

華東理工大學

信息科学与工程学院

《软件工程》 实验报告

系 别 计算机系

专 业 计算机科学与技术

年 级 2020 级

姓 名 刘子言

指导教师 阮 彤

2022-2023 学年 第 1 学期

实验四 基于 GIT 的版本管理

一、实验目的

- 1、了解 Git 与 Gitee 的使用，掌握其基本操作；
- 2、学会创建远端仓库，并与本地仓库交互；
- 3、学会团队操作中主线 master 与分支的合并。

二、实验装置

个人 PC 机器，Git 安装，Gitee 网页版。

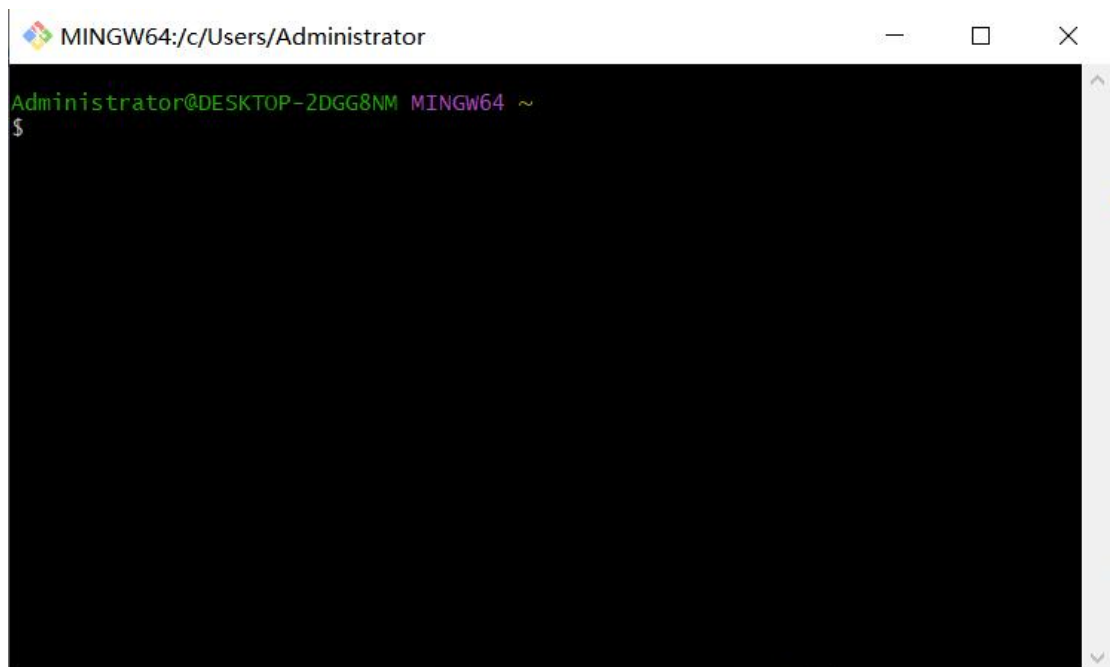
三、实验内容

1、安装 Git

1) 通过 <https://git-scm.com/download/> 下载 Git，然后选择安装路径，按默认选项安装。

名称	修改日期	类型	大小
Git	2022/10/28 11:35	文件夹	
Git-2.38.1-64-bit.exe	2022/10/28 11:33	应用程序	52,110 KB

2) 安装完成后，在开始菜单里找到“Git”->“Git Bash”，跳出一个类似命令行窗口，说明 Git 安装成功。



3) 使用 `git config` 命令设置 username 和 email，即为提交 commit 时的签名；执行 `git config` 命令以后，会在目录下建立一个 `.gitconfig` 文件，使用 `vim` 或 `cat` 查看文件。

```
Administrator@DESKTOP-2DGG8NM MINGW64 ~
$ git config --global user.name "20002462"

Administrator@DESKTOP-2DGG8NM MINGW64 ~
$ git config --global user.email "1846206177@qq.com"

Administrator@DESKTOP-2DGG8NM MINGW64 ~
$ cat ~/.gitconfig
[filter "lfs"]
    clean = git-lfs clean -- %f
    smudge = git-lfs smudge -- %f
    process = git-lfs filter-process
    required = true
[user]
    name = 20002462
    email = 1846206177@qq.com
```

2、注册 Gitee

1) 在 <https://gitee.com> 网站注册 Gitee 账户。



2) 然后新建一个远端仓库，名为“test4_project”。

新建仓库

在其他网站已经有仓库了吗？[点击导入](#)

仓库名称 * ✓

test4_project

归属

刘子言

路径 * ✓

/ test4_project

仓库地址: https://gitee.com/xiaoyu-conghua/test4_project

仓库介绍

18/200

实验四：基于Git的版本管理远程仓库

☐ 开源 (所有人可见) ②

☒ 私有 (仅仓库成员可见)

☐ 企业内部开源 (仅企业成员可见) ②

所有 我自己的 我参与的 | Forks 暂停/关闭的

全部 公开的 0 私有的 1

刘子言 / test4_project

暂无标签 +

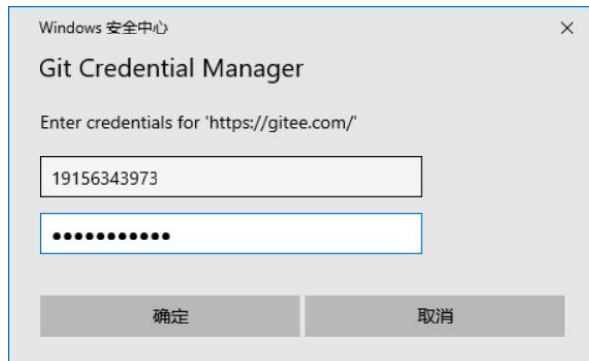
最近更新：暂未更新

3、Git 基本操作

1) 使用 git clone [远端 git 仓库] 将新建的项目克隆到 20002462Project 目录下。

远端仓库地址: https://gitee.com/xiaoyu-conghua/test4_project.git

- 来自 Gitee 端的用户名及密码验证:



- 克隆成功

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project
$ git clone https://gitee.com/xiaoyu-conghua/test4_project.git
Cloning into 'test4_project'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project
$ ls -a
./ ../ test4_project/
```

在 20002462Project 文件夹下发现已经存在新的项目文件, 说明已经克隆成功。

此电脑 > 本地磁盘 (D:) > 360安全浏览器下载 > GIT > 20002462Project				
名称	修改日期	类型	大小	
test4_project	2022/10/28 12:25	文件夹		

2) 进入刚才 test4_project 目录, 新建文件 text1 和 text2。修改文件, 使用 vim 编辑内容, 也可以直接 echo 添加内容。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project
$ cd test4_project

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ touch text1 text2

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ echo "hello text1">> text1

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ echo "hello text2">> text2

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ ls
README.en.md  README.md  text1  text2

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ |
```

3) 使用 `git status` 命令显示有变更的文件，可以看到有 2 个文件处于 `untracked` 状态。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        text1
        text2

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

4) 使用 `git add` 命令，将处于 `untracked` 状态的 2 个文件 `text1/text2` 加到缓存区 (Index)。再次执行 `git status`，会发现：这两个文件已经不再是 `untracked` 状态，而是被修改后的新文件状态，待 `commit` 到本地仓库。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ git add text1 text2
warning: in the working copy of 'text1', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'text2', LF will be replaced by CRLF the next time Git touches it

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (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)
        new file:   text1
        new file:   text2
```

5) 以上步骤已经为 `commit` 到本地仓库做好了准备。再使用 `git commit -m` 命令将缓存区文件提交到本地仓库。再次执行 `git status`，会发现没有变化的文件，也没有待提交的文件。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ git commit -m "add two files"
[master 9c8ffff] add two files
 2 files changed, 2 insertions(+)
 create mode 100644 text1
 create mode 100644 text2

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ git status
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
```

6) 使用 `git push` 命令，将本地仓库的文件更新到远端仓库中。可以看到 `gitee` 项目已经新增了 `text1` 和 `text2` 文件。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 353 bytes | 353.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-6.4]
To https://gitee.com/xiaoyu-conghua/test4_project.git
   9a19aa7..9c8ffff  master -> master
```


master ▾	🔗 分支 1 🔖 标签 0	+ Pull Request	+ Issue	文件 ▾	Web IDE	克隆/下载 ▾
2 20002462 add two files 9c8ffff 5分钟前 2 次提交						
📄 README.en.md		Initial commit		24分钟前		
📄 README.md		Initial commit		24分钟前		
📄 text1		add two files		5分钟前		
📄 text2		add two files		5分钟前		

7) 多成员操作

一般一个项目会有多个成员进行操作，当其他成员新增代码文件或者改动代码文件并 push 到远端仓库后，远端仓库的进程会比本地仓库快。这时就需要把远端新增的文件 git pull 到本地。

- 首先，手动在 Gitee 的 test4_project 项目中新建一个文件 text3 来模拟其他成员 push 到远端的代码文件。text3 的文件内容可以任意添加。

test4_project /

text3

提示: 输入 / 可以将文件创建到新文件夹下

```
1 This is text3, which is created by others.
```

master ▾	🔗 分支 1 🔖 标签 0	+ Pull Request	+ Issue	文件 ▾	Web IDE	克隆/下载 ▾
刘子言 add text3. 19ad745 刚刚 3 次提交						
📄 README.en.md		Initial commit		31分钟前		
📄 README.md		Initial commit		31分钟前		
📄 text1		add two files		11分钟前		
📄 text2		add two files		11分钟前		
📄 text3		add text3.		刚刚		

- 这时远端仓库进程比本地仓库快。再使用 git pull 命令，或者先 git fetch 再 git checkout，同步远端仓库的文件 text3 到本地。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 1010 bytes | 72.00 KiB/s, done.
From https://gitee.com/xiaoyu-conghua/test4_project
  9c8ffff..19ad745 master    -> origin/master
Updating 9c8ffff..19ad745
Fast-forward
 text3 | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 text3

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
master)
$ ls
README.en.md  README.md  text1  text2  text3
```

8) 分支与合并

Git 的分支，允许在主线（master 分支）之外进行代码提交，同时不会影响代码库主线。

- 首先，使用 `git checkout -b [分支名]` 命令，新建一个分支 b1，并跳转到这个分支目录下。再使用 `git branch` 查看所有的本地分支。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
master)
$ git checkout -b b1
Switched to a new branch 'b1'

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ git branch
* b1
  master
```

- 在 b1 分支下，创建并写入文件 text4；随后重复上述操作：`git add`—>`git commit`，将 text4 commit 到本地仓库。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ touch text4

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ echo "hello b1's text4">> text4

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ ls
README.en.md  README.md  text1  text2  text3  text4

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ git add text4
warning: in the working copy of 'text4', LF will be replaced by CRLF the next time Git touches
it

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ git commit -m "add a file"
[b1 fleefae] add a file
1 file changed, 1 insertion(+)
create mode 100644 text4
```

- 再使用 `git checkout master` 切换到主线 master 上，即：使合并分支操作的目标分支是 master 分支，使用 `git merge [源分支]` 命令将分支 b1 合并到 master 分支上。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
b1)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (
master)
$ git merge b1
Updating 19ad745..fleefae
Fast-forward
 text4 | 1 +
1 file changed, 1 insertion(+)
create mode 100644 text4
```

- 再次将新增内容 `git push` 到远端。

```
Administrator@DESKTOP-2DGG8NM MINGW64 /d/360安全浏览器下载/GIT/20002462Project/test4_project (master)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 274 bytes | 274.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Powered by GITEE.COM [GNK-6.4]
To https://gitee.com/xiaoyu-conghua/test4_project.git
19ad745..f1eefae master -> master
```

查看远程仓库 Gitee 已更新:

master 分支 1 标签 0 + Pull Request + Issue 文件 Web IDE 克隆/下载

2	20002462	add a file	f1eefae	6分钟前	4次提交
1	README.en.md	Initial commit		1小时前	
1	README.md	Initial commit		1小时前	
1	text1	add two files		31分钟前	
1	text2	add two files		31分钟前	
1	text3	add text3.		19分钟前	
1	text4	add a file		6分钟前	

查看本地文件夹 test4_project 中文件内容同上:

此电脑 > 本地磁盘 (D:) > 360安全浏览器下载 > GIT > 20002462Project > test4_project

名称	修改日期	类型	大小
README.en.md	2022/10/28 12:38	Markdown 源文件	1 KB
README.md	2022/10/28 12:38	Markdown 源文件	1 KB
text1	2022/10/28 12:41	文件	1 KB
text2	2022/10/28 12:42	文件	1 KB
text3	2022/10/28 13:06	文件	1 KB
text4	2022/10/28 13:20	文件	1 KB

四、实验心得

通过本实验，我掌握了如何使用 Git 与 Gitee 进行项目版本的管理。

我们通过 Git 与 Gitee 实现可以远端仓库与本地仓库的交互与同步，有利于项目相关文件的管理以及安全性（相当于备份功能）。

分支的作用，我觉得更多的是体现在多人协作的团队开发中。比如：一个团队开发软件，我负责其中独立的某一块功能，需要一个月的时间来完成，这时我就可以通过创建一个分支，把该功能的代码提交到这个分支，而其它成员仍然可以继续使用主线进行开发；这样我每天在分支上的提交，都不会对我的队友的主线开发造成任何影响；当我完成这个功能开发，并且测试通过以后，再将这个功能分支合并到主线上。这样的流程，有利于的项目版本的管理与更新，避免开发过程中造成的不必要的混乱。