

信息科学与工程学院

软件工程 实验报告

**系 别 信息科学与工程学院**

**专 业 计算机科学与技术**

**年 级 2021级**

**姓 名 徐昊博**

**指导教师 阮彤**

**2024 学年 第 二 学期**

**实验四 基于GIT的版本管理**

1. **实验目的**

本次实验主要目的是通过gitee平台进行代码项目的创建与团队代码项目管理的学习与测试实践从而达到对团队软件项目代码管理方法的熟练掌握。

1. **实验主要内容**

（1）学习关于git的原理

（2）在gitee网站上创建自己的账号并下载git bash用于代码开发仓库的搭建

（3）在gitbash中建立自己的项目仓库并在项目中进行文件新建的测试，测试add和commit等功能：

（3）测试将远端代码仓库中新建的文件pull到本地

（4）测试建立分支并将测试文件所在的分支合并到主分支中。

**三、实验过程**

**（1）关于GIT的知识总结：**

Git 是一个分布式版本控制系统，它主要用于跟踪文件的变化并协调多人之间的工作。Git 的代码仓库功能是其核心功能之一，git的主要功能和实现原理如下：

1. 版本控制：Git 能够跟踪文件的每一次变化，包括文件内容的修改、文件的添加和删除等操作。通过提交(commit)操作，可以将文件变化保存为一个新的版本，并记录相关的提交信息，如作者、时间等。

2. 分支管理：Git 允许在同一个代码仓库中创建多个分支(branch)，每个分支可以独立地进行开发和修改。这样不同的开发人员可以在不影响主线开发的情况下进行工作，最终再将各自的分支合并(merge)到主线上。

3. 远程仓库：Git 支持与远程代码仓库进行交互，可以将本地的代码推送(push)到远程仓库，也可以从远程仓库拉取(pull)最新的代码。这样可以方便多人协作，也可以备份和共享代码。

4. 分布式架构：Git 是一个分布式版本控制系统，每个开发者都可以拥有完整的代码仓库，包括完整的提交历史。这样即使在没有网络连接的情况下，开发者仍然可以进行版本控制和修改。

5. 分支引用：Git 使用分支引用(branch reference)来标记当前所在的分支以及最新的提交。分支引用存储在 `.git` 目录下的 `refs/heads` 目录中。

6. 提交对象：每次提交操作会创建一个提交对象(commit object)，其中包含了提交的作者、时间、提交消息以及指向树对象的引用。树对象(tree object)记录了当前文件状态的快照，包括文件名、文件权限和指向文件内容的指针。

7. 工作区和暂存区：工作区是指当前工作目录中的文件，而暂存区是索引文件中暂存的文件变化。通过 `git add` 命令可以将工作区的文件变化添加到暂存区，然后通过 `git commit` 命令将暂存区的内容提交到本地仓库。

Git工作的基本流程图可以如下简单表示：

本地仓库 工作区 暂存区 远程仓库

| | | |

|--- git clone ---> | | |

| |--- 修改文件 ---> | |

| | |--- git add ---> |

| | | |

| | |--- git commit -> |

| | | |

|--- git fetch <--- | | |

| | | |

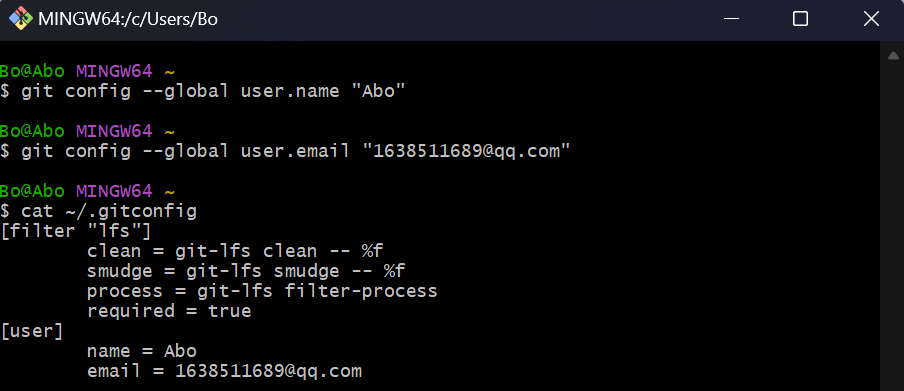
|--- git pull ----> | | |

| | | |

| | | |--- git push --->

| | | |

**（2）在gitee网站上创建自己的账号并下载git bash用于代码开发仓库的搭建**

1、安装gitee，完成后，设置名字和email，注意git config命令的--global参数，用了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然也可以对某个仓库指定不同的用户名和Email地址。设置完成后如下：

2、注册Gitee账号<https://gitee.com>。如下所示：

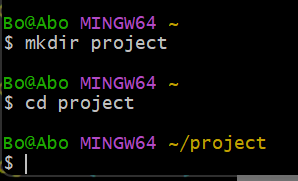


1. **在gitbash中建立自己的项目仓库并在项目中进行文件新建的测试，测试add和commit等功能：**

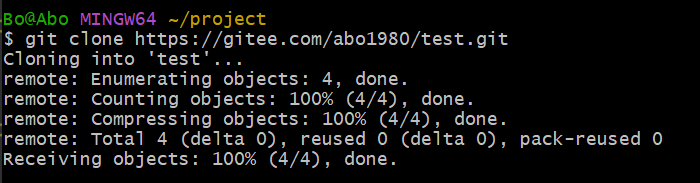
1、新建一个远段仓库：



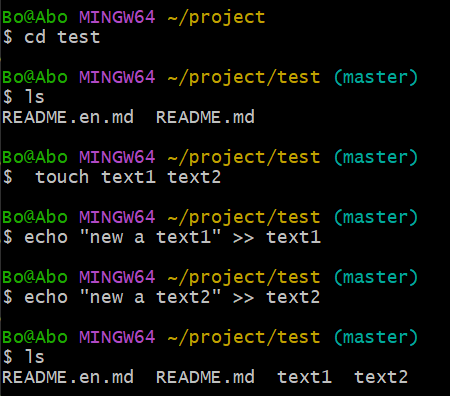
2、先进行路径转换，将路径转换到事先准备好的文件夹目录下：



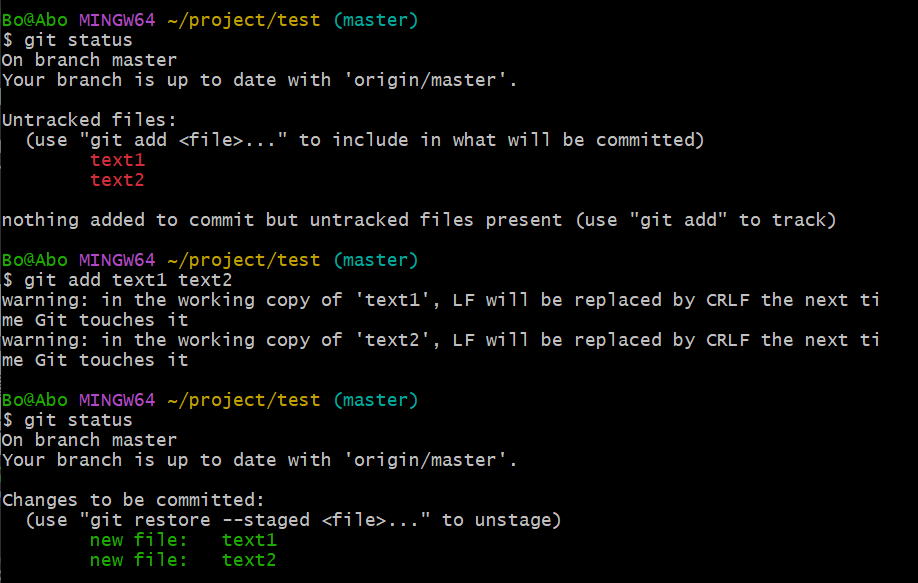
3、使用git clone [远端git仓库] 将新建的项目克隆到project 目录下：



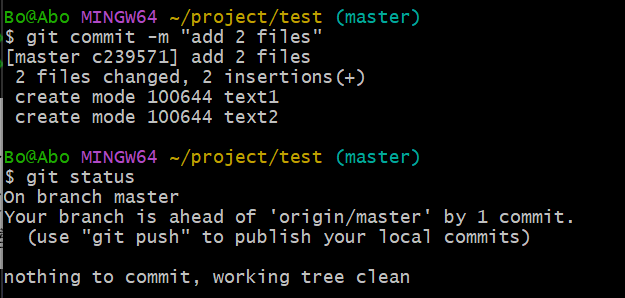
4、进入刚才textProject目录，新建文件text1和text2。修改文件，使用vim编辑内容，也可以直接echo添加内容：



5、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。使用git add命令将处于untracked状态的2个文件加入到缓存区（Index）。再次执行git status会发现新的变化。



6、现在已经为commit到本地仓库做好了准备。（可以使用git diff--cached命令查看缓存区和上一次commit差异，看哪些文件被修改了。）使用 git commit -m “本次提交内容的摘要” 命令将缓存区文件提交到本地仓库。再次执行git status会发现新的变化。



7、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的项目已新增text1和text2文件。

文本

描述已自动生成

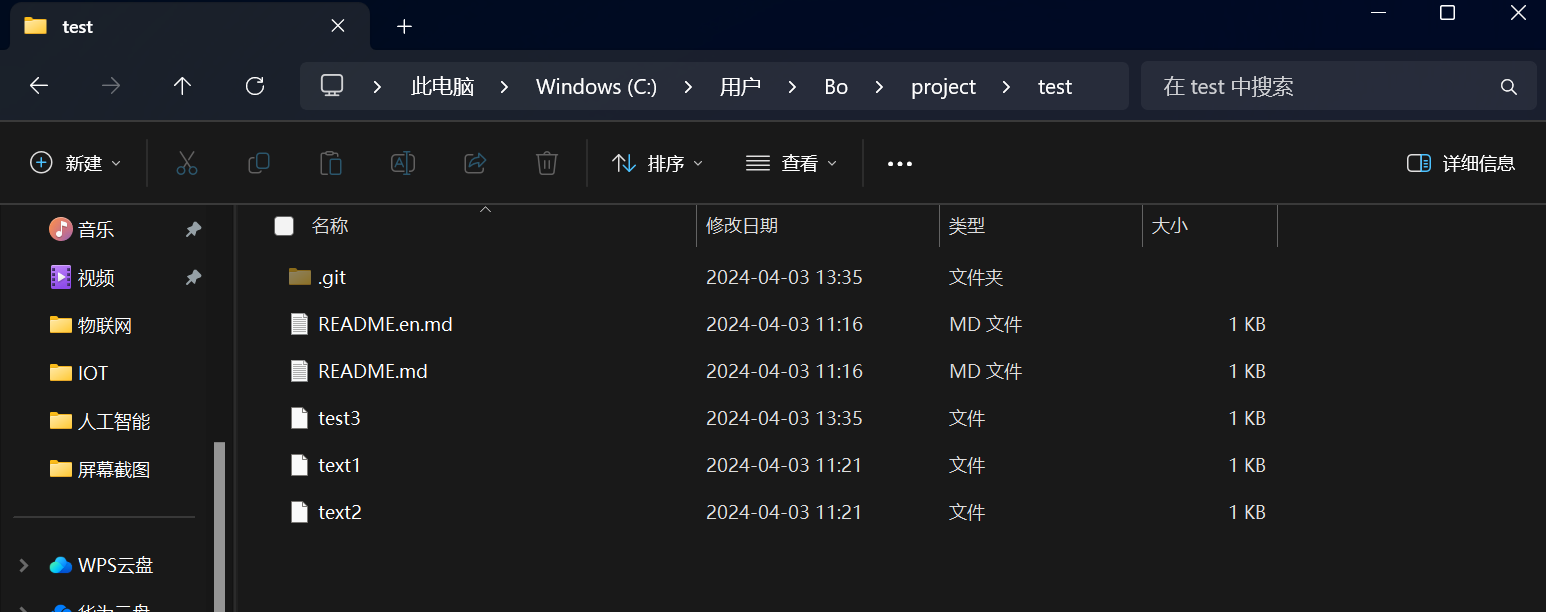
1. **测试将远端代码仓库中新建的文件pull到本地**

1、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。这是需要把远端新增文件git pull到本地。 这里，我们手动在gitee的testProject项目中新建一个文件test3 来模拟其他成员push到远端的代码文件。test3的文件内容可以任意添加,如图在远程代码仓库中建立新的test3。



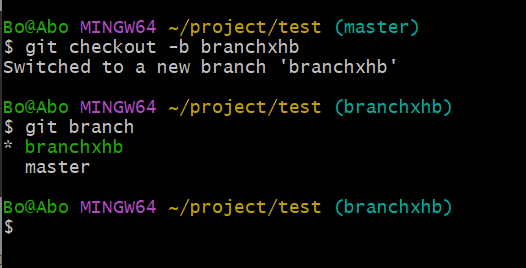
2、使用git pull 命令（或者先git fetch 再 git checkout）同步远端仓库的文件到本地。可以看到原先没有test3，pull之后则有了test3：



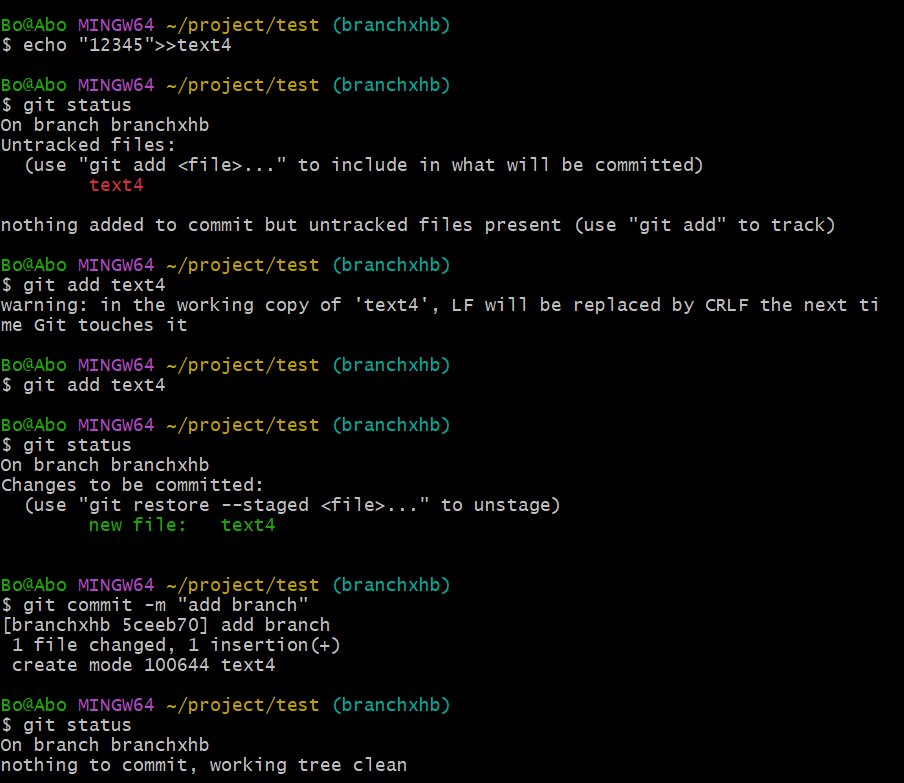


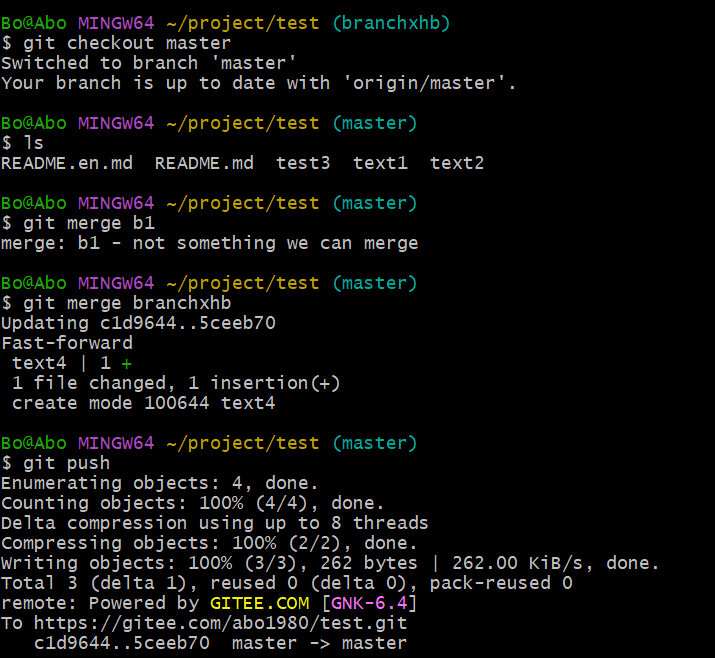
**（5）测试建立分支并将测试文件所在的分支合并到主分支中。**

1、使用git checkout -b [分支名] 命令新建一个分支并跳转到这个分支目录下。使用git branch 查看所有本地分支。

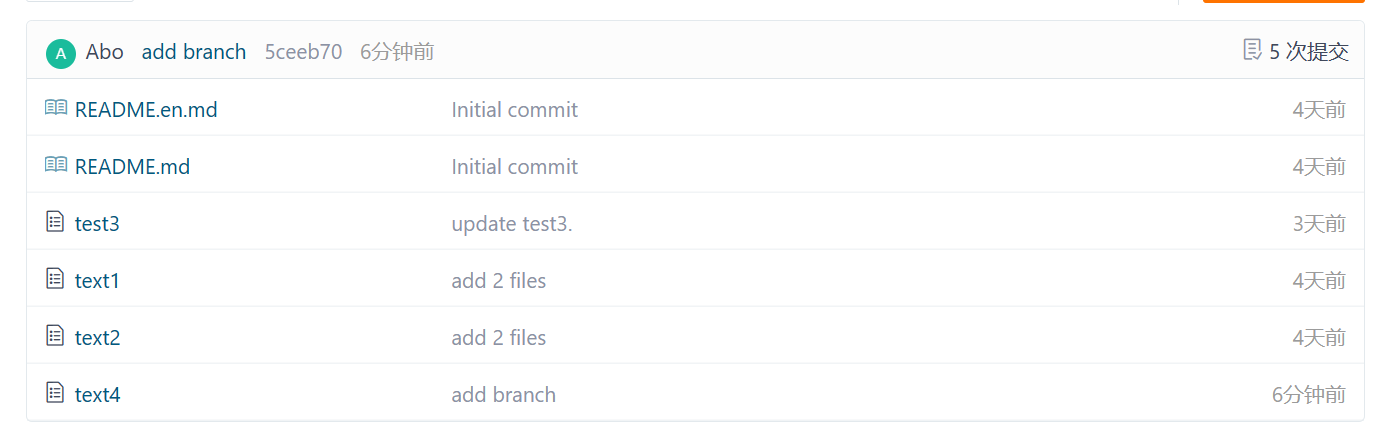


2、在branchxhb分支下，创建并写入文件text4，并重复上述操作，git add—>git commit，将text4 commit到本地仓库。使用git checkout master切换到主线上，即，使合并分支操作的目标分支是master分支，使用git merge [源分支] 命令将分支branchxhb合并到master分支上。



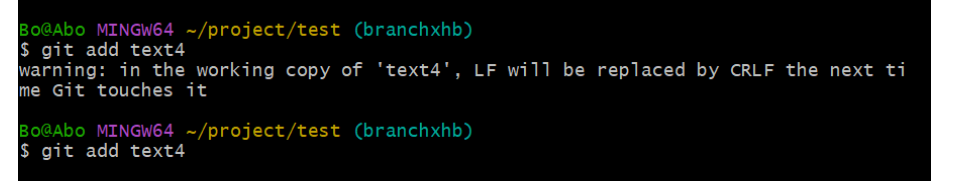


2、再次 git push到远端。查看远程仓库git已经更新如下所示：



1. **实验中碰到的问题**

当运行git add指令时出现warning，此时只需要重新再输入一次指令即可。



**五、实验总结**

本次实验的主要目的是通过Gitee平台进行代码项目的创建与团队代码项目管理的学习与测试实践，以达到对团队软件项目代码管理方法的熟练掌握。

1. 学习了Git的基本原理和操作，包括仓库（Repository）、分支（Branch）、提交（Commit）等概念以及克隆（Clone）、添加（Add）、提交（Commit）、推送（Push）、拉取（Pull）等操作。

2. 在Gitee网站上成功创建了个人账号，并下载安装了Git Bash工具，为代码开发仓库的搭建做好了准备。

3. 在Git Bash中初始化了一个新的项目仓库，并通过创建文件、添加文件到暂存区、提交更改等操作，测试了基本的Git功能。

4. 测试了将远端代码仓库中新建的文件pull到本地的过程，加深了对Git远程仓库管理的理解。

5. 成功创建了分支，并在分支上进行了文件的修改和添加。最后将分支合并到主分支中，掌握了分支管理和合并的基本技巧。

通过本次实验，我深入理解了Git的工作原理和基本操作，掌握了在Gitee平台上进行代码项目的创建和团队代码项目管理的方法。这对于今后的软件开发和团队协作有着重要的意义。实验过程中，我也遇到了一些问题，比如分支合并时的冲突解决，但通过查阅资料和不断尝试，我最终解决了这些问题。这次实验让我更加自信地使用Git和Gitee进行代码管理和团队协作，为将来的项目开发打下了坚实的基础。