

系统开发工具基础第一次实验报告

张烨 23020007162

2024 年 8 月 31 日

目录

| | |
|---|----|
| 一、 实验目的 | 3 |
| 二、 实验内容 | 3 |
| (一) Git | 3 |
| 1. 创建本地空仓库 | 3 |
| 2. 添加和提交文件 | 4 |
| 3. 查看提交历史 | 5 |
| 4. 给 <code>git log --oneline --graph --decorate</code> 起别名 <code>graph</code> | 7 |
| 5. 将本地仓库与远程仓库连接起来 | 8 |
| 6. 将本地文件添加进远程仓库 | 9 |
| 7. 克隆 GitHub 上的仓库 | 10 |
| 8. 修改远程仓库上的内容 | 12 |
| 9. 删除文件 | 13 |
| 10. <code>git reset</code> 回退版本的三种命令 | 14 |
| (二) LaTeX | 16 |
| 1. LaTeX 中的文章结构 | 16 |
| 2. LaTeX 中使用中文 | 17 |
| 3. LaTeX 中的数学公式 | 17 |
| 4. LaTeX 中的字体 | 18 |
| 5. LaTeX 中表格的生成 | 18 |
| 6. LaTeX 中插入图片 | 19 |
| 7. LaTeX 中创建章节和子章节 | 20 |

| | |
|--------------------------------|----|
| 目录 | 2 |
| 8. LaTeX 中目录的生成 | 20 |
| 9. LaTeX 中书写参考文献 | 21 |
| 10. LaTeX 中一些特殊字体的书写 | 21 |
| 三、 心得体会 | 21 |
| 四、 相关练习、报告和代码查看链接 | 22 |

一、实验目的

1. 学习版本控制 (Git)

通过学习课程资料和课下的一些自主学习，能够掌握基本的 Git 指令，通过不断学习，继续强化。

2. 学习 LaTeX 文档编辑

学习 LaTeX 文档编辑器，掌握 LaTeX 的排版方法，能够用 LaTeX 来生成自己的实验报告。

二、实验内容

(一) Git

1. 创建本地空仓库

创建本地仓库的条件是需要一个空目录，然后在空目录中初始化你的项目

比如说我想要创建一个名为”mytest”的空项目：

1. 创建目录

```
ZY@LAPTOP-HLMQDURI MINGW64 ~ (master)
$ mkdir mytest
```

图 1: mkdir

2. 进入目录

```
ZY@LAPTOP-HLMQDURI MINGW64 ~ (master)
$ cd mytest
```

图 2: cd

3. 使用 git init 初始化当前仓库

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/mytest (master)
$ git init
Initialized empty Git repository in C:/Users/ZY/mytest/.git/
```

图 3: git init

2. 添加和提交文件

首先我们要知道 Git 的本地数据管理分为三个区域，分别是工作区，暂存区和本地仓库，而文件只有被放进暂存区才能通过 `git commit` 添加进仓库

1.git status

用于显示当前 git 仓库的状态信息

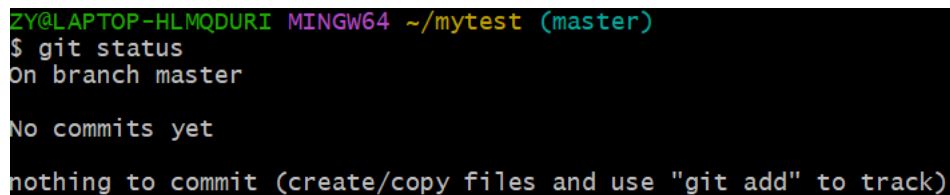
例如：

工作目录和暂存区的修改情况

哪些文件已被修改但尚未暂存

哪些文件已被暂存但尚未提交

当前分支的状态和与远程分支的差异



```
ZY@LAPTOP-HLMQDURI MINGW64 ~/mytest (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

图 4: git status

2.echo

在此我们使用 `echo` 创建文件并写入内容

3.git add 该命令用于将文件添加到暂存区

常见的有：

`git add 文件名 + 文件类型`

eg: `git add file1.txt`

`git add *.txt` 当文件较多时这个命令可以把所有的 `txt` 文件都放入暂存区

`git add .` 这个可以把所有文件都放入暂存区

4.git commit/git commit -m

`git commit -m 描述`，`-m` 后面的内容是提交的信息用于描述这次提交的目的或内容

eg: `git commit -m "这是第一次提交"`

当然也可以不用-m 直接用 git commit 这时候就会跳转到一个页面先输入 i 才能进入编辑模式然后可以写这是第一次提交然后点 Esc 输入:wq 就可以保存退出了

```
CYBLAPTOP-HLMQDURI MINGW64 ~/mytest (master)
$ echo file1 > file1.txt

CYBLAPTOP-HLMQDURI MINGW64 ~/mytest (master)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

CYBLAPTOP-HLMQDURI MINGW64 ~/mytest (master)
$ git commit -m "这是第一次提交"
[master (root-commit) e193f58] 这是第一次提交
1 file changed, 1 insertion(+)
create mode 100644 file1.txt
```

图 5: 关于文件

3. 查看提交历史

在此，多写几个文件便于查看

```
CYBLAPTOP-HLMQDURI MINGW64 ~ (master)
$ cd learn

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git init
Reinitialized existing Git repository in C:/Users/ZY/learn/.git/

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ echo 111 > file1.txt

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git commit -m "这是第一次提交"
[master (root-commit) 0dad109] 这是第一次提交
1 file changed, 1 insertion(+)
create mode 100644 file1.txt

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ echo 222 > file2.txt

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git add .
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time Git touches it

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git commit -m "这是第二次提交"
[master 7274eal] 这是第二次提交
1 file changed, 1 insertion(+)
create mode 100644 file2.txt

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ echo 333 > file3.txt

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git add .
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time Git touches it

CYBLAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git commit -m "这是第三次提交"
[master 15c4d0f] 这是第三次提交
1 file changed, 1 insertion(+)
create mode 100644 file3.txt
```

图 6: 添加文件

1.git log 是一个用于查看 Git 仓库提交历史的命令。运行它会列出所有的提交记录，从最新的提交开始，显示每个提交的哈希值、作者、日期和提交信息

```
$ git log
commit 15c4d0fc5e4aa3c228546e4c16217e0820a96b51 (HEAD -> master)
Author: Ye Zhang <i960567770@qq.com>
Date: Sat Aug 31 16:09:36 2024 +0800

    这是第三次提交

commit 7274ea15eb4b209dbff9a65eea5d9decfb84637
Author: Ye Zhang <i960567770@qq.com>
Date: Sat Aug 31 16:09:11 2024 +0800

    这是第二次提交

commit 0dad109f0b63e1ed2ec804556951d125779a4d91
Author: Ye Zhang <i960567770@qq.com>
Date: Sat Aug 31 16:08:39 2024 +0800

    这是第一次提交
```

图 7: git log

2. 其他:

git log --oneline: 以简洁的单行格式显示每个提交

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git log --oneline
15c4d0f (HEAD -> master) 这是第三次提交
7274ea1 这是第二次提交
0dad109 这是第一次提交
```

图 8: git log --oneline

git log --graph: 以图形化形式展示分支和合并历史

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git log --graph
* commit 15c4d0fc5e4aa3c228546e4c16217e0820a96b51 (HEAD -> master)
  Author: Ye Zhang <i960567770@qq.com>
  Date: Sat Aug 31 16:09:36 2024 +0800

    这是第三次提交

* commit 7274ea15eb4b209dbff9a65eea5d9decfb84637
  Author: Ye Zhang <i960567770@qq.com>
  Date: Sat Aug 31 16:09:11 2024 +0800

    这是第二次提交

* commit 0dad109f0b63e1ed2ec804556951d125779a4d91
  Author: Ye Zhang <i960567770@qq.com>
  Date: Sat Aug 31 16:08:39 2024 +0800

    这是第一次提交
```

图 9: git log --graph

git log -p: 显示每个提交的差异

```

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git log -p
commit 15c4d0fc5e4aa3c228546e4c16217e0820a96b51 (HEAD -> master)
Author: Ye Zhang <1960567770@qq.com>
Date: Sat Aug 31 16:09:36 2024 +0800

    这是第三次提交

diff --git a/file3.txt b/file3.txt
new file mode 100644
index 0000000..55bd0ac
--- /dev/null
+++ b/file3.txt
@@ -0,0 +1 @@
+333

commit 7274ea15eb4b209dbff9a65eea5d9decfc84637
Author: Ye Zhang <1960567770@qq.com>
Date: Sat Aug 31 16:09:11 2024 +0800

    这是第二次提交

diff --git a/file2.txt b/file2.txt
new file mode 100644
index 0000000..c200906

```

图 10: git log -p

git log -n : 只显示最近的 n 个提交

```

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git log -2
commit 15c4d0fc5e4aa3c228546e4c16217e0820a96b51 (HEAD -> master)
Author: Ye Zhang <1960567770@qq.com>
Date: Sat Aug 31 16:09:36 2024 +0800

    这是第三次提交

commit 7274ea15eb4b209dbff9a65eea5d9decfc84637
Author: Ye Zhang <1960567770@qq.com>
Date: Sat Aug 31 16:09:11 2024 +0800

    这是第二次提交

```

图 11: git log -n

git log --oneline --graph --decorate: 以简洁的格式和图形化展示提交历史, 并包含分支和标签信息

```

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git log --oneline --graph --decorate
* 15c4d0f (HEAD -> master) 这是第三次提交
* 7274ea1 这是第二次提交
* 0dad109 这是第一次提交

```

图 12: git log --oneline --graph --decorate

4. 给 git log --oneline --graph --decorate 起别名 graph
打开.gitconfig,

```
ZY@LAPTOP-HLMQDURI MINGW64 ~ (master)
$ vim .gitconfig
```

图 13: 打开

添加:

```
[alias] graph = log --all --graph --decorate --oneline
```

```
[user]
  name = Ye Zhang
  email = 1960567770@qq.com
[credential]
  helper = store
[alias]
  graph = log --all --graph --decorate --oneline
```

图 14: graph

5. 将本地仓库与远程仓库连接起来

首先我们在 GitHub 上创建一个仓库

Top repositories

New

Find a repository...

图 15: 创建仓库

然后使用 `git remote add origin HTTPS/SSH`, 在此, 我使用的是 SSH

Quick setup — if you've done this kind of thing before



Set up in Desktop

or

HTTPS

SSH

git@github.com:zhangtantan77/zy.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include

图 16: 复制地址


```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git remote add origin git@github.com:zhangtatan77/zy.git
```

图 17: 连接

6. 将本地文件添加进远程仓库

在 learn 仓库中, 之前已经添加了三个文件, 我们使用 ls 命令查看

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ ls
file1.txt file2.txt file3.txt
```

图 18: 显示文件和目录的名称

我们使用 git push 命令将这三个文件添加进远程仓库

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git push -u origin master
Enter passphrase for key '/c/Users/ZY/.ssh/id_rsa':
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 20 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (9/9), 733 bytes | 733.00 KiB/s, done.
Total 9 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:zhangtatan77/zy.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

图 19: 添加进远程仓库

然后刷新一下, 就可以看到仓库中出现了这三个文件

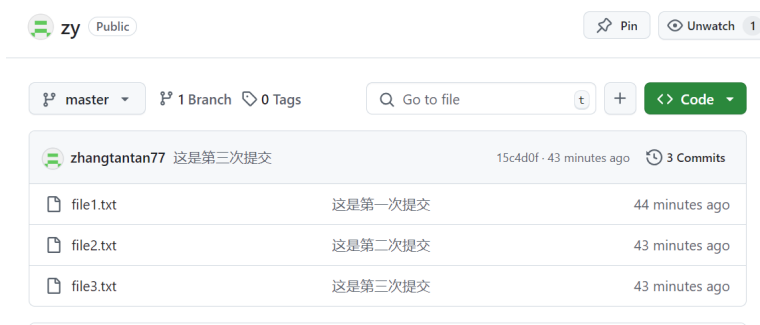


图 20: 仓库

7. 克隆 GitHub 上的仓库

在此我克隆的是之前自己在 GitHub 上创建过的一个仓库 remote-repo

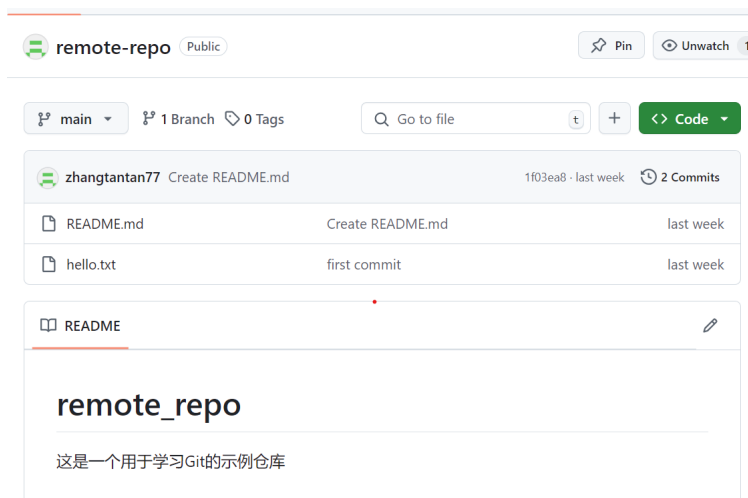


图 21: remote-repo 仓库

点击绿色的 code，然后复制 SSHSH

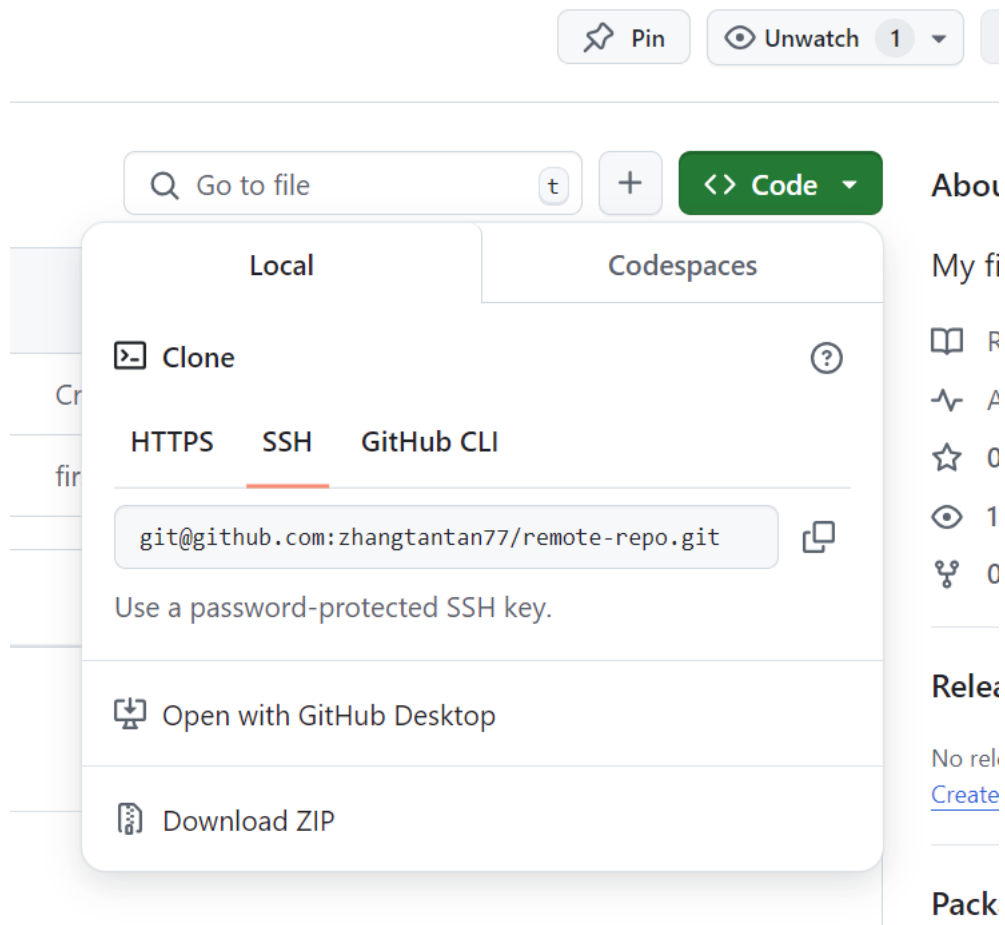


图 22: 复制

git clone SSH

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git clone git@github.com:zhangtantan77/remote-repo.git
Cloning into 'remote-repo'...
Enter passphrase for key '/c/Users/ZY/.ssh/id_rsa':
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), done.
```

图 23: 克隆

因为我之前设置过 SSH, 所以需要输入密码
打开 learn 仓库所在目录, 就可以看见 learn 目录下除了之前的三个文件, 还有 remote-repo

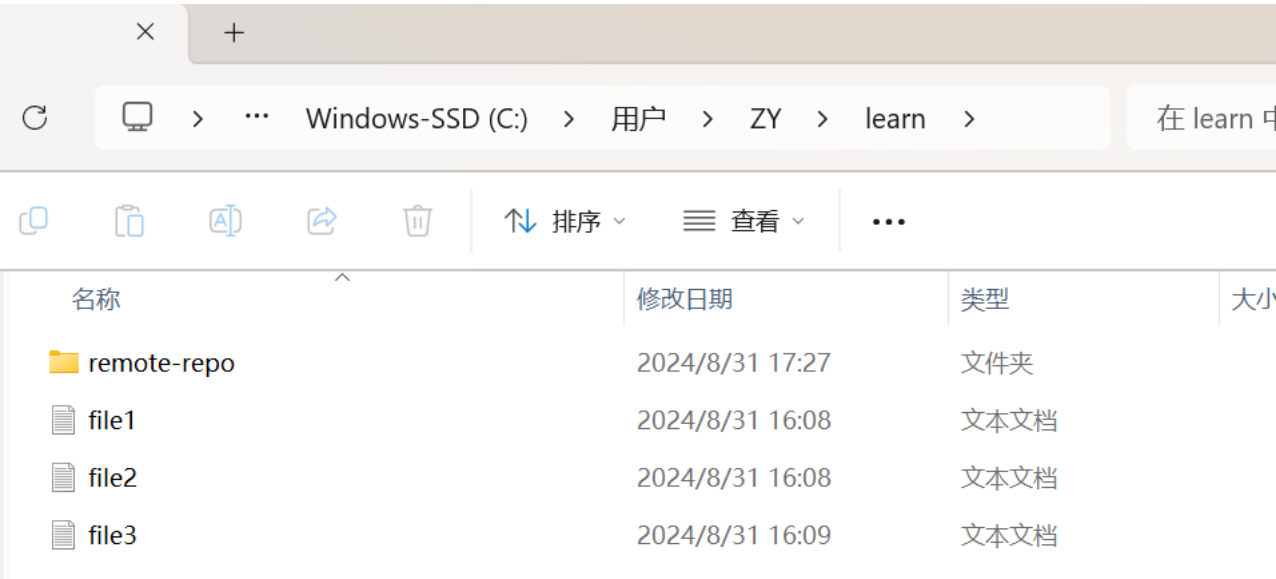


图 24: 成功

8. 修改远程仓库上的内容

现在我们要修改 remote-repo 中的内容, 先前可以看到, remote-repo 中有 hello.txt 文件, 打开 hello.txt 可以看到原来的内容是 hello

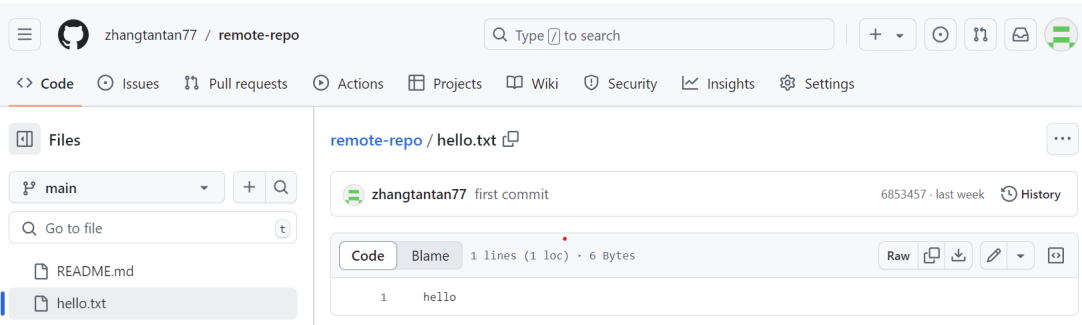


图 25: hello

现在我们把原来的内容替换成 11111

首先 cd remote-repo, 进入当前目录

然后使用 echo 命令修改内容

再使用 git add 和 git commit 命令, 最后使用 git push 命令推送到远程仓库

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ ls
file1.txt file2.txt file3.txt hello.txt remote-repo
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ ls
file1.txt file2.txt file3.txt hello.txt remote-repo/
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ cd remote-repo
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn/remote-repo (main)
$ echo 11111 > hello.txt
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn/remote-repo (main)
$ git add .
warning: in the working copy of 'hello.txt', LF will be replaced by CRLF the next time Git touches it
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn/remote-repo (main)
$ git commit -m "修改内容"
[main 32fee26] 修改内容
1 file changed, 1 insertion(+), 1 deletion(-)
ZY@LAPTOP-HLMQDURI MINGW64 ~/learn/remote-repo (main)
$ git push
Enter passphrase for key '/c/Users/ZY/.ssh/id_rsa':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:zhangtantan77/remote-repo.git
1f03ea8..32fee26 main -> main
```

图 26: 修改

任何刷新一下就可以看到修改成功

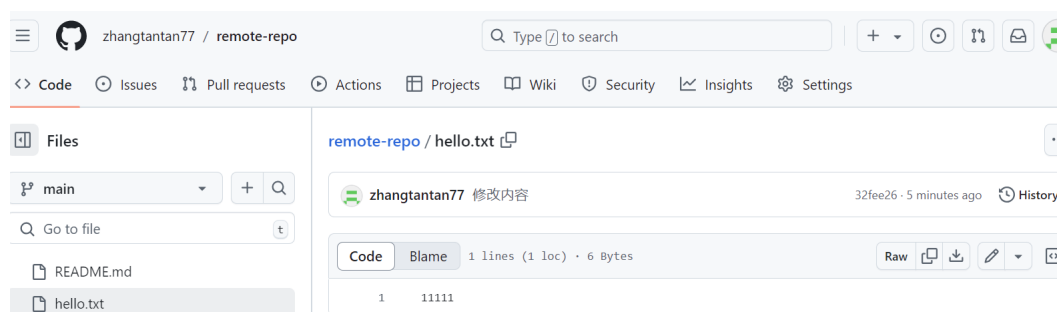


图 27: 成功

9. 删除文件

git rm 文件

该命令把文件从工作区和暂存区同时删除，要注意提交，否则删除的文件在版本库中还是存在的，在这里我们删除 file1.txt

```

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ ls
file1.txt file2.txt file3.txt hello.txt remote-repo/

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git rm file1.txt
rm 'file1.txt'

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        deleted:   file1.txt

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:  remote-repo (new commits)

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ ls
file2.txt file3.txt hello.txt remote-repo/

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git ls-files
file2.txt
file3.txt
hello.txt
remote-repo

ZY@LAPTOP-HLMQDURI MINGW64 ~/learn (master)
$ git commit -m "删除了文件1"
[master dd55577] 删除了文件1
1 file changed, 1 deletion(-)
delete mode 100644 file1.txt

```

图 28: 删除文件

10.git reset 回退版本的三种命令

在此新建一个仓库 repo, 并分三次添加三个文件

```

ZY@LAPTOP-HLMQDURI MINGW64 ~ (master)
$ git init learnlearn
Initialized empty Git repository in C:/Users/ZY/learnlearn/.git/

ZY@LAPTOP-HLMQDURI MINGW64 ~ (master)
$ cd learnlearn

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ git init repo
Initialized empty Git repository in C:/Users/ZY/learnlearn/repo/.git/

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cd repo

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ echo 111 > file1.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ echo 222 > file2.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ echo 333 > file3.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ git add file1.txt
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time git touches it

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ git commit -m "commit1"
[master (root-commit) 18a54f7] commit1
1 file changed, 1 insertion(+)
create mode 100644 file1.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ git add file2.txt
warning: in the working copy of 'file2.txt', LF will be replaced by CRLF the next time git touches it

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ git commit -m "commit2"
[master 7168ac7] commit2
1 file changed, 1 insertion(+)
create mode 100644 file2.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ git add file3.txt
warning: in the working copy of 'file3.txt', LF will be replaced by CRLF the next time git touches it

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo (master)
$ git commit -m "commit3"
[master 3161d41] commit3
1 file changed, 1 insertion(+)
create mode 100644 file3.txt

```

图 29: 新建

为了方便比较，将仓库目录复制三份

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cp -rf repo repo-soft

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cp -rf repo repo-hard

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cp -rf repo repo-mixed
```

图 30: 复制

1.git reset --soft 表示回退到某一个版本，并且保留工作区和暂存区的所有修改内容

```
ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cd repo-soft

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ git reset --soft 7569ac7

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ git log --oneline
7569ac7 (HEAD -> master) commit2
184b54f commit1

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ ls
file1.txt file2.txt file3.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ cat file3.txt
333

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ git ls-file
git: 'ls-file' is not a git command. See 'git --help'.

The most similar command is
ls-files

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ git ls-files
file1.txt
file2.txt
file3.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-soft (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   file3.txt
```

图 31: soft

2.git reset --hard 表示回退到某一个版本，并且丢弃工作区和暂存区的所有修改内容

```

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cd repo-hard

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-hard (master)
$ git reset --hard HEAD^
HEAD is now at 7569ac7 commit2

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-hard (master)
$ git log --oneline
7569ac7 (HEAD -> master) commit2
184b54f commit1

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-hard (master)
$ ls
file1.txt  file2.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-hard (master)
$ git ls-files
file1.txt
file2.txt

```

图 32: hard

3.git reset --mixed 表示回退到某一个版本，并且只保存工作区的修改内容

```

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn (master)
$ cd repo-mixed

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-mixed (master)
$ git reset HEAD^

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-mixed (master)
$ git log --oneline
7569ac7 (HEAD -> master) commit2
184b54f commit1

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-mixed (master)
$ ls
file1.txt  file2.txt  file3.txt

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-mixed (master)
$ cat file3.txt
333

ZY@LAPTOP-HLMQDURI MINGW64 ~/learnlearn/repo-mixed (master)
$ git ls-files
file1.txt
file2.txt

```

图 33: mixed

(二) LaTeX

1.LaTeX 中的文章结构

LaTeX 中的文章分为引言区和正文区

引言区：

```
\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

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. 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

$$\sin^2 x + \cos^2 x = 1 \quad (1)$$

$$a + b = b + a \quad (2)$$

此外, 很多数学公式的实现还需要引入 `amsmath` 这个宏包, 实现方法是在导言区添加 `\usepackage{amsmath}`

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

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

```
\begin{tabular}{l | c | c | c | p{1.5cm}}%两个就是双 可以用p{}指定列的宽度，当内容超过时，会自动
换行
\hline
姓名 & 语文 & 数学 & 外语 & 备注 \\
\hline \hline
张三 & 87 & 100 & 93 & 优秀 \\
\hline
李四 & 75 & 64 & 52 & 补考另行通知 \\
\hline
王二 & 80 & 82 & 78 & \\
\hline
\end{tabular}
```

图 34: 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 | |

这样的表格

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，我的图片都在 figures 文件夹中，那么就可以添加上 `\graphicspath{{figures/}}` 指定搜索目录

然后在正文区使用 `\includegraphics{图片文件名}` 就好了，也可以使用 `\includegraphics[缩放比例]{图片文件名}`，在 [] 里面加入缩放比例或者指定的长宽高

下面是相关代码：

```
\documentclass{ctexart}

\usepackage{graphicx}
\graphicspath{{figures/}}

\begin{document}
  \includegraphics{first}
  \includegraphics{second}

  \includegraphics[scale=0.3]{first}
  \includegraphics[scale=0.03]{second}

  \includegraphics[height=2cm]{first}
  \includegraphics[height=2cm]{second}

  \includegraphics[width=2cm]{first}
  \includegraphics[width=2cm]{second}

  \includegraphics[height=0.1\textheight]{first}
  \includegraphics[height=0.1\textheight]{second}

  \includegraphics[width=0.2\textwidth]{first}
  \includegraphics[width=0.2\textwidth]{second}

  \includegraphics[angle=-45,width=0.2\textwidth]{first}
  \includegraphics[angle=45,width=0.2\textwidth]{second}

\end{document}
```

图 35: 插图

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:
```

| | |
|-------|------|
| 1 | 引言 |
| 2 | 实验方法 |
| 3 | 实验结果 |
| 3.1 | 数据 |
| 3.2 | 图表 |
| 3.2.1 | 实验条件 |
| 3.2.2 | 实验过程 |
| 3.3 | 结果分析 |
| 4 | 结论 |
| 5 | 致谢 |

图 36: 章节

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

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\documentclass[plain]{article}宏包，然后在最后的正文区加上\usepackage{bibliography}和\usepackage{bibliography}，这里\usepackage{bibliography}的作用是显示我们刚才创建的后缀为.bib 文件里面的内容，也可以不使用这个命令，每次只添加某条文献，但我感觉这种方法更简单一点，下面我来导入刚才的三个文献：

参考文献

- [1] 余筱琴. 数学建模视角下的高中化学解题策略研究. 西藏教育, (08):38–42, 2024.
- [2] 王瑞雪, 王雪, 周果清, 肖照林, and 王庆. 基于梯度下降深度均衡模型的动态光场重建（特邀）. 激光与光电子学进展, 61(16):93–100, 2024.
- [3] 郭威帅, 郭锐, 孙宏浩, and 安然. 装配阶段发动机导管二次设计方法研究. 机械设计与制造, pages 1–6.

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 \LaTeXe

\XeLaTeX 输出 \XeLaTeX , 不过这个需要在导言区添加\usepackage{xetex}宏包

`表示左单引号 ‘

’表示右单引号’

三、心得体会

通过学习 Git 和 LaTeX，进一步丰富了自己的技能，同时，在学习过程中也遇到很多问题，但是通过上网查询和同学帮助，最后终于解决了问题。在今后的学习中，我会更加认真，继续提升自己。

四、相关练习、报告和代码查看链接

本次报告相关练习、报告和代码均可以在 <https://kkgithub.com/zhangtatan77/work> 查看