GIT实验报告

5032018012 卢镝

目录

[一、 查看用户名与相应Email地址 2](#_Toc9457)

[二、 创建版本库 2](#_Toc25405)

[三、 工作区与暂存区 2](#_Toc13392)

[四、 文件修改操作： 4](#_Toc29348)

[五、 版本回退操作 5](#_Toc25267)

[六、 修改撤销操作 6](#_Toc4690)

[七、 删除文件 7](#_Toc16701)

[八、 分支 7](#_Toc917)

[（一） 创建分支： 7](#_Toc30191)

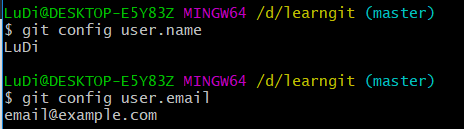
[（二） 切换分支： 8](#_Toc16043)

[（三） 分支合并 8](#_Toc24908)

[九、 远程仓库 9](#_Toc11868)

[十、 总结感悟 11](#_Toc26115)

# 查看用户名与相应Email地址

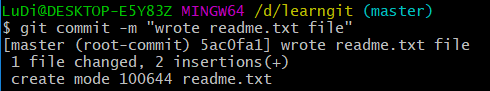


# 创建版本库

在D盘新建learngit文件夹并将此目录变成Git可以管理的仓库：



将文件添加到仓库：



# 工作区与暂存区

learngit文件夹就是一个工作区。工作区有个隐藏目录 .git ，这个不算工作区，而是 Git 的版本库。版本库里面的 index(stage) 文件叫暂存区，还有Git为我们自动创建的第一个分支 master。



在实践中，如果想把文件添加到Git里面时，需要分两步：

第一步是用 git add 把文件添加进去，实际上就是把文件修改添加到暂存区。

第二步是用 git commit 提交更改，实际上就是把暂存区的所有内容提交到当前分支。

从新建一个readme.txt文件到最终提交，工作区、暂存区大概要经历一下变化：

gitadd:

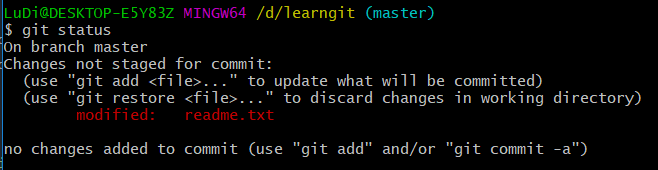


Gitcommit:

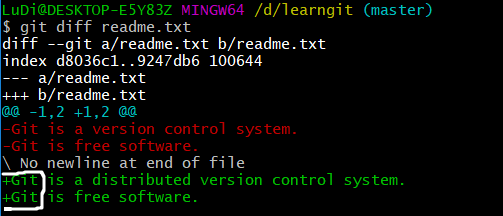


# 文件修改操作：

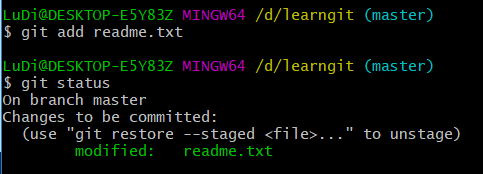
更改readme.txt中内容后，利用git status命令查看当前仓库状态，可见：readme.txt 文件已经被修改，但还没有准备提交的修改。



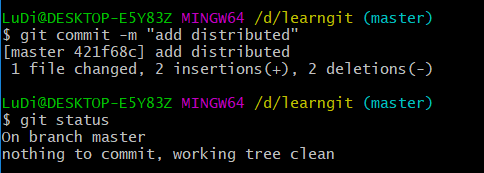
通过git diff命令查看修改操作：



可以看到相较于上次提交，文件中多了两句输入。再次提交（加到缓存区）并查看仓库状态：



可以看到readme.txt文件为将要被提交的文件。提交后并查看：



可见当前没有需要提交的修改，而且，工作目录是干净的。

# 版本回退操作

修改readme.txt文件如下：

|  |
| --- |
|  |

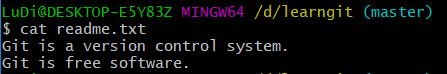
提交并查看日志：



可见readme.txt 文件一共有三个版本被提交到了Git仓库里。回退的操作是git reset，其中HEAD^^代表回退两个版本，如果是HEAD^则是回退到前一个版本：



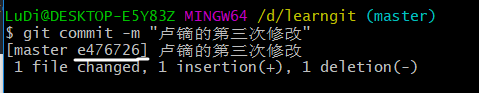
查看：



如果要再返回最新版本则是git reset --hard命令。



其中“e476726”是第三次提交的id，可以在提交时查看：



# 修改撤销操作

修改文件如下：

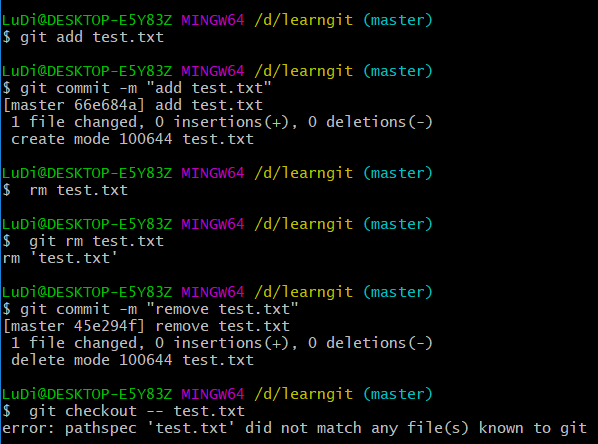
|  |
| --- |
|  |

如果没有git add至工作区则直接手动删除即可。如果git aad但没有git commit，则可以用git reset HEAD file 命令把把暂存区的修改撤销掉，重新放回工作区，然后git checkout -- readme.txt修改。

|  |
| --- |
|  |
|  |
|  |

# 删除文件

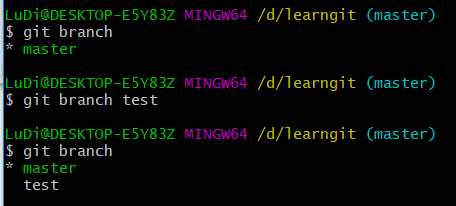
在工作区即 learngit 文件夹下新建一个 test.txt 文件，并添加和提交到Git：



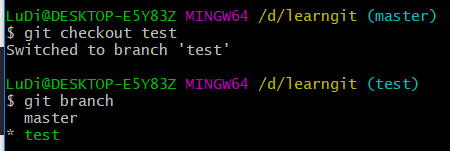
可见在git commit后已经无法通过 git checkout -- test.txt用版本库里的版本替换工作区的版本。

# 分支

## 创建分支：



## 切换分支：

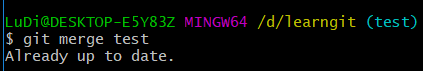


在test分支提交文件后返回master分支：

|  |  |
| --- | --- |
|  |  |
|  |  |

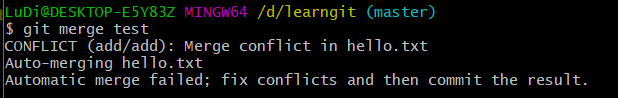
## 分支合并

无冲突：

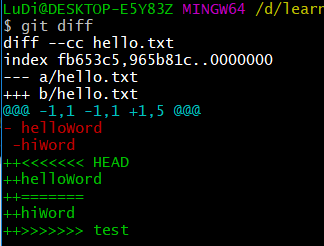


有冲突：

在这种情况下，两个分支中的hello.txt文件内容分别为“helloword”以及“hiword”，在合并时就会报出异常：



通过gif diff查看冲突内容如下：



然后直接使用vim对冲突内容进行更改，然后重新提交修改后的hello.txt文件。在此基础上再次进行合并：

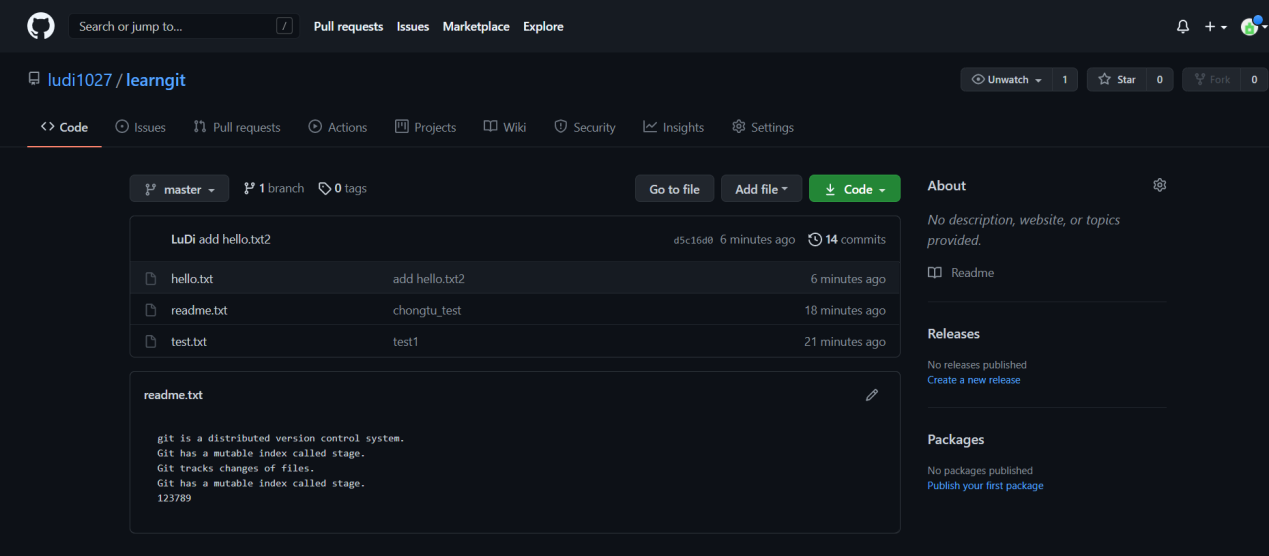
|  |
| --- |
|  |

# 远程仓库

首先进行SSH Key的创建，将id\_rsa.pub中的内容复制到Key文本框内，其中id\_rsa.pub为实现免密登录后自动生成的文件：

|  |
| --- |
|  |
|  |

通过git remote add origin git@github.com:ludi1027/learngit.git添加远程库（origin）。然后将本地库的所有内容推送到远程库上，如下图所示，可以看到在ludi1027下已经有了本地仓库“learngit”。



下面在master分支更改hello.txt的内容，然后添加到工作区、提交，再一次将本地仓库推送到远程库，刷新后即可发现hello.txt的内容已经发生改变。

|  |  |
| --- | --- |
|  |  |

# 总结感悟

通过本次实践，能真正去实现课上讲解的部分操作。在搜集网上资料来进行实验的过程中，也对git的理论，例如工作区等概念，有了更深一步的理解与认识。除此以外，在处理错误、异常的过程中也积累了一定经验。尤其是在合并分片时，对之前课堂上讲解的冲突原因以及处理冲突的方法有了更深刻的理解并付出了实践。

以往只是在GitHub使用别人的代码，也是通过本次实验，我拥有了自己的第一个远程库，为日后自己代码、工程的管理工作打下了一个良好开端。