件夹安装	16
2.1.2 fc-cache 无法执行	17
2.2 卸载 T _E X Live	18
2.2.1 卸载源内的版本	18
2.2.2 手动卸载	18
2.2.3 使用 tlmgr 工具	19
2.3 跨版本升级 T _E X Live	19
2.4 升级宏包	19
2.5 安装宏包	20
2.6 调出宏包手册	20

2.7	编译文件	20
2.7.1	无法使用 <code>xelatex</code> 命令	21
3	macOS	22
3.1	安装 MacTeX	22
3.1.1	直接下载 MacTeX 安装包进行安装	22
3.1.2	借助 Homebrew 以命令行方式安装	22
3.2	卸载 MacTeX	23
3.3	跨版本升级 MacTeX	24
3.4	升级宏包	24
3.5	安装宏包	24
3.6	调出宏包手册	24
3.7	编译文件	24
4	Windows Subsystem for Linux	26
4.1	安装 TeX Live	26
4.1.1	在用户文件夹安装	28
4.2	卸载 TeX Live	28
4.3	跨版本升级 TeX Live	29
4.4	升级宏包	29
4.5	安装宏包	29
4.6	调出宏包手册	29
4.7	编译文件	30
4.8	不懂 vim 的处理方法	31
5	使用编辑器	33
5.1	Windows 10 系统	33
5.1.1	TeXworks	33
5.1.2	TeXstudio	35
5.2	Ubuntu 20.04 系统	38
5.3	macOS	39
5.3.1	TeXShop	40
5.3.2	TeXstudio	40
5.4	Windows Subsystem for Linux	40
5.5	其他编辑器	40
5.6	多款编辑器对比参考	41
6	在线的 L^AT_EX 平台	42
6.1	Overleaf	42
6.1.1	注册 Overleaf	42
6.1.2	使用 Overleaf	42

目录	4
6.1.3 选择不同发行版版本	43
6.1.4 学习与帮助	43
6.2 TeXPage	43
6.2.1 注册 TeXPage	43
6.2.2 使用 TeXPage	43
6.2.3 文档和帮助	43
6.2.4 TeXPage 的优势	43
附录 A 大陆地区的源	44
附录 B 有关 WSL 的一点补充内容	46
附录 C 离线安装宏包	49
附录 D 版本信息	50

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** \gg **显示/隐藏** 中选择 **文件扩展名**.

用户可以从最近的 CTAN 源¹下载 T_EX Live 的 iso 镜像文件, 也可以找大陆地区的源自行下载². 通过大陆地区的源下载 iso 文件的方法请参考附录 A.

假设镜像文件被下载到本地后, 其存放路径为 X:\Y³. 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 C_TE_X 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.

下面我们要确保下载到的 iso 文件没有损坏, 一种常用做法是检查其 MD5 值. 当然, 如果确信自己下载到的文件没有问题, 也可以跳过这一步. 打开 cmd 窗口依次执行以下代码

```
1 cd /d X:\Y
2 certutil -hashfile texlive2022.iso md5
```

若系统显示

```
1 MD5 的 texlive2022.iso 哈希:
2 adadcf7ea5224f174ec2abcea6923a42
3 CertUtil: -hashfile 命令成功完成。
```

则表明下载的镜像文件正常. 除 MD5 值外, 还可检查 SHA512 值. 依然是在 cmd 执行


```
1 cd /d X:\Y
2 certutil -hashfile texlive2022.iso sha512
```

若系统显示

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 , 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** \gg **此电脑** \gg **属性** \gg **高级系统设置** \gg **环境变量** \gg **系统变量** \gg **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

```

1 SHA512 的 texlive2022.iso 哈希:
2 b3d4d2e9b0e3d34a7ca0763fb5c669bb9a26518f86af5cc292c56f7d61ff25dc5fce3373f814d970d99b0d40df393ad
   594336cf1b35a95a9655289666d964d59
3 CertUtil: -hashfile 命令成功完成。

```

即为验证成功.

接下来开始安装. 推荐用户先手动建立 TeX Live 的安装路径 (文件夹) 再进行安装, 注意路径的名称是**不带空格的英文**, 默认路径为 C:\texlive\2022. 假如系统用了中文作为用户名, 那么用户需要先按照 1.1.3 节的方法更改临时环境变量后再安装. 将镜像文件装载至虚拟光驱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, 打开 cmd 窗口并执行

```
1 Z:\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 = help> https://tug.org/texlive/doc/install-tl.html <=====
6
7 Detected platform: Windows
8
9 <B> set binary platforms: 1 out of 6
10
11 <S> set installation scheme: scheme-full
12
13 <C> set installation collections:
14     41 collections out of 41, disk space required: 7462 MB
15
16 <D> set directories:
17     TEXDIR (the main TeX directory):
18     C:/texlive/2022
19     TEXMFLOCAL (directory for site-wide local files):
20     C:/texlive/texmf-local
21     TEXMFSYSVAR (directory for variable and automatically generated data):
22     C:/texlive/2022/texmf-var
23     TEXMFSYSCONFIG (directory for local config):
24     C:/texlive/2022/texmf-config
25     TEXMFVAR (personal directory for variable and automatically generated data):
26     ~/.texlive2022/texmf-var
27     TEXMFCONFIG (personal directory for local config):
28     ~/.texlive2022/texmf-config
29     TEXMFHOME (directory for user-specific files):
30     ~/.texmf

```

⁹Windows 10 默认双击镜像文件便可装载; 受某些压缩软件影响双击失效时, 可在文件夹菜单栏 **管理**》**光盘映像工具**》**装载**; 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

```

31
32 <O> options:
33   [ ] use letter size instead of A4 by default
34   [X] allow execution of restricted list of programs via \write18
35   [X] create all format files
36   [X] install macro/font doc tree
37   [X] install macro/font source tree
38   [X] adjust search path
39   [1] add menu items, shortcuts, etc.
40   [1] update file associations
41   [X] install TeXworks front end
42   [X] after install, set CTAN as source for package updates
43
44 <V> set up for portable installation
45
46 Actions:
47 <I> start installation to hard disk
48 <P> save installation profile to 'texlive.profile' and exit
49 <Q> quit
50
51 Enter command:

```

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

```

1 =====
2 Directories customization:
3
4 <1> TEXDIR: C:/texlive/2022
5   main tree: C:/texlive/2022/texmf-dist
6
7 <2> TEXMFLOCAL: C:/texlive/texmf-local
8 <3> TEXMFSYSVAR: C:/texlive/2022/texmf-var
9 <4> TEXMFSYSCONFIG: C:/texlive/2022/texmf-config
10
11 <5> TEXMFVAR: ~/.texlive2022/texmf-var
12 <6> TEXMFCONFIG: ~/.texlive2022/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/2022]:
```

这时可更改路径, 如 D:/texlive/2022. 接下来, 按键盘 **[R]** 即可回到初始的安装界面, 再按键盘 **[I]** 便可在刚才指定的路径安装. 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¹⁰. 安装完成后, 关闭再打开 cmd 窗口¹¹, 执行

```
1 tex -v
```

显示

```
1 TeX 3.141592653 (TeX Live 2022)
2 kpathsea version 6.3.4
3 Copyright 2022 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¹². 弹出前面装载的光盘镜像, 安装到此结束.

1.1.1 下载镜像一直不成功的处理办法

目前发现有些用户使用某些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/texlive2022.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/texlive2022.iso
```

其他国内镜像只需要将链接替换即可, 链接和方法请参考附录 A.

1.1.2 需要管理员权限的处理办法

有些用户反馈安装过程中需要管理员权限. 以管理员身份打开 cmd 窗口的方法是点击 **[Win]**, 输入 cmd, 按 **[Ctrl]+[Shift]+[Enter]** 三个键. 一般而言, 在系统盘内直接安装 TeX Live 会碰到此类问题, 因此鼓励用户在安装前先手动建立安装路径. 直接使用管理员身份打开 cmd 窗口进行安装可能会对以后的升级造成影响.

¹⁰Windows 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

¹¹正常安装 TeX 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

¹²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> 需与你方才的安装路径一致

1.1.3 系统用中文作为用户名

很多中文 Windows 用户习惯于用中文作为用户名, 但是这样会导致 T_EX Live 2021 安装失败. 更改系统的 TEMP 和 TMP 这两个环境变量可以避免这样的问题. 在 cmd 中执行

```
1 mkdir C:\temp
2 set TEMP=C:\temp
3 set TMP=C:\temp
```

便可更改这两个环境变量为 C:\temp. 更改后按照前述内容安装 T_EX Live 2022 即可. 这样的更改是临时的, 待 cmd 窗口关闭, 环境变量将恢复为原始状态.

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. 下面介绍几种卸载方法.

1.2.1 使用批处理文件

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¹³. 接下来, 用户只需执行¹⁴安装路径中的卸载批处理文件即可实现卸载, 如默认安装时执行

```
1 C:\texlive\2022\tlpkg\installer\uninst.bat
```

结束以上步骤后, 手动删除用户文件夹中的 .texlive2022. 如果用户安装 T_EX Live 时需要管理员权限, 那么在卸载时同样需要管理员权限.

卸载后, 用户需要清理注册表

```
1 HKEY_CURRENT_USER\Software\Classes\TL.bitmap2eps.convert.2022
2 HKEY_CURRENT_USER\Software\Classes\TL.DVIOUT.view.2022
3 HKEY_CURRENT_USER\Software\Classes\TL.TeXworks.edit.2022
4 HKEY_CURRENT_USER\Software\Classes\TL.PSViewer.view.2022
5 HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts\TL.
  bitmap2eps.convert.2022_.jpg
6 HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts\TL.
  bitmap2eps.convert.2022_.png
7 HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts\TL.
  TeXworks.edit.2022_.sty
8 HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\ApplicationAssociationToasts\TL.
  TeXworks.edit.2022_.tex
```

但若在安装时使用管理员权限“为所有人安装”, 则注册表位置将变为

¹³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

¹⁴双击或在 cmd 中运行

```
1 HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TL.bitmap2eps.convert.2022
2 HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TL.DVIOUT.view.2022
3 HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TL.TeXworks.edit.2022
4 HKEY_LOCAL_MACHINE\SOFTWARE\Classes\TL.PSViewer.view.2022
```

1.2.2 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 $\text{T}_{\text{E}}\text{X}$ Live 的环境变量, 并且清理注册表.

1.3 跨版本升级 $\text{T}_{\text{E}}\text{X}$ 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¹⁵.

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, 方法见附录 A.

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.

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¹⁶.

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
```

¹⁵原文是: There is no comparable upgrade procedure for Windows. Doing a new installation is necessary.

¹⁶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

1.4.3 升级宏包的必要性

自本手册撰写以来, 每年镜像文件打包时都可能存在一些问题, 具体讨论见[这里](#). 用户安装完毕后, 请按照本节提供的方法升级.

1.5 安装宏包

在默认状态下, 用户将完整安装 T_EX 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 调出宏包手册

当正确安装后, 用户可以调出宏包手册以查阅相应内容. 例如在 `cmd` 执行

```
1 texdoc texlive-zh-cn
```

或

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

就可分别调出 `texlive-zh-cn.pdf` 和 `lshort-zh-cn.pdf`, 这两本手册中都有讲解安装的相关内容, 建议用户阅读.

本手册业已被 T_EX Live 收录, 调出本手册的命令为

```
1 texdoc install-latex-guide-zh-cn
```

1.7 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¹⁷建立一个 `tex` 文件:

```
1 mkdir D:\work-latex
2 cd /d D:\work-latex
3 notepad main.tex
```

第 1 行表示创建一个工作路径 `D:\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¹⁸. 在打开的记事本中编写一个最小示例

¹⁷以下称工作路径

¹⁸Windows 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

```

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 `D:\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¹⁹

```

1 \documentclass[UTF8,fontset=windows]{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**.

总结一下, 在使用 \LaTeX 时, 用户首先要在工作路径²⁰中建立一个完整的 `tex` 文件²¹. 当 `tex` 文件内容确定后, 再保存文件并使用编译命令²², 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
```

有些编译还需要额外调用系统命令, 例如使用 `minted` 包就有这类要求, 因此可以在执行编译时添加参数

```
1 pdflatex -shell-escape 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²³.

¹⁹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

²⁰建议不同的工程放入不同的工作路径

²¹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

²²目前常用的编译命令为 `pdflatex` 和 `xelatex`

²³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

1.7.1 L^AT_EX 2_ε版本不匹配导致 xelatex 失败

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 tlmgr install --reinstall xetex
```

1.7.2 编译出现 l3backend-* 问题

这是 fmt 文件未能更新所导致的问题. 可以用

```
1 fmtutil-sys --all
```

刷新 fmt 文件后再编译 tex 文件.

2 Ubuntu 20.04 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 2022, 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 镜像文件的方法见 1.1 节. 下载完毕后, 打开 Terminal 窗口验证 MD5 值, 即执行以下命令

```
1 cd ~/Downloads
2 md5sum texlive2022.iso
```

若显示

```
1 adadcf7ea5224f174ec2abcea6923a42 texlive2022.iso
```

则镜像文件下载正确. 同样可验证 SHA512 值, 执行

```
1 cd ~/Downloads
2 sha512sum texlive2022.iso
```

正确的返回结果为

```
1 b3d4d2e9b0e3d34a7ca0763fb5c669bb9a26518f86af5cc292c56f7d61ff25dc5fce3373f814d970d99b0d40df393ad
   594336cf1b35a95a9655289666d964d59 texlive2022.iso
```

将下载的光盘镜像进行装载

```
1 sudo mkdir /mnt/texlive
2 sudo mount ./texlive2022.iso /mnt/texlive
```

接下来执行

```
1 sudo /mnt/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 = help> https://tug.org/texlive/doc/install-tl.html <=====
6
7 Detected platform: GNU/Linux on x86_64
```

```
8
9 <B> set binary platforms: 1 out of 6
10
11 <S> set installation scheme: scheme-full
12
13 <C> set installation collections:
14     40 collections out of 41, disk space required: 7262 MB (free: 239411 MB)
15
16 <D> set directories:
17     TEXDIR (the main TeX directory):
18         /usr/local/texlive/2022
19     TEXMFLOCAL (directory for site-wide local files):
20         /usr/local/texlive/texmf-local
21     TEXMFSYSVAR (directory for variable and automatically generated data):
22         /usr/local/texlive/2022/texmf-var
23     TEXMFSYSCONFIG (directory for local config):
24         /usr/local/texlive/2022/texmf-config
25     TEXMFVAR (personal directory for variable and automatically generated data):
26         ~/.texlive2022/texmf-var
27     TEXMFCONFIG (personal directory for local config):
28         ~/.texlive2022/texmf-config
29     TEXMFHOME (directory for user-specific files):
30         ~/texmf
31
32 <O> options:
33     [ ] use letter size instead of A4 by default
34     [X] allow execution of restricted list of programs via \write18
35     [X] create all format files
36     [X] install macro/font doc tree
37     [X] install macro/font source tree
38     [ ] create symlinks to standard directories
39     [X] after install, set CTAN as source for package updates
40
41 <V> set up for portable installation
42
43 Actions:
44 <I> start installation to hard disk
45 <P> save installation profile to 'texlive.profile' and exit
46 <Q> quit
47
48 Enter command:
```

点击键盘 进行默认安装. 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, 如安装路径等. 为简化起见, 以下内容我们均认为用户采用默认安装. 安装完毕后, 将装载的光盘镜像弹出并删除文件夹

```
1 sudo umount /mnt/texlive
```



```
2 sudo rm -r /mnt/texlive
```

安装完成后, 用户需要设置环境变量. 继续在 Terminal 中执行

```
1 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/2022/bin/x86_64-linux:$PATH
3 export MANPATH=/usr/local/texlive/2022/texmf-dist/doc/man:$MANPATH
4 export INFOPATH=/usr/local/texlive/2022/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 3.141592653 (TeX Live 2022)
2 kpathsea version 6.3.4
3 Copyright 2022 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. 在 Terminal 中执行

```
1 sudo cp /usr/local/texlive/2022/texmf-var/fonts/conf/texlive-fontconfig.conf /etc/fonts/conf.d/09-
  texlive.conf
```

将配置文件复制到系统, 然后继续执行

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

刷新字体缓存. 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 [texlive-zh-cn](#).

如果用户更改了安装路径, 要注意将

```
1 /usr/local/texlive/2022/
```

在各处替换.

2.1.1 在用户文件夹安装

如果有用户打算在用户文件夹安装 TeX Live, 那么可以不必调用权限, 直接执行

```
1 /mnt/texlive/install-tl
```

只不过这时安装路径需要选择

```
1 ~/texlive/2022
```

安装完毕后, 继续修改环境变量为

```
1 # Add TeX Live to the PATH, MANPATH, INFOPATH
2 export PATH=~/texlive/2022/bin/x86_64-linux:$PATH
3 export MANPATH=~/texlive/2022/texmf-dist/doc/man:$MANPATH
4 export INFOPATH=~/texlive/2022/texmf-dist/doc/info:$INFOPATH
```

有关字体的处理变为

```
1 cp ~/texlive/2022/texmf-var/fonts/conf/texlive-fontconfig.conf ~/.fonts.conf/09-texlive.conf
```

然后执行

```
1 fc-cache -fv
```

实际上, 无论将 TeX Live 安装到哪里, 字体的配置文件都可以仿照上面只复制于用户文件夹中.

2.1.2 fc-cache 无法执行

这种情况往往是缺少了 fontconfig, 用户可以在命令行中执行以下命令安装

```
1 sudo apt install fontconfig
```

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
```

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¹

```
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/ focal main restricted universe multiverse
3 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
4 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
5 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe
6     multiverse
7 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe
8     multiverse
9 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe
10     multiverse
11 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe
12     multiverse
13 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe
14     multiverse
```

¹本段文字可至[清华大学镜像网站](https://mirrors.tuna.tsinghua.edu.cn/)获取

```

10
11 # 预发布软件源, 不建议启用
12 # deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe
    multiverse
13 # deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-proposed main restricted universe
    multiverse

```

然后执行

```
1 sudo apt update && sudo apt upgrade
```

换源并更新. 若更改错误, 可执行

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

恢复文件.

2.2 卸载 TeX Live

下面提供几种卸载方法, 卸载后需注意清理环境变量.

2.2.1 卸载源内的版本

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

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

来删除相应的依赖包.

2.2.2 手动卸载

如果是从光盘镜像安装, 直接删除文件夹即可. 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/2022
2 rm -rf ~/.texlive2022
```

卸载完成后, 可以进一步移除之前设置的环境变量. 在 Terminal 中执行

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

然后移除之前在文件末尾添加的

```
1 # Add TeX Live to the PATH, MANPATH, INFOPATH
2 export PATH=/usr/local/texlive/2022/bin/x86_64-linux:$PATH
3 export MANPATH=/usr/local/texlive/2022/texmf-dist/doc/man:$MANPATH
4 export INFOPATH=/usr/local/texlive/2022/texmf-dist/doc/info:$INFOPATH
```

并保存退出. 同时在 Terminal 中执行

```
1 sudo visudo
```

在 `secure_path` 中删除

```
1 /usr/local/texlive/2022/bin/x86_64-linux:
```

2.2.3 使用 `tlmgr` 工具

同样, 用户也可以使用 `tlmgr` 工具卸载²,

```
1 sudo tlmgr remove --all
```

2.3 跨版本升级 `TeX 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 `TeX Live`³.

2.4 升级宏包

首次升级前, 在 Terminal 中执行

```
1 sudo visudo
```

将

```
1 /usr/local/texlive/2022/bin/x86_64-linux:
```

添加在 `secure_path` 中⁴. 然后依次 `Ctrl`+`X`, `Y`, `Enter` 保存退出.

接下来在 Terminal 中执行

²`sudo tlmgr` 无法使用的情况可以参考 2.4 节解决

³原文是: By default, please get the new TL by doing a new installation instead of proceeding here

⁴这里只讨论默认安装的情况

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

让系统自动选择源, 同样可以使用大陆地区的源升级, 详情见附录 A. 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 20.04 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 宏包.

2.6 调出宏包手册

与 Windows 10 类似, 当正确安装后, 用户可以调出宏包手册以查阅相应内容. 例如在 Terminal 中执行

```
1 texdoc texlive-zh-cn
```

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

```
1 texdoc install-latex-guide-zh-cn
```

就可分别调出 texlive-zh-cn.pdf, lshort-zh-cn.pdf 或本手册.

2.7 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⁵

```
1 \documentclass[UTF8,fontset=fandol]{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 B.

编译命令可添加参数, 这里与 1.7 节中的情形一致.

2.7.1 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. 这或许是因为缺失 `libfontconfig`, 用户可在命令行中执行

```
1 sudo apt install libfontconfig1
```

进行安装.

⁵ `gedit` 默认使用 UTF-8 编码

3 macOS

3.1 安装 MacTeX

MacTeX 是 TeX Live 在 macOS 下的再打包版本, 并额外加入了一些辅助程序. 在 macOS 上安装 MacTeX 可以选择以下两种方式:

直接下载 MacTeX 安装包进行安装 将安装文件下载到本地, 然后按照安装提示进行安装.

借助 Homebrew 以命令行方式安装 先安装 Homebrew, 然后通过命令行进行在线安装.

目前比较推荐直接下载 MacTeX 安装包的安装方式, 因为最近一段时间以来, 大陆用户对 GitHub 的访问极不稳定, 有可能会导导致 Homebrew 安装失败.

3.1.1 直接下载 MacTeX 安装包进行安装

下载 MacTeX 的安装包 `MacTeX.pkg`, 在系统内点击已下载的 `MacTeX.pkg` 文件, 根据系统提示进行安装即可. 这个过程与安装 macOS 上的其他软件并无太多不同.

另外, 用户也可以选择国内各个镜像来下载 `MacTeX.pkg`. 因其独特性, 用户需调整附录 A 中提供的大陆地区的源的网址, 具体来讲, 是将 `.../texlive` 改为 `.../mac/mactex`. 例如, 阿里云的镜像地址为 <https://mirrors.aliyun.com/CTAN/systems/mac/mactex/>, 北京交通大学的镜像地址为 <https://mirror.bjtu.edu.cn/ctan/systems/mac/mactex/>, 其他镜像类似, 不再一一列举.

3.1.2 借助 Homebrew 以命令行方式安装

建议用户学习使用 **Homebrew**, Homebrew 是一个包管理工具, 类似 Ubuntu 上面的 `apt-get`.

安装 Homebrew

Homebrew 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打开方法为:  +  , 输入 `terminal` 并打开  应用

```

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

```

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
```

安装 MacTeX

安装 Homebrew 后, 只需在终端执行以下命令即可完成安装:

```
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 basicctex
```

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.

3.2 卸载 MacTeX

如果用户直接下载了 MacTeX 安装包进行安装, 可以对照[这里](#)的介绍来卸载, 通常我个人会比较建议在跨版本升级前卸载就版本.

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.

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

3.3 跨版本升级 MacTeX

如果用户直接下载了 MacTeX 安装包进行安装, 可以先卸载旧版本, 再安装新版本, 或者干脆保留多个版本, 具体见[这里](#).

如果用户借助 Homebrew cask 安装了 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
```

3.4 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.

3.5 安装宏包

安装 CTAN 中的宏包方法与 2.5 节一致.

3.6 调出宏包手册

调出宏包手册方法与 2.6 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

```
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注意建立最小示例前先确定工作路径

⁴注意使用 UTF-8 编码

```
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 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. 下载及文件验证方法同 1.1 节.

在正式安装前, 用户需要在 `bash` 中执行

```
1 sudo apt install fontconfig
2 sudo apt install ttf-mscorefonts-installer
```

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 2.1.2 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 4.8 节.

接下来, 在主系统中将镜像装载, 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` 中找到装载的光盘镜像. 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 = help> https://tug.org/texlive/doc/install-tl.html <=====
6
7 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

```

8
9 <B> set binary platforms: 1 out of 6
10
11 <S> set installation scheme: scheme-full
12
13 <C> set installation collections:
14     40 collections out of 41, disk space required: 7262 MB (free: 239411 MB)
15
16 <D> set directories:
17     TEXDIR (the main TeX directory):
18         /usr/local/texlive/2022
19     TEXMFLOCAL (directory for site-wide local files):
20         /usr/local/texlive/texmf-local
21     TEXMFSYSVAR (directory for variable and automatically generated data):
22         /usr/local/texlive/2022/texmf-var
23     TEXMFSYSCONFIG (directory for local config):
24         /usr/local/texlive/2022/texmf-config
25     TEXMFVAR (personal directory for variable and automatically generated data):
26         ~/.texlive2022/texmf-var
27     TEXMFCONFIG (personal directory for local config):
28         ~/.texlive2022/texmf-config
29     TEXMFHOME (directory for user-specific files):
30         ~/texmf
31
32 <O> options:
33     [ ] use letter size instead of A4 by default
34     [X] allow execution of restricted list of programs via \write18
35     [X] create all format files
36     [X] install macro/font doc tree
37     [X] install macro/font source tree
38     [ ] create symlinks to standard directories
39     [X] after install, set CTAN as source for package updates
40
41 <V> set up for portable installation
42
43 Actions:
44 <I> start installation to hard disk
45 <P> save installation profile to 'texlive.profile' and exit
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。为简化起见, 以下内容均认为用户进行默认安装。安装完毕后, 需继续在 `bash` 中执行

```

1 sudo umount /mnt/x
2 sudo rmdir /mnt/x

```

弹出已装载的光盘镜像。

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 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/2022/bin/x86_64-linux:$PATH
3 export MANPATH=/usr/local/texlive/2022/texmf-dist/doc/man:$MANPATH
4 export INFOPATH=/usr/local/texlive/2022/texmf-dist/doc/info:$INFOPATH
```

并保存退出. 同样, 不熟悉 `vim` 的用户可参考 4.8 节. 退出 WSL 再进入, 执行

```
1 tex -v
```

将显示

```
1 TeX 3.141592653 (TeX Live 2022)
2 kpathsea version 6.3.4
3 Copyright 2022 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 20.04 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/2022/texmf-var/fonts/conf/texlive-fontconfig.conf /etc/fonts/conf.d/09-
  texlive.conf
```

然后在 Terminal 执行

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

刷新字体缓存.

4.1.1 在用户文件夹安装

WSL 同样也可以在用户文件夹中安装, 它的用户文件夹往往在主系统的系统盘中. 具体过程也可以参考第 2.1.1 节.

4.2 卸载 TeX Live

WSL 中的方法同 2.2 节.

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 调出宏包手册

由于 WSL 的 `bash` 默认是无窗口化的纯命令行, 因此将 T_EX Live 安装至 WSL 后, 用户无法直接通过命令 `texdoc` 打开相应的手册. 下面几种方法均为收集而来的方案, 其中比较推荐更改 `texdoc.cnf`.

更改 `texdoc.cnf`

此方案源自本手册的 [issue](#). 首先在 `bash` 中执行

```
1 texdoc --files
```

系统返回类似如下内容

```
1 /mnt/d/texlive-wsl/2022/texmf-dist/scripts/texdoc/texdoclib.tlu 3.4.1
2 Configuration file(s):
3   active /mnt/d/texlive-wsl/2022/texmf-dist/texdoc/texdoc.cnf
4 Recommended file(s) for personal settings:
5   /home/<username>/texmf/texdoc/texdoc.cnf
```

这里给出了 `texdoc` 的配置文件所在位置. 打开用户文件夹下的配置文件

```
1 vim ~/texmf/texdoc/texdoc.cnf
```

在文件内写入

```
1 viewer_pdf = (powershell.exe -Command Start-Process $(wslpath -w %s)) &
```

并且保存退出. 然后再执行 `texdoc` 即可打开相应文档.

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/2022/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

```

1 cd /usr/local/texlive/2022/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](#).

4.7 编译文件

编译一个最小示例 `main.tex`.

在主系统的 `cmd` 中执行

```

1 mkdir D:\work-latex
2 cd /d D:\work-latex
3 notepad main.tex

```

建立文档, 内容为

```

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

```

接下来在 `cmd` 中执行

```

1 bash -i -c "pdflatex main"

```

实现编译.

也可以直接在 WSL 中建立文档并编译, 在 `bash` 中依次执行

```

1 mkdir /mnt/d/work-latex
2 cd /mnt/d/work-latex
3 vim main.tex

```

建立文档, 内容同上, 然后在 `bash` 中执行

```

1 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, 可以编写以下最小示例²

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

在 `cmd` 中执行

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

或在 `bash` 中执行

```
1 xelatex main
```

等待系统完成编译过程. `xelatex` 同样可以使用 WSL 内安装的字体³, 在 WSL 中安装字体的方法见附录 B.

编译命令可添加参数, 这里与 1.7 节中的情形一致, 不再赘述.

4.8 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. 该方法借助 `VS Code` 的 `Remote-WSL` 插件实现.

先来更改 `.bashrc`. 将 `.bashrc` 备份为 `.bashrc.bak`

```
1 cp ~/.bashrc ~/.bashrc.bak
```

用 `VS Code` 打开文档

```
1 code ~/.bashrc
```

并对其中内容进行更改并保存. 若更改错误, 可以执行

```
1 cp ~/.bashrc.bak ~/.bashrc
```

恢复文档.

接下来更改 `sources.list`. 将 `sources.list` 备份为 `sources.list.bak`

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

用 `VS Code` 打开文档

```
1 code /etc/apt/sources.list
```

²注意使用 UTF-8 编码

³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](#), 鉴于 WSL2 和主系统文件系统互访速度慢, 该方法很可能拖慢 `xelatex` 速度

并对其中内容进行更改, 并且另存为

```
1 ~/sources.list.sav
```

在 `bash` 中执行

```
1 sudo cp ~/sources.list.sav /etc/apt/sources.list
```

替换文档. 若发现文档有错误, 执行

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

恢复文档.

使用 VS Code 同样可以编写 `tex` 文件, 这涉及到如何使用 `LATEX` 的编辑器, 这里不做展开.

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 `TeXworks` 和功能更加丰富的 `TeXstudio`。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 `TeXworks`

`TeXworks` 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，`TeXworks` 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 `TeXworks` 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 `TeXworks` 打开。

刚开始使用 `TeXworks` 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 `TeXworks` 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

`TeXworks` 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 [TeXworks 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，`TeXworks` 默认排版快捷键是 `ctrl+T`

表 5.1: TeXworks 已有定义

Option	Description
<code>\$synctexoption</code>	expands to “-synctex=1” if your tools support SyncTeX
<code>\$fullname</code>	expands to the full name of your root document (e.g. rootfile.tex)
<code>\$basename</code>	expands to the name (without extension) of your root document (e.g. rootfile)
<code>\$suffix</code>	expands to the extension of your root document (e.g. tex)
<code>\$directory</code>	expands to the absolute path to the directory containing your root document

使用模板

TeXworks 支持从模板新建文档, 在 **文件** 菜单中可见.

更改编译命令设置

TeXworks 默认提供了若干编译命令, 具体见 **编辑** **》** **首选项** **》** **排版**. 实际上, 用户可以根据自己的需要, 添加、更改其中的设置. 首先明确 TeXworks 中已有一些定义, 见表 5.2³. 接下来可以定义 `latexmkpdf`, 它调用 `latexmk` 并使用 `pdflatex` 等命令编译文档. 在 **排版** 上点击 **+**, 输入名称 `latexmkpdf` 和程序 `latexmk`, 相关参数需要再次点击 **+** 来予以添加, 例如添加以下这些

```
1 -e
2 $pdflatex=q/pdflatex $synctexoption %0 %S/
3 -pdf
4 $fullname
```

这里的 `$synctexoption` 和 `$fullname` 均是 TeXworks 预定义的内容, 而其他的 `-e`, `$pdflatex=...` 和 `-pdf` 需要用户自己阅读 `latexmk` 手册了解. 然后选择 **运行后查看 pdf 文件**, 再点击 **OK**, 即可发现已经添加成功. 之后用户可以调用 `latexmkpdf` 来编译文档. 当然, 在 **排版** 处还可以将 `latexmkpdf` 设置为默认.

类似地, 如果要定义 `latexmkxe`, 可以考虑以下参数

```
1 -e
2 $xelatex=q/xelatex $synctexoption %0 %S/
3 -xelatex
4 $fullname
```

程序依然是 `latexmk`.

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 `.oxt` 文件. 将 `.oxt` 文件重命名为 `.zip` 文件, 解压缩得到里面的 `.dic` 和 `.aff` 文件. 接下来将它们安装到 `<resources>\dictionaries`⁴. 安装完成后, 在 **编辑** **》** **拼写** 中选择即可.

³参考 [Advanced Typesetting Tools](#).

⁴`<resources>` 的具体位置可以在 **帮助** **》** **TeXworks 配置与资源** 中找到, 若不存在该 `dictionaries` 文件夹, 用户可自行建立.

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 = ../main.tex
```

即可告诉系统上一级目录中的 `main.tex` 是主文档⁵。

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

表示该文档使用“UTF-8”编码⁶。

```
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 T_EXstudio

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[环境变量](#)中是否有 `system32`。具体方法在前面已有说明，此处不再赘述。此外，根据 [Github 上的讨论](#)，Windows 10 21H1 (64bit) 上 `TEXstudio Qt6` 版本存在一些问题，因此目前推荐使用 `Qt5` 版本。以下有关 `TEXstudio` 的用法介绍，还请用户提前在 [Options](#) [configure TeXstudio](#) 中勾选 [Show Advanced Options](#)。

下载方法

`TEXstudio` 有自己的[官方网站](#)，但因网络原因，打开不便。网站上提供的下载链接跳转到了 `github` 上，有兴趣的朋友可以直接到 `github` 上查看[源码](#)和[可执行文件](#)；而清华大学也提供了相应的[镜像](#)，便于用户下载。此外，有[第三方网站](#)可以辅助用户下载 `github` 上的发行文件，网站提供了下载教程，这里不再展开。

⁵主文档和子文档的更多内容请参考 `\include` 和 `\input` 的用法

⁶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`⁷.

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。

⁷ 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 = txs:///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" | txs:///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 TKS-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** \gg **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** \gg **Check LaTeX Installation** 中有 ini 文件的具体位置.

调用外部 PDF 阅读器

TeXstudio 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 \langle SumatraPDFROOT \rangle , 而 TeXstudio 在本地的位置是 \langle TeXstudioROOT \rangle . 打开 **Options** \gg **Configure TeXstudio** 窗口, 在 **Build** \gg **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** \gg **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** \gg **Options** \gg **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 TeXstudio 中对应代码的行首.

5.2 Ubuntu 20.04 系统

简化起见, 这里只介绍如何使用 TeXstudio. 根据官网推荐, 我们安装源内的 TeXstudio. 安装前可参照 2.1.2 节的介绍换源以增速.

在 Terminal 中执行

```
1 sudo apt install texstudio
```

即可安装 TeXstudio. 注意安装过程中会产生一些依赖, 它们包括了源内的 TeX 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 TeXstudio. 以此在 Terminal 中执行

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

鉴于网络原因, 直接从 PPA 下载容易丢包. 这里介绍中科大的反向镜像, 将

```
1 /etc/apt/sources.list.d/
```

中文件的内容

```
1 http(s)://ppa.launchpad.net
```

改为

```
1 https://launchpad.proxy.ustclug.org
```

而后再

```
1 sudo apt update
```

即完成换源. 有时地址太多而逐个换太麻烦, 可以直接使用如下命令直接替换⁸

```
1 sudo find /etc/apt/sources.list.d/ -type f -name "*.list" -exec sed -i.bak -r 's#deb(-src)?\s*http(s)?://ppa.launchpad.net#deb\1 https\2://launchpad.proxy.ustclug.org#ig' {} \;
```

安装完毕后, 用户可以在 Terminal 中执行

```
1 texstudio main.tex
```

使用 TeXstudio 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 macOS

这里介绍两款 macOS 下的编辑器: TeXShop 和 TeXstudio.

⁸参见 <https://mogeke.me/2018/035/>

5.3.1 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.

5.3.2 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](#).

5.4 Windows Subsystem for Linux

若使用 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⁹

```
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.

5.5 其他编辑器

VS Code 目前已有插件 `LaTeX Workshop` 供用户使用. 具体设置方式建议阅读插件手册, 一些网友的教程可以参考, 如 [EthanDeng 的教程](#). 另外, VS Code 的 `Remote-WSL` 插件可以帮助用户更方便地使用 WSL. 参考配置文件见附录 B.

⁹这里省略了中间路径, 用户根据自己的情形进行补充即可

Vim 是一款优秀的编辑器, 它的用法较目前常见的编辑器有很大不同, 因此用户上手时容易感觉无所适从, 但熟练掌握用法后, 用户将很容易对其进行定制. **Vim-LaTeX** 尝试提供一套全面的工具来查看、编辑和编译 LaTeX 文档, 用户可以考虑使用.

5.6 多款编辑器对比参考

显然, 能够支持 \LaTeX 的编辑器并不局限于本手册所介绍的几款. 表 5.2 是 [ddswhu](#) 在 \LaTeX 小圈子内部对 \LaTeX 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 \LaTeX 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 5.2: \LaTeX 编辑器对比

	WinEdt	T _E Xstudio	T _E Xworks	Sublime Text	VS Code
插件依赖				L^AT_EX Tools	L^AT_EX 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	开源软件
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 在线的 L^AT_EX 平台

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 和 TeXPage 的相关内容.

6.1 Overleaf

6.1.1 注册 Overleaf

Overleaf 是全球范围内首屈一指的在线 L^AT_EX 编辑平台. 它为每位用户提供了 Ubuntu 系统下的 T_EX 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. 2022 年 8 月 10 日,它将后台的 T_EX Live 升级为 2022 版本,同时,Overleaf 还允许用户自主选择项目中的 T_EX 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 C_T_EX 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.

6.1.2 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**.

另外,Overleaf 在 9 月 30 日更新了新的功能, **Stop on first error**, 这个功能一旦开启,出现报错就立即停止编译. 它可以让用户高度关注报错信息并改正,也可以帮助纠正一些会导致超时 (timeout) 的代码错误,例如 tikz 里的 `\draw` 忘了最后的 `;`.

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.

6.1.3 选择不同发行版版本

Overleaf 将后台 T_EX Live 升级后, 用户新建项目默认使用 T_EX Live 2022, 而老项目还是使用 T_EX Live 的老版本. 如果用户打算使用 T_EX Live 2022, 只需在 `Menu` `Settings` `TeX Live version` 选择版本即可.

6.1.4 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.

6.2 TeXPage

6.2.1 注册 TeXPage

TeXPage 是由国内公司开发的在线 L^AT_EX 平台. 相较于 Overleaf 的注册困难, TeXPage 的注册则要方便得多. 用户只需访问它的主页, 使用邮箱注册即可, 具体过程不再赘述.

6.2.2 使用 TeXPage

新用户注册 TeXPage 后会在页面看到一份使用教程. 这份教程简明扼要地概括了一般中文用户会遇到的常见问题, 然而从代码的角度而言, 我个人不欣赏在浮动体中使用 `H` 选项的做法.

在页面右上角的位置有许多菜单, 在设置菜单中, 用户可以选择默认的编译命令、发行版版本等等. 设置完毕, 再点击编译即可.

6.2.3 文档和帮助

TeXPage 有自己的文档中心, 文档数量不多, 但涵盖了相当一部分基础知识. 同时, 它也提供了联系方式 support@texpage.com, 用户如果遇到了一些使用上的问题也可以直接发邮件咨询.

6.2.4 TeXPage 的优势

在全球范围内, Overleaf 依然是在线 L^AT_EX 平台的主流, 然而由于种种原因, 大陆地区用户体验不佳, 因此 TeXPage 就成为了一个稳定有效的新的选择.

目前在 TeXPage 上也具有一些颇具特色的模板, 例如建模比赛的一些模板就已经被收录在 TeXPage 当中. 如果未来 TeXPage 能够再尽可能收录大陆各个高校的毕业论文模板的话, 那可能是一件非常有趣的事情.

附录 A 大陆地区的源

由于各种不知名的原因, 下载 TeX Live 的 iso 文件或升级宏包时, 系统自动选择的源并没有满足用户关于“附近”的要求. 因此, 很多时候, 用户需要手动选择 CTAN 的源. 目前已搜集到的大陆地区的源见表 A.1. 在 CTAN 上面, 也可以找到[亚洲镜像列表](#)和[连接情况](#). 接下来简述使用大陆地区的源下载 iso 文件和升级宏包.

使用最近的 CTAN 源下载 iso 文件, 实际上是下载

```
1 https://mirrors.ctan.org/systems/texlive/Images/texlive2021.iso
```

使用大陆的源下载它, 只需将以上网址中的

```
1 https://mirrors.ctan.org/systems/texlive/
```

替换为相应的源的地址, 而将

```
1 Images/texlive2021.iso
```

予以保留. 以清华大学为例, 从清华大学源中下载 iso 镜像文件, 实际上是下载

```
1 https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/Images/texlive2021.iso
```

同理, 升级宏包也只需要将对应的网址替换为源地址即可. 前面正文中所设置的

```
1 tlmgr option repository ctan
```

等价于

```
1 tlmgr option repository http://mirror.ctan.org/systems/texlive/tlnet
```

使用大陆的源升级宏包, 只需将

```
1 http://mirror.ctan.org/systems/texlive/
```

替换为大陆源的地址而保留其他部分即可. 依旧以清华大学为例, 那么设置将变为

```
1 tlmgr option repository https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/tlnet
```

表 A.1: 大陆地区目前可用的源 (名称按拼音排序)

阿里云	https://mirrors.aliyun.com/CTAN/systems/texlive/
北京交通大学	https://mirror.bjtu.edu.cn/ctan/systems/texlive/
北京理工大学	https://mirrors.bit.edu.cn/CTAN/systems/texlive/
北京外国语大学	https://mirrors.bfsu.edu.cn/CTAN/systems/texlive/
重庆大学	https://mirrors.cqu.edu.cn/CTAN/systems/texlive/
哈尔滨工业大学	https://mirrors.hit.edu.cn/CTAN/systems/texlive/
华为云	https://mirrors.huaweicloud.com/CTAN/systems/texlive/
华中科技大学	http://mirrors.hust.edu.cn/CTAN/systems/texlive/
兰州大学	https://mirror.lzu.edu.cn/CTAN/systems/texlive/
南方科技大学	https://mirrors.sustech.edu.cn/CTAN/systems/texlive/
南京大学	https://mirrors.nju.edu.cn/CTAN/systems/texlive/
清华大学	https://mirrors.tuna.tsinghua.edu.cn/CTAN/systems/texlive/
上海交通大学	https://mirrors.sjtug.sjtu.edu.cn/ctan/systems/texlive/
上海科技大学	https://mirrors.geekpie.club/CTAN/systems/texlive/
腾讯云	https://mirrors.cloud.tencent.com/CTAN/systems/texlive/
中国科学技术大学	https://mirrors.ustc.edu.cn/CTAN/systems/texlive/

附录 B 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
```

实际上有部分字体可直接使用源进行安装, 如安装 Noto CJK

```
1 sudo apt-get install fonts-noto-cjk  
2 sudo apt-get install fonts-noto-cjk-extra
```

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 [L^AT_EX 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 [latex-editor-sumatrapdf](#) 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         "-xelatex",
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 // 1,

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 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

附录 C 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 D 版本信息

v2022.10.1

1. Overleaf 添加 Stop on first error 功能

v2022.9.1

1. Overleaf 升级至 TeXLive 2022

v2022.8.1

1. 补充 MacTeX 卸载和跨版本升级的一些内容
2. 添加了一个编译参数的例子
3. 删除了失效的国内镜像

v2022.7.1

1. 补充 TeXPage 相关内容

v2022.5.1

1. Windows、Ubuntu 和 WSL 下安装 TeX Live 2022 的内容更新
2. 统一挂载、加载为装载, 与现行 Windows 系统名称统一

v2022.4.1

1. 根据 Dongsheng Deng 提供的内容, 修改 macOS 安装文本

v2022.3.1

1. 更改 TeXworks 的编译命令设置

v2021.12.1

1. T_EXstudio Qt6 版本在 Windows 系统上存在问题

v2021.11.1

1. Overleaf 升级为 T_EX Live 2021

v2021.10.1

1. 添加 ubuntu 和 WSL 在用户文件夹安装的相关内容

v2021.8.1

1. 改动一些文本上的小问题
2. 改动错误的链接
3. 添加南方科技大学镜像
4. 添加 Windows 10 系统内注册表位置

v2021.7.3

1. 添加 Windows 中文用户名处理方法的提示
2. 添加 T_EXworks 拼写检查文件夹建立的提示

v2021.5.1

1. 添加 Windows 中文用户名导致安装失败的解决方法
2. 添加升级宏包的必要性
3. 更正两处超链接

v2021.4.7

1. 更新安装 T_EX Live 2021 内容
2. 添加东莞理工学院源
3. 补充 T_EXstudio 下载方法
4. 补充 WSL 中使用 `texdoc` 的方法

v2021.2.1

1. 更新 l3backend-* 解决方案
2. 补充调整安装路径后的操作
3. 调整大陆镜像地址

v2020.12.1

1. 更新哈工大源
2. 添加 CTAN 镜像列表和连接情况网址
3. 调整 Ubuntu 和 WSL 字体相关内容

v2020.11.1

1. Overleaf 后台更新为 T_EXLive 2020
2. 补充卸载方法
3. 补充 PPA 反向代理安装 T_EXStudio 的方法

v2020.10.1

1. 增加北京外国语大学镜像链接
2. 修正 Windows 10 中用 `sudo` 命令的错误
3. 完善 Windows 下手动删除涉及的注册表位置

v2020.8.2

1. 增加验证 sha512 的内容
2. 将本文提交至 CTAN 而进行必要更改
3. 利用 `texdoc` 调出本手册
4. 给出直接下载 MacT_EX 的链接

v2020.7.1

1. TeXworks 拼写检查调整
2. VS Code 是开源软件
3. 改摘要为前言并增加内容
4. 增加调出宏包手册内容

v2020.6.1

1. 重述大陆地区的源的相关内容
2. 为适应 Ubuntu 20.04 调整某些内容
3. 调整有关管理员权限的内容
4. 调整有关编辑器的内容
5. 借助网站加速下载 TeXstudio
6. 提示卸载时的权限
7. 给出 make.bat 与 makefile
8. 增加 MD5 的说明
9. 更正设置主文档的错误

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 C_TE_X 套装和发行版、编辑器
2. 更改 Windows 10 安装 T_EX 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 T_EXstudio 数学符号表和插入、改写调整
5. 解决 L^AT_EX 2_ε版本不匹配导致 x_el_at_ex 失败的问题
6. 添加打开 cmd 的方法
7. 改写 T_EXstudio 显示行号的设置
8. 更新 MiK_TE_X 相关网址
9. 添加调整 T_EXworks 字体的内容
10. 添加 Windows 中手动卸载 T_EX 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 T_EXstudio 的下载网址
3. 添加 macOS 中 x_el_at_ex 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 TeXworks 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