

# 实验报告

——Git 与 Latex 的学习与使用

姓名：翟一航

学号：23020011046

班级：23级软件工程五八班

## 1 实验要求

### 1.1 学习版本控制 Git 的使用

### 1.2 完成 4 个课堂练习与 20 个与 Git 和 Latex 有关的实例

### 1.3 学习用 Latex 进行文档编辑，并设计自己的实验报告模板

## 2 实验内容

### 2.1 Git (Bash) 的学习

2.1.1 版本控制系统 (VCSs) 是一类用于追踪源代码 (或其他文件、文件夹) 改动的工具。顾名思义, 这些工具可以帮助我们管理代码的修改历史; 不仅如此, 它还可以让协作编码变得更方便。VCS 通过一系列的快照将某个文件夹及其内容保存了起来, 每个快照都包含了文件或文件夹的完整状态。同时它还维护了快照创建者的信息以及每个快照的相关信息等等。

2.1.2 Git 包含以下特点:

1. 分布式存储
2. 数据完整性
3. 分支管理
4. 非线性开发
5. 安全性
6. 轻量级操作
7. 强大的合并和冲突解决功能

2.1.3 Git 的基本工作流程通常包括以下几个步骤:

1. 克隆 (Clone)
2. 修改 (Modify)
3. 提交 (Commit)
4. 推送 (Push)
5. 拉取 (Pull)

接下来也将围绕这些操作来完成课堂任务与实例应用

### 2.2 Latex 的学习

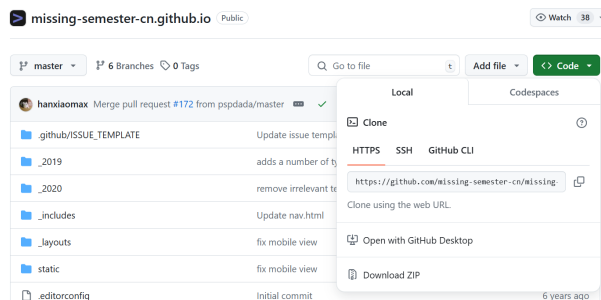
2.2.1 Latex 是一个基于 TeX 的排版系统, 广泛用于生成科学和数学文档。它特别适合于生成包含复杂公式、表格和参考文献的文档。

2.2.2 为了更好地学习与使用 Latex 来进行文档的编写, 我首先根据教程在电脑上安装了 Latex Live, 配置了 Latex 的使用环境, 接着下载了专门用于撰写 Latex 文档的工具 Texstudio。

2.2.3 通过在网上查阅资料、浏览教学视频, 我学会了 Latex 的一些基本用法, 如: 设置文档的标题以及文段的格式、插入图片与数学公式、加入拓展的宏和包等, 并将这些内容应用在了此次实验报告撰写中。

首先，在文件资源管理器中找到合适的位置创建一个文件夹，来储存克隆下来的仓库，在 Git 中先通过 `cd` 指令进入相应磁盘中的文件夹。

然后，打开 GitHub 仓库，复制仓库的链接（如下图所示）



之后，在 Git 中输入指令 `git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io.git`（链接为刚刚复制而来），接着就会进行仓库的克隆，会显示克隆进度

```
19568@DESKTOP-2800C1Q MINGW64 ~/repos (main)
$ git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io.git
Cloning into 'missing-semester-cn.github.io'...
fatal: unable to access 'https://github.com/missing-semester-cn/missing-semester-cn.github.io.git/': Failed to connect to github.com port 443 after 21087 ms: Could not connect to server

19568@DESKTOP-2800C1Q MINGW64 ~/repos (main)
$ git config --global --unset http.proxy

19568@DESKTOP-2800C1Q MINGW64 ~/repos (main)
$ git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io.git
Cloning into 'missing-semester-cn.github.io'...
fatal: unable to access 'https://github.com/missing-semester-cn/missing-semester-cn.github.io.git/': Failed to connect to github.com port 443 after 21112 ms: Could not connect to server

19568@DESKTOP-2800C1Q MINGW64 ~/repos (main)
$ git clone https://github.com/missing-semester-cn/missing-semester-cn.github.io.git
Cloning into 'missing-semester-cn.github.io'...
remote: Enumerating objects: 3194, done.
remote: Counting objects: 100% (3194/3194), done.
remote: Compressing objects: 100% (1126/1126), done.
remote: Total 3194 (delta 2040), reused 2735 (delta 2033), pack-reused 0 (from 0)
Receiving objects: 100% (3194/3194), 15.44 MiB | 5.10 MiB/s, done.
Resolving deltas: 100% (2040/2040), done.
```

最后，该进行版本历史可视化的探索。在 Git 中输入指令 `git log --all --graph --decorate`，会出现如下界面，版本可视化探索完成。

```
LENOVO@DESKTOP-00A7G4V MINGW64 ~
$ cd d:

LENOVO@DESKTOP-00A7G4V MINGW64 /d
$ cd test1
bash: cd: test1: No such file or directory

LENOVO@DESKTOP-00A7G4V MINGW64 /d
$ cd git/test1

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1
$ ls
missing-semester-cn.github.io/

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1
$ cd missing-semester-cn.github.io/

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git log
```

```

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git log --all --graph --decorate
* commit f956db4d10d688dbfa85e0497d8ca593c9a529bf (HEAD -> master)
| Author: zhaiyihang <2905271851@qq.com>
| Date:   Fri Aug 23 11:00:49 2024 +0800
|
|     add password123 to file
|
| * commit af054fa1aea2f2599e4474d96b63f73dd9bd145f (origin/master, origin/HEAD)
| \ Merge: dd3f3dd 9baa48c
|   Author: Lingfeng_Ai <hanxiaomax@gmail.com>
|   Date:   Fri Aug 16 06:54:16 2024 +0800
|
|       Merge pull request #172 from pspdada/master
|
|       Thank you so much
|
| * commit 9baa48c778012164179e4e60725418941f41743b
|   Author: psp_dada <1824427006@qq.com>
|   Date:   Thu Aug 15 02:07:36 2024 +0800
|
|       remove irrelevant text
|
| * commit f5df7de89dc7712483665cc6fe8a787aafbef9bf
|   Author: psp_dada <1824427006@qq.com>
|   Date:   Thu Aug 15 01:46:12 2024 +0800
|
|       fix wrong index
|
| * commit ef9a2f75409ff7746c03f6233066e3d2c634cd12
| / Author: psp_dada <1824427006@qq.com>
|   Date:   Thu Aug 15 01:32:44 2024 +0800
|
|       fix typo
|
| * commit dd3f3dd37ce88bb78f802a425e652a65d2ce70d5
| \ Merge: d284d3e 8e26b4a
|   Author: Lingfeng_Ai <hanxiaomax@gmail.com>
|   Date:   Thu Aug 1 17:58:28 2024 +0800
|
|       Merge pull request #171 from HowieChih/for-better-understanding
|
|       翻译优化: For better understanding
|
| * commit 8e26b4a0242ddaa6e239d314edbca8cb61e27cd3
|   Author: Howie Chih <howiechih@gmail.com>
|   Date:   Thu Aug 1 17:00:37 2024 +0800
|
|       更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译
|
| * commit 4e2ff4369c0c6aca74bdfd6ab12a0b397f6a9b11
| / Author: Howie chih <howiechih@gmail.com>
|   Date:   Thu Aug 1 14:54:53 2024 +0800
|
|       更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译

```

### 2.3.2 查看是谁最后修改了 README.md 文件

在 Git 中输入指令 `git log -1 README.md`，其中，`-1` 表示查看最新的 1 次提交或特定文件的版本信息（结果如下图所示）。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git log -1 README.md
commit de98852ef0604cf918bab7f39c63a53932c845d8
Author: yuzq <yuzq@sunwayworld.com>
Date: Thu Jun 6 14:43:07 2024 +0800
```

将readme文件中的url的绝对路径改为相对路径，不用重复访问github，利于分享传播

### 2.3.3 查看最后一次修改 \_\_config.yml 文件中 collections: 行时的提交信息是什么

在 Git 中依次输入指令：git blame \_\_config.yml | grep collections、git show -pretty=format:"%s"  
a88b4eac | head -1 出现以下界面即为查看成功。

```

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git blame _config.yml | grep collections
a88b4eac (Anish Athalye 2020-01-17 15:26:30 -0500 18) collections:

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git show a88b4eac
commit a88b4eac326483e29bdac5ee0a39b180948ae7fc
Author: Anish Athalye <me@anishathalye.com>
Date: Fri Jan 17 15:26:30 2020 -0500

    Redo lectures as a collection

diff --git a/2020/index.html b/2020/index.html
deleted file mode 100644
index 153ddc8..0000000
--- a/2020/index.html
+++ /dev/null
@@ -1,5 +0,0 @@
-----
-layout: redirect
-redirect: /
-title: Missing Semester 2020
-----
diff --git a/2019/automation.md b/_2019/automation.md
similarity index 100%
rename from 2019/automation.md
rename to _2019/automation.md
diff --git a/2019/backups.md b/_2019/backups.md
similarity index 100%
rename from 2019/backups.md
rename to _2019/backups.md
diff --git a/2019/command-line.md b/_2019/command-line.md
similarity index 100%
rename from 2019/command-line.md
rename to _2019/command-line.md
diff --git a/2019/course-overview.md b/_2019/course-overview.md
similarity index 100%
rename from 2019/course-overview.md
rename to _2019/course-overview.md
diff --git a/2019/data-wrangling.md b/_2019/data-wrangling.md
similarity index 100%
rename from 2019/data-wrangling.md
rename to _2019/data-wrangling.md
diff --git a/2019/dotfiles.md b/_2019/dotfiles.md
similarity index 100%
rename from 2019/dotfiles.md
rename to _2019/dotfiles.md
diff --git a/2019/editors.md b/_2019/editors.md
similarity index 100%
rename from 2019/editors.md
rename to _2019/editors.md
diff --git a/2019/files/example.c b/_2019/files/example.c
similarity index 100%
rename from 2019/files/example.c

```

2.3.4 使用 Git 时的一个常见错误是提交本不应该由 Git 管理的大文件，或是将含有敏感信息的文件提交给 Git。尝试向仓库中添加一个文件并添加提交信息，然后将其从历史中删除

首先，我们来提交一些敏感信息，在 Git 中依次输入 `echo "password123">my_password`、`git add`、`git commit -m "add password123 to file"`、`git log HEAD`，即可向库中提交密码这一敏感信息。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ echo "password123">my_password

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git add .
warning: in the working copy of 'my_password', LF will be replaced by CRLF the next time Git touches it

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Disable this message with "git config advice.addEmptyPathsSpec false"

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git commit -m "add password123 to file"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git log HEAD
commit f956db4d10d688dbfa85e0497d8ca593c9a529bf (HEAD -> master)
Author: zhaiyihang <2905271851@qq.com>
Date:   Fri Aug 23 11:00:49 2024 +0800

    add password123 to file

commit af054fa1aea2f2599e4474d96b63f73dd9bd145f (origin/master, origin/HEAD)
Merge: dd3f3dd 9baa48c
Author: Lingfeng_Ai <hanxiaomax@gmail.com>
Date:   Fri Aug 16 06:54:16 2024 +0800

    Merge pull request #172 from pspdada/master

    Thank you so much

commit 9baa48c778012164179e4e60725418941f41743b
Author: psp_dada <1824427006@qq.com>
Date:   Thu Aug 15 02:07:36 2024 +0800

    remove irrelevant text

commit f5df7de89dc7712483665cc6fe8a787aafbef9bf
Author: psp_dada <1824427006@qq.com>
Date:   Thu Aug 15 01:46:12 2024 +0800

    fix wrong index

commit ef9a2f75409ff7746c03f6233066e3d2c634cd12
Author: psp_dada <1824427006@qq.com>
Date:   Thu Aug 15 01:32:44 2024 +0800

    fix typo
```

```

commit f5df7de89dc7712483665cc6fe8a787aafbef9bf
Author: psp_dada <1824427006@qq.com>
Date: Thu Aug 15 01:46:12 2024 +0800

    fix wrong index

commit ef9a2f75409ff7746c03f6233066e3d2c634cd12
Author: psp_dada <1824427006@qq.com>
Date: Thu Aug 15 01:32:44 2024 +0800

    fix typo

commit dd3f3dd37ce88bb78f802a425e652a65d2ce70d5
Merge: d284d3e 8e26b4a
Author: Lingfeng_Ai <hanxiaomax@gmail.com>
Date: Thu Aug 1 17:58:28 2024 +0800

    Merge pull request #171 from HowieChih/for-better-understanding

    翻译优化: For better understanding

commit 8e26b4a0242ddaa6e239d314edbca8cb61e27cd3
Author: Howie Chih <howiechih@gmail.com>
Date: Thu Aug 1 17:00:37 2024 +0800

    更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译

commit 4e2ff4369c0c6aca74bdfd6ab12a0b397f6a9b11
Author: Howie Chih <howiechih@gmail.com>
Date: Thu Aug 1 14:54:53 2024 +0800

    更新#课程概览与 shell##一个功能全面又强大的工具关于修改亮度文件报错的翻译

commit d284d3e43bc43b81979e9ffd7a9b69d64450f707
Merge: f3773ac 6e27b4f
:|

```

然后，我们再来清除提交记录，在 Git 中依次输入 `git filter-branch --force --index-filter \`、`'git rm --cached --ignore-unmatch ./my_password' \`、`--prune-empty --tag-name-filter cat -- --all`，即可清除。

```

LENOVO@DESKTOP-00A7G4V MINGW64 /d/git/test1/missing-semester-cn.github.io (master)
$ git filter-branch --force --index-filter \
> 'git rm --cached --ignore-unmatch ./my_password' \
> --prune-empty --tag-name-filter cat -- --all
WARNING: git-filter-branch has a glut of gotchas generating mangled history
rewrites. Hit Ctrl-C before proceeding to abort, then use an
alternative filtering tool such as 'git filter-repo'
(https://github.com/newren/git-filter-repo/) instead. See the
filter-branch manual page for more details; to squelch this warning,
set FILTER_BRANCH_SQUELCH_WARNING=1.
Proceeding with filter-branch...

usage: git filter-branch [--setup <command>] [--subdirectory-filter <directory>] [--env-filter <command>]
      [--tree-filter <command>] [--index-filter <command>]
      [--parent-filter <command>] [--msg-filter <command>]
      [--commit-filter <command>] [--tag-name-filter <command>]
      [--original <namespace>]
      [-d <directory>] [-f | --force] [--state-branch <branch>]
      [--] [<rev-list options>...]

```



## 3 实验中遇到的问题与解决方法

### 3.1 在进行 Github 上仓库的克隆时，出现 Git 无法验证 GitHub 的 SSL 证书的报错

在搜索报错语句后得知，这是本地的证书问题或网络问题导致的。在 Git 中输入代码 `git config --global http.sslVerify false`，这样就可以绕过 SSL 证书检查，成功克隆仓库。

### 3.2 在进行课堂练习——向克隆下来的仓库中提交敏感信息时，无法提交，显示 `who you are?` 的提示

这是因为没有登记个人信息，在命令行终端输入指令 `git config --global user.email "邮箱"`、`git config --global user.name "名称"`，登记个人名称和邮箱之后即可继续进行。

### 3.3 在使用 Latex 进行文本编辑时，只能用英文进行编辑，使用中文会发生错误

在查阅相关书籍后得知，这是因为 Texstudio 默认用来编辑英文文档，无法识别中文，如果要编写中文文档需要在文档最上面写上代码 `\documentclass[UTF8,a4paper]{ctexart}`，让中文能够被识别，但是涉及到中英文都有的文档编写，就会展现出弊端——英文文本也是以中文的形式显示，具体体现在英文语句中的符号会被转换为中文输入法中的相应符号。

### 3.4 使用 Texstudio 撰写实验报告时，发现编译出来的 PDF 文档中出现部分文字无法显示或显示不全的问题

这是由于页边距没有调整好，需要在文档中加入 `geometry` 拓展包，然后用 `geometry` 指令来修改合适的页边距，即可将文本完全显示。

### 3.5 在插入图片时，发现即使加入了 `graphicx` 拓展包也无法插入

这是因为在中文文本中，想要插入图片需要将图片打包成一个文件夹后，放在与 `tex` 文件同一目录中，然后再通过语句 `\graphicspath` 引入图片所在的文件夹，用 `\includegraphics` 语句来表示要插入的图片、设置图片的长宽。

### 3.6 完成文本写作后，发现第一页是空白页，文本内容从第二页开始

这是应为第一页默认为封面页，需要添加封面。

## 4 实例练习

### 4.1 初始化 Git 仓库

在 Git 中输入命令 `git init`，即可在当前目录下创建一个新的仓库。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1
$ git init
Initialized empty Git repository in D:/Git/test1/.git/
```

### 4.2 查看当前状态

输入 `git status` 指令，可以显示当前工作目录和暂存区的状态，包括哪些文件被修改了、哪些文件是新添加的等。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git status
On branch master

No commits yet

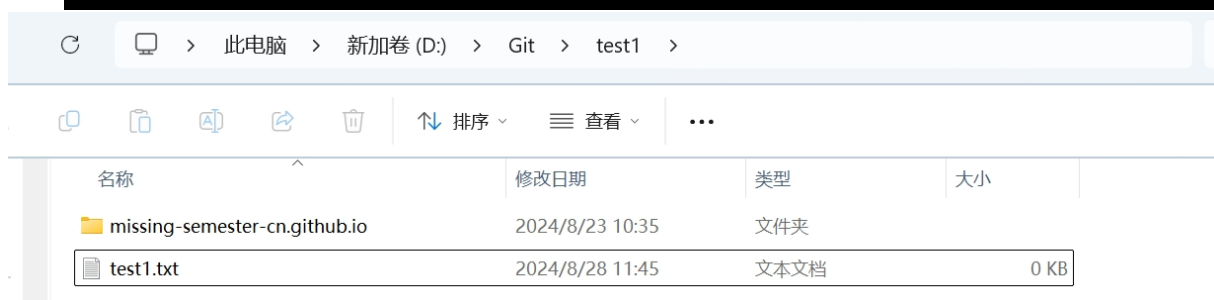
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    missing-semester-cn.github.io/

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

### 4.3 添加文件到暂存区

`git add < 文件名 >`，表示将指定文件添加到暂存区，准备提交。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git add test1.txt
```



### 4.4 提交更改

`git commit -m "提交信息"`，将暂存区的更改提交到仓库中，并附上提交信息。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git commit -m "test"
[master (root-commit) 514bffa] test
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test1.txt
```

#### 4.5 查看提交历史

git log, 显示当前分支的提交历史, 包括提交 ID、作者、日期和提交信息。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git log
commit 514bfffaba6aa52c3598e5fa1b279ed32e443ef94 (HEAD -> master)
Author: zhaiyihang <2905271851@qq.com>
Date:   Wed Aug 28 11:47:18 2024 +0800

    test
```

#### 4.6 查看文件差异

git diff, 查看工作目录中已修改但尚未暂存的文件与最后一次提交之间的差异。若要查看暂存区与最后一次提交之间的差异, 可以使用 git diff -cached。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git diff

LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$
```

#### 4.7 撤销工作目录的更改

git checkout - < 文件名 >, 将指定文件从暂存区或最近一次提交中检出到工作目录, 以撤销工作目录中的更改。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git checkout -- test1.txt

LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ |
```

## 4.8 撤销暂存区的更改

`git reset HEAD < 文件名 >`，将指定文件从暂存区撤销，回到工作目录，但文件内容不会改变（仅取消暂存）。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git add test2.txt

LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git reset HEAD test2.txt

LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ |
```

## 4.9 切换到其他分支

`git checkout < 分支名 >`，切换到指定分支。如果分支不存在，则会报错。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (newname)
$ git checkout master
Switched to branch 'master'

LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ |
```

## 4.10 创建并切换新分支

`git checkout -b < 新分支名 >`，创建并切换到一个新的分支。这个命令是 `git branch < 新分支名 >` 和 `git checkout < 新分支名 >` 的合并。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (master)
$ git checkout -b newname
Switched to a new branch 'newname'

LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1 (newname)
$ |
```

## 4.11 查看远程仓库信息

`git remote -v`，这个命令会列出所有配置的远程仓库及其 URL。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /d/Git/test1/missing-semester-cn.github.io (master)
$ git remote -v
origin https://github.com/missing-semester-cn/missing-semester-cn.github.io.git
(fetch)
origin https://github.com/missing-semester-cn/missing-semester-cn.github.io.git
(push)
```

#### 4.12 克隆远程仓库

`git clone < 仓库 URL >`，克隆一个远程仓库到本地目录。（由于该实例在课堂练习的准备阶段已经克隆过 Github 上的一个仓库，在此处不再附加图片展示）

#### 4.13 删除本地分支

`git branch -D < 分支名 >`，-D 是大写的，用于强制删除分支，即使分支未合并。如果只是想删除已合并的分支，可以使用-d。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /test1 (master)
$ git branch -D newname
Deleted branch newname (was 514bffa).
```

#### 4.14 比较暂存区与最近一次提交的差异

`git diff --cached`，显示暂存区与最近一次提交之间的差异。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /test1 (newname)
$ git add test2.txt

LENOVO@DESKTOP-00A7G4V MINGW64 /test1 (newname)
$ git diff --cached
diff --git a/test2.txt b/test2.txt
new file mode 100644
index 0000000..be01025
--- /dev/null
+++ b/test2.txt
@@ -0,0 +1 @@
+456789
\ No newline at end of file
```

### 4.15 合并分支

`git checkout` < 目标分支 >、`git merge` < 源分支 >，将 < 源分支 > 的更改合并到 < 目标分支 > 中。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /test1 (master)
$ git checkout newname
Switched to branch 'newname'
A       test2.txt

LENOVO@DESKTOP-00A7G4V MINGW64 /test1 (newname)
$ git merge master
Already up to date.
```

### 4.16 查看分支列表

`git branch`，列出本地分支。

```
LENOVO@DESKTOP-00A7G4V MINGW64 /test1/.git (GIT_DIR!)
$ git branch
* master
  newname
```

### 4.17 用 Latex 创建一个简单的文档

<pre>1 \documentclass{article} 2 \begin{document} 3   Hello, LaTeX! 4 \end{document}</pre>	<p>Hello, LaTeX!</p>
--	----------------------

### 4.18 添加标题和作者

<pre>\documentclass{article} \title{我的LaTeX文档} \author{作者名} \begin{document} \maketitle Hello, LaTeX! \end{document}</pre>	<p>LaTeX</p> <p>August 28, 2024</p> <p>Hello, LaTeX!</p>
--	--

### 4.19 插入数学公式

```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
  爱因斯坦的质能方程是：
  \[ E = mc^2 \]
\end{document}
```

$$E = mc^2$$

### 4.20 注释

```
\subsection{注释}
% 这是一个注释，不会出现在最终的文档中。
```

4.20 注释

## 5 实验收获与感悟

通过学习 Git 与 Latex，我深刻认识到了版本控制在项目管理中的重要性，也知道 Git 可以对有效记录进行更改、解决冲突；掌握了掌握 Git 分支与合并技巧，同时也明白 Git 可以成为团队协作的桥梁，用来减少沟通成本，提升效率；此次对于 Git 的学习也使我进一步适应了其独特的工作和思维方式。LaTeX 拥有卓越的排版能力，可以生成生成美观、专业的文档；同时，其强大的数学公式编辑功能，可以轻松呈现复杂数学表达式，提升论文可读性，并使之符合学术规范；除此之外我可以在网上查找合适、美观的论文模板，并将其 tex 文件直接移植过来方便自己应用，大大提高了写论文的工作效率，这为我将来撰写数学建模比赛论文打下了坚实的基础。

学习 Git 和 LaTeX 的过程虽然充满挑战，但它们带给我的收获和感悟却是无法估量的。Git 让我掌握了版本控制和团队协作的精髓，而 LaTeX 则让我能够以更加专业和高效的方式编写文档。这两种工具不仅提升了我的技术能力，还让我更加深入地理解了项目管理和学术研究的本质。我相信在未来的学习和工作中，它们将继续发挥重要作用并助我一臂之力。

Github 仓库链接：<https://github.com/Locuslaer/- .git>