系统开发工具基础第一次实验报告

LaTeX by 阳鑫桐 23120011028

2025 年 9 月 2 日

目录

1 实验目的

本次实验主要学习了 Git 版本控制工具和利用 Latex 语言规范书写格式。通过本次实验，可以学习到利用 Git 控制开发版本并和远程代码托管平台建立联系，为将来企业实际开发应用打下基础；同时学习 Latex 基本语法，有助于养成清晰、结构化的写作习惯。

2 Git

安装 Git 并学习 Git 基本操作

2.1 初始化用户信息

Git 要求设置每一次使用的用户信息，因此通过命令设置用户名和邮箱

```
24695@LAPTOP-JQI687BK MINGW64 ~ (master)
$ git config --global user.name "yxt"

24695@LAPTOP-JQI687BK MINGW64 ~ (master)
$ git config --global user.email "xty398305@gmail.com"
```

图 1: 设置用户数据

2.2 配置环境文件

通过.bashrc 文件配置环境，解决中文乱码问题，并且给部分复杂命令设置别名，使其在后续过程中中便于使用

```
24695@LAPTOP-JQI687BK MINGW64 ~ (master)
$ touch ~/.bashrc

24695@LAPTOP-JQI687BK MINGW64 ~ (master)
$ source ~/.bashrc
```

图 2: 配置文件并更新

测试命令别名

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ ll
total 24
drwxr-xr-x 1 24695 197609 0 Aug 30 22:07 ./
drwxr-xr-x 1 24695 197609 0 Aug 30 22:00 ../
drwxr-xr-x 1 24695 197609 0 Aug 30 22:07 .git/
```

图 3: 测试别名

2.3 建立仓库

在文件夹中打开 Git Bash，将该文件夹初始化为本地仓库

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository
$ git init
Initialized empty Git repository in D:/gitRepository/.git/
```

图 4: 创建仓库

2.4 创建文件

在仓库中创建文件 file01.txt，此时文件处于 unstaged 状态

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ touch file01.txt
```

图 5: 创建文件

2.5 检查仓库状态

通过 git status 命令检查仓库状态，可以看到此时 file01.txt 并未添加到暂存区

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file01.txt

nothing added to commit but untracked files present (use "git add" to track)
```

图 6: 检查仓库状态

2.6 添加文件到暂存区

将 file01.txt 添加到暂存区，文件状态变为 staged

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git add .
```

图 7: 添加文件到暂存区

2.7 提交

将此次修改提交为一个版本，并添加备注

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git commit -m "add file01"
[master (root-commit) 55e5ed5] add file01
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 file01.txt
```

图 8: 提交

2.8 查看提交日志

查看提交记录，可以看到刚才的提交记录，以及对应的版本号、备注等信息

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git log
commit 55e5ed5183d262afa26094e17a5a46cf7a94c433 (HEAD -> master)
Author: yxt <123456@qq.com>
Date:   Sat Aug 30 22:21:09 2025 +0800

    add file01
```

图 9: 提交日志

2.9 修改文件再次提交

修改文件，再次将文件放入暂存区，并提交

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ vi file01.txt
```

图 10: 修改文件

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ vi file01.txt

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git add file01.txt
warning: in the working copy of 'file01.txt', LF will be replaced by CRLF the n
xt time Git touches it
```

图 11: 再次提交

2.10 回退

通过 log 查看历史提交记录和对应备注, 可以看到当前处于的版本号

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git-log
* 471c4cc (HEAD -> master) update file01
* 55e5ed5 add file01
```

图 12: 查看提交记录

通过版本号，回退到我们想要的版本。回退后再次查看，可以看到已经回退到初始的提交版本

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git reset --hard 55e5ed5
HEAD is now at 55e5ed5 add file01

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git-log
* 55e5ed5 (HEAD -> master) add file01
```

图 13: 回退

2.11 恢复

若回退后，想要回到回退之前的版本，此时 log 命令不能查看到回退前的版本号，需要通过 reglog 命令，查看到历史操作记录，此时可以看到回退前的版本号，通过版本号可再次回退到之前的版本

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git reflog
55e5ed5 (HEAD -> master) HEAD@{0}: reset: moving to 55e5ed5
471c4cc HEAD@{1}: reset: moving to HEAD
471c4cc HEAD@{2}: reset: moving to 471c4c
471c4cc HEAD@{3}: commit: update file01
55e5ed5 (HEAD -> master) HEAD@{4}: commit (initial): add file01
```

图 14: 查看历史操作

2.12 查看分支状态

可以查看当前所有的分支，此时还没有创建新的分支，因此只显示默认分支 master

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git branch
* master
```

图 15: 查看分支状态

2.13 创建分支

创新新的分支 dev01，并再次查看当前分支状态，可以看到当前共有两个分支，master 和 dev01

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git branch dev01

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git branch
  dev01
* master
```

图 16: 创建分支

2.14 切换分支

默认分支为 master，因此要处于其他分支需要执行切换分支操作，可以看到，本次操作过后切换到 dev01 分支

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git checkout dev01
Switched to branch 'dev01'
```

图 17: 切换分支

2.15 合并分支

当在其他分支上进行了一些操作，操作完成之后，需要合并分支，将修改更新，因此切换到 master，将其他分支合并到 master

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git merge dev01
Already up to date.

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git-log
* 708dbb7 (HEAD -> master) add file02
* 55e5ed5 (dev02, dev01) add file01

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git branch -d dev02
Deleted branch dev02 (was 55e5ed5).
```

图 18: 合并分支

2.16 分支合并冲突

在不同分支同时对同一文件做了修改后，在合并时就会发生冲突，Git 会提示冲突，并且把修改都显示在对应文件中。此时应打开对应文件，选择要保留的修改，保存即可

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git merge dev
Auto-merging file02.txt
CONFLICT (content): Merge conflict in file02.txt
Automatic merge failed; fix conflicts and then commit the result.
```

图 19: 分支合并冲突

修改后重新提交

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master|MERGING)
$ vi file02.txt

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master|MERGING)
$ git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
        add

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master|MERGING)
$ git add .

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master|MERGING)
$ git commit -m "after update"
[master 47f2a7f] after update
```

图 20: 重新提交

2.17 远程连接

Git 可以将文件托管到源成代码平台，这里使用 GitHub 作为操作实例。使用 HTTPS 连接到远程仓库

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git remote add origin https://github.com/xty398305-del/learn01.git
```

图 21: 远程连接

2.18 上传文件

连接到远程仓库后，可以把本地文件上传到自己的仓库。此时 GitHub 会验证身份是否为本人，通过 token 验证后，把本地仓库中的文件上传到远程仓库

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git push origin master
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 16 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (14/14), 1.10 KiB | 281.00 KiB/s, done.
Total 14 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/xty398305-del/learn01.git
 * [new branch]      master -> master
```

图 22: 上传文件

在 GitHub 上查看，文件上传成功

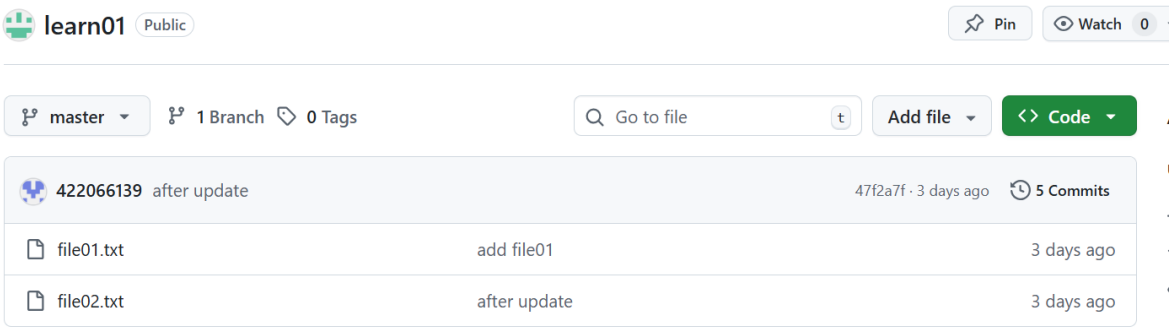

图 23: GitHub 查看上传文件

2.19 拉取文件

此时，我在远程仓库对项目做了修改，增加了文件 main.cpp

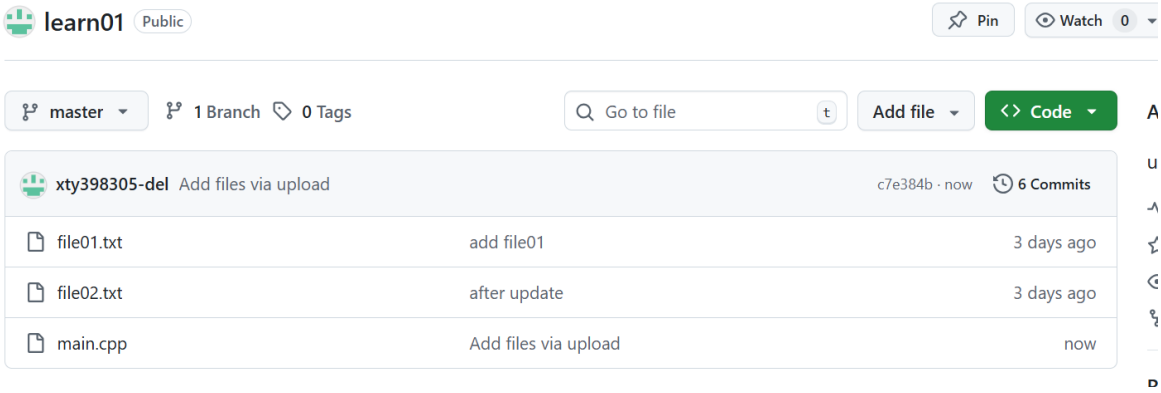

图 24: GitHub 上修改项目

通过终端命令，把远程仓库的更新拉取到本地仓库

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 3.81 KiB | 650.00 KiB/s, done.
From https://github.com/xty398305-del/learn01
 * branch            master     -> FETCH_HEAD
   47f2a7f..c7e384b  master     -> origin/master
Updating 47f2a7f..c7e384b
Fast-forward
 main.cpp | 315 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 1 file changed, 315 insertions(+)
 create mode 100644 main.cpp
```

图 25: 拉取远程仓库更新

拉取成功，可以看到 main.cpp 已经成功更新到本地仓库

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ ls
file01.txt  file02.txt  main.cpp
```

图 26: 拉取结果查看

2.20 克隆远程项目

此时，我在 GitHub 中创建了一个新的仓库，并添加了一些文件，如下图所示

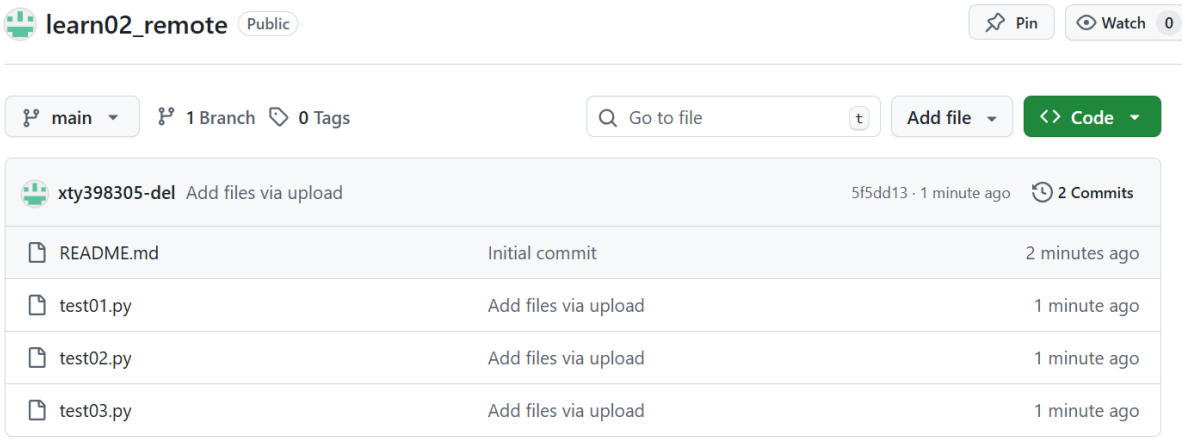

图 27: 远程仓库项目

通过 clone 命令，在本地仓库克隆一个远程仓库副本

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ git clone https://github.com/xty398305-del/learn02_remote.git
Cloning into 'learn02_remote'...
remote: Enumerating objects: 8, done.
remote: Counting objects: 100% (8/8), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (8/8), done.
```

图 28: 克隆远程仓库

查看克隆结果，可以看到，克隆成功

```
24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ ls
file01.txt  file02.txt  learn02_remote/  main.cpp

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository (master)
$ cd learn02_remote

24695@LAPTOP-JQI687BK MINGW64 /d/gitRepository/learn02_remote (main)
$ ls
README.md  test01.py  test02.py  test03.py
```

图 29: 克隆成功

3 LaTeX

3.1 LaTeX 中的文章结构

LaTeX 中的文章分为导言区和正文区

导言区：

`\documentclass{article}`

使用 documentclass 引入文档类，此外还有 book 类，report 类，letter 类

标题：

`\title{my first article}`

作者：

`\author{tantan}`

日期：

`\date{\today}`

还有一些宏包也都是在导言区

正文区：

`\begin{document}`

内容

`\end{document}`

用 begin 和 end 输入一个环境名称，我们设定为 document，注意一个 latex 文件只能有一个 document 环境

我们使用`\maketitle`输出整个标题

3.2 LaTeX 中使用中文

为了在 LaTeX 中使用中文，我们需要在导言区引入一个宏包`\usepackage{ctex}`

还有另一种方法，就是在导言区只写`\documentclass{ctexart}`，对应的其他类为`ctexbook ctexrep`，注意没有 letter 对应的

3.3 LaTeX 中的数学公式

在 LaTeX 中，我们使用`$`符号进入数学模式，举个例子：

我们使用`$ 1 + 1 = 2 $`来输出这个简单的数学公式

此外还有`$$`符号

要注意的是，两者的差别在于`$`符号表示行内公式，不会换行，`$$`符号表示行间公式，会换换行

此外，如果我们想产生自动带编号的行间公式，就要使用 equation 环境

`\begin{equation}`

`\end{equation}`

下面我使用这个环境写几个自动带编号的数学公式

$$\sin^2 x + \cos^2 x = 1 \tag{1}$$

$$a + b = b + a \tag{2}$$

此外，很多数学公式的实现还需要引入 amsmath 这个宏包

3.4 LaTeX 中的字体

在 latex 中，一个字体有五种属性：

字体编码 1. 正文字体编码：OT1,T1,EU1 等 2. 数字字体编码:OML,OMS,OMX 等

字体族 1. 罗马字体：笔画起始处有装饰 2. 无衬线字体：笔画起始处无装饰 3. 打字机字体：每个字符宽度相同，又称等宽字体

字体系列 1. 粗细 2. 宽度

字体形状 1. 直立 2. 斜体 3. 伪斜体 4. 小型大写

字体大小

这里我们主要讨论中文

使用`\songti`让后续输出的字体为宋体

使用`\heiti` 让后续输出的字体为黑体

使用`\fangsong`让后续输出的字体为仿宋

使用`\kaishu`让后续输出的字体为楷书

使用`\texbf`让后续输出的字体为粗体

使用`\textit`让后续输出的字体为斜体

使用`\zihao{}` 来限定字体大小，括号里面填数字

3.5 LaTeX 中表格的生成

在 LaTeX 中，我们使用 tabular 环境来生成表格，举个例子：下面这段代码就可以生成这样的一个表格

姓名	语文	数学	外语	备注
张三	87	100	93	优秀
李四	75	64	52	补考另行通知
王二	80	82	78	

3.6 LaTeX 中插入图片

在 LaTeX 中插入图片需要使用 `\usepackage{graphicx}`这个宏包，同时插入图片需要和你的 txt 文件在一个文件夹下

然后在正文区使用`\includegraphics{`图片文件名`}`即可

3.7 LaTeX 中创建章节和子章节

在 LaTeX 中，创建章节一般要用到`\section{}`和`\subsection{}`，`\subsubsection{}`下面是一个简单例子的代码：

```
\section{引言}
\section{实验方法}
\section{实验结果}
\subsection{数据}
\subsection{图表}
\subsubsection{实验条件}
\subsubsection{实验过程}
```

\subsection{结果分析}
\section{结论}
\section{致谢}
实现效果为:

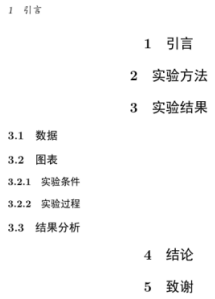

图 30: 章节

3.8 LaTeX 中目录的生成

在 LaTeX 中为了自动生成目录，我们需要在正文区添加\tableofcontents，通常，为了让目录和后面的正文内容不在同一页，我们可以加上\newpage来实现，\newpage的作用是另起一页

3.9 LaTeX 中书写参考文献

在论文的最后，我们需要书写参考文献，在知网上导出我们需要的参考文献，注意选择 BibTeX 格式,然后新建一个文件,复制进去,注意后缀最好是.bib,在导言区引入\bibliographystyle{plain}宏包,然后在最后的正文区加上\nocite{*}和\bibliography{刚才新建的后缀为.bib的文件文件名}，这里\notice{*}的作用是显示我们刚才创建的后缀为.bib 文件里面的所以内容，也可以不使用这个命令，每次只添加某条文献。

3.10 LaTeX 中一些特殊字体的书写

比较常见的有:
\TeX{}输出 TEX
\LaTeX{}输出 LATEX
\LaTeXe{}输出LATEX 2_ε
\XeLaTeX输出, 不过这个需要在导言区添加\xltxtra宏包
`表示左单引号 ‘
'表示右单引号’

4 实验总结

通过系统学习 Git 与 LaTeX，我的技能树得到了丰富与拓展。这个过程像一次全面的技能洗礼：掌握了版本控制的高效协作，也学会了排版的精确与美感。学习中难免遇到合并冲突、宏包调试、图片与公式排布等挑战，但我通过上网检索、阅读文档，以及同学的帮助，逐步找到解决思路，提升了问题解决与合作效率。这些经历让我体会到坚持探索、勇于求助、勤于实践的价值，并将信

心转化为实际能力。未来我将更专注地提升在分支管理、协同工作流，以及 LaTeX 排版美学和宏包应用方面的水平，力求写作与表达同样清晰优雅，不断追求更高的专业成果。

5 代码、练习过程查看链接

本次报告相关代码、练习均可以在 GitHub 上查看，链接为：https://github.com/xty398305-del/learn02_remote