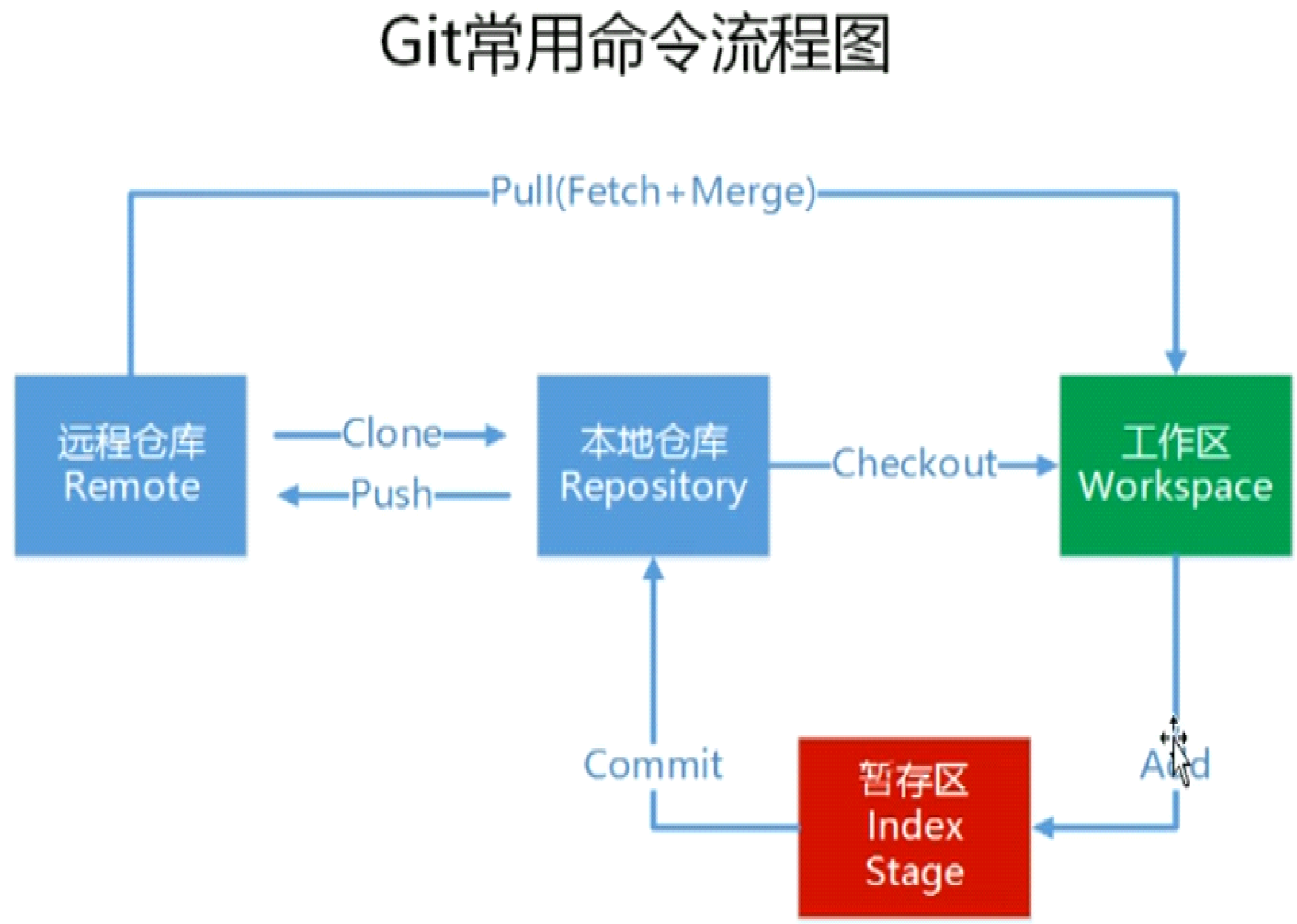
2019年12月25日 14:19

1.SVn 集中式版本控制器

2.git共享仓库



3.GitBush命令行/GitGUI图形化界面

4.GitBush在Windows下建立了一个Linux环境

Git简介

2020年6月22日　　10:50

1.git概念

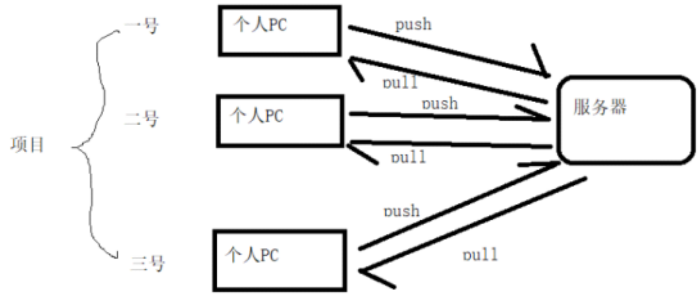
免费、开源

分布式（项目存储的位置在很多个点上，工作组的人都有，建立一个项目可以让多个人开发共享，分工上传；把相同的内容分布在很多地方；磁盘有上限但项目无上限，把相同的数据分散到多个服务器或终端上）

版本控制系统，

可以快速高效地处理从小到大型的项目。

✦项目：一组文件的集合（不仅代码、文件等等都可）



水平扩容：电脑加移动硬盘，不需要提升电脑能力。

2.其他常见的版本控制工具

subversion（简称SVN，局域网，特殊搭建才可实现）、CVS（部分银行，基于unix,在linux也可用，比较繁琐）等。

3.版本控制工具的作用

✦协同修改

✦多人并不悖的修改服务器端的同一个文件

✦数据备份

✦不仅保存目录和文件的当前状态，还能够保存每一个提交过的历史状态

✦版本管理

✦在保存每一个版本的文件信息的时候要做到不保存重复数据，以节约存储空间，提高运行效率。这方面SVN采用的是增量式管理的方式，而Git采用了文件系统快照的方式（效率更高）。

✦权限管理

✦对团队中参与开发的人员进行权限控制

✦对团队外开发者贡献的代码进行审核--Git独有（分支、开副本）

✦历史记录

✦查看修改人、修改时间、修改内容、日志信息

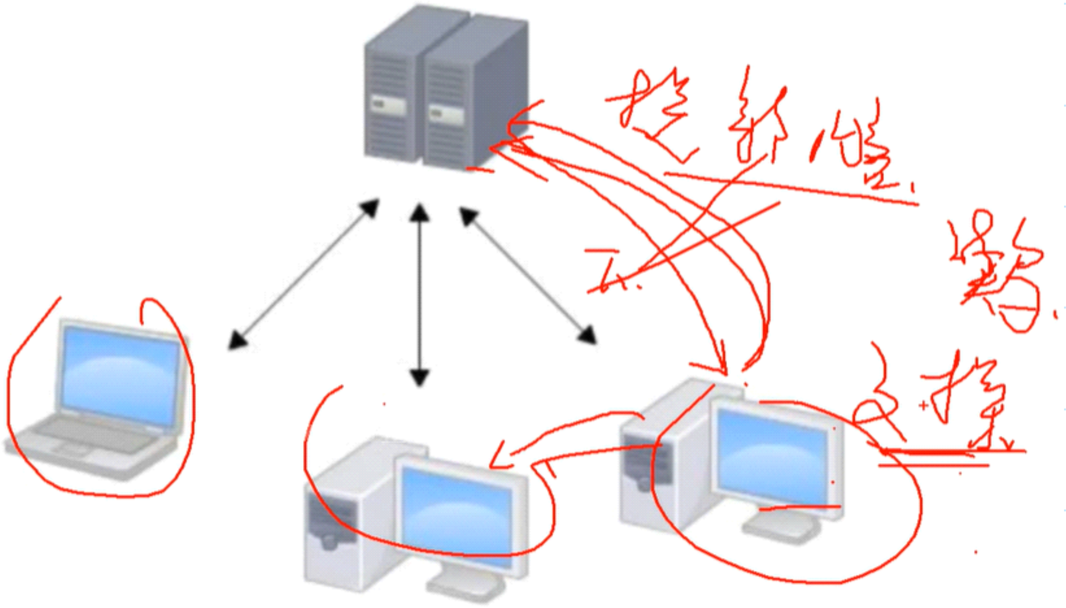
✦将本地文件恢复到某一个历史状态

✦分支管理

✦允许开发团队在工作过程中多条生产线同时推进任务，进一步提高效率

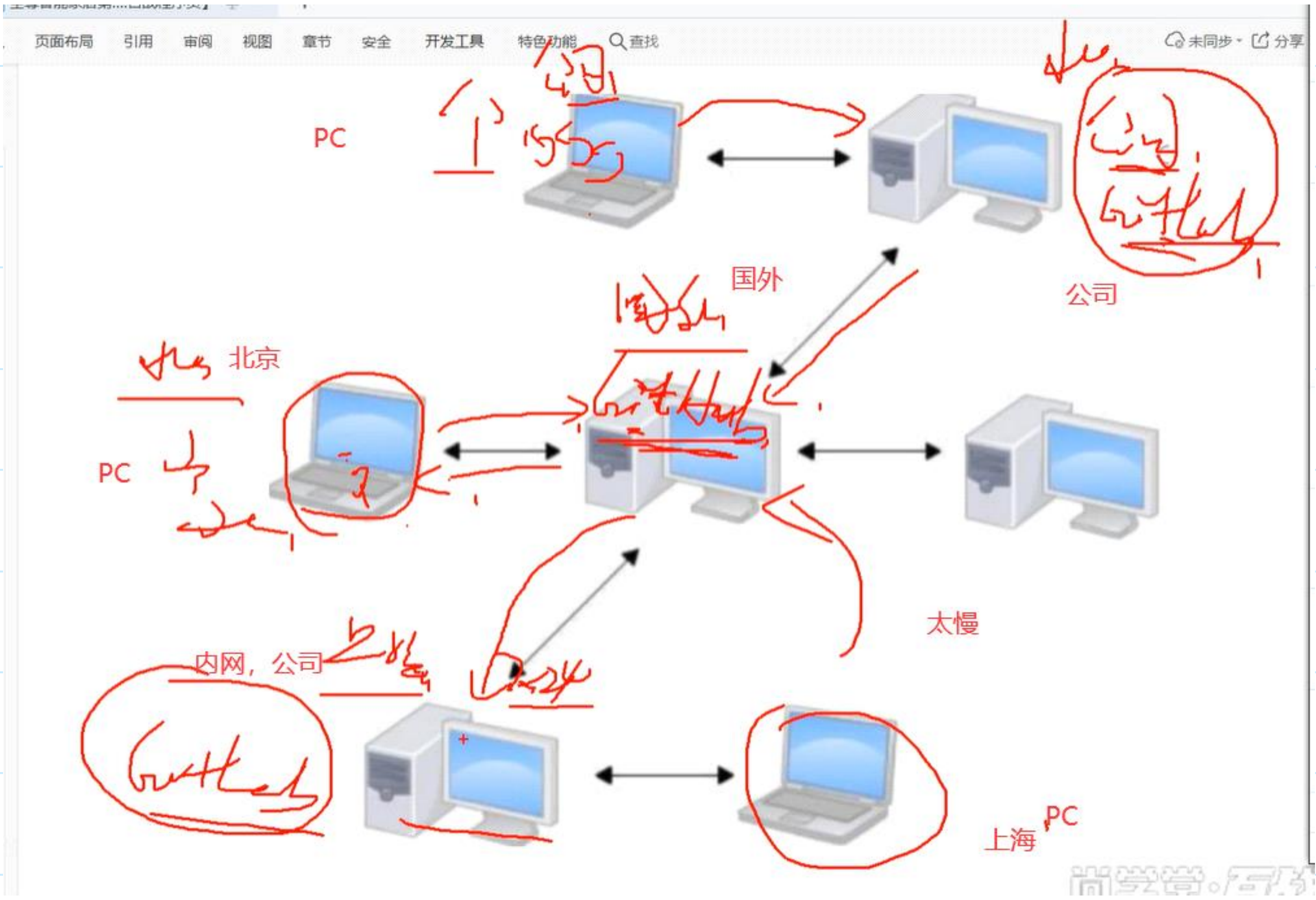
4.集中式版本控制工具及分布式版本控制工具示意图

集中式



提交过程中每一次文件的操作，每一次版本的管理都必须访问服务端

分布式



可以是外网服务器（GitHub）也可是内网服务器（公司内部gitlub）

外网或者是gitee

5.git官网

http://git-scm.com/ 独立的技术，本地电脑（客户端）安装软件

logo:

6.git优势

大部分操作在本地完成，不需要联网

尽可能添加数据而不是删除或修改数据（增量维护，不是变量，即便删除的文档也可恢复-<快照>）

分支操作非常快捷流畅（快照，备份）

与Linux命令全面兼容

7.git和代码托管中心（多托管）

代码托管中心得任务：维护远程库

局域网环境下

gitlab服务器（企业内部搭建私有服务器，方便成本低）

外网环境下

GitHub

码云（gitee）<开源中国架设的>

git的基本使用

2020年6月22日　　10:50

1.安装及初始化本地仓库

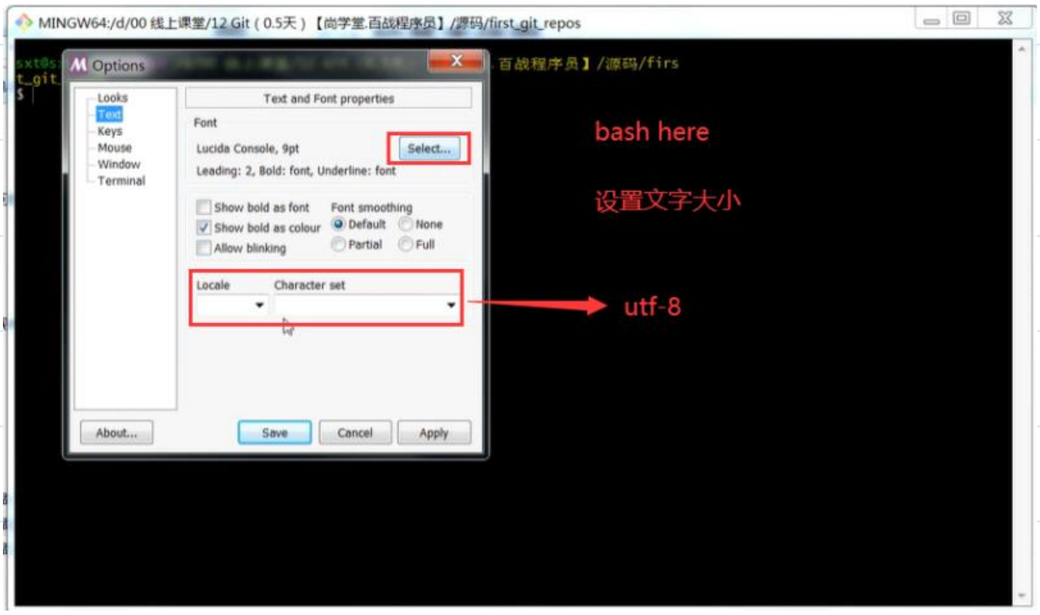
不能安装在中文目录下

安装完成打开git bash 窗口

输入git --version输出版本信息证明安装成功

填写用户名和邮箱作为一个标识

git config --global user.name "用户名"



MINGW64:/d/00 线上课堂/12 Git（0.5天）【尚学堂.百战程序员】/源码/first_git_repos

Options

Text and Font properties

Font: Lucida Console, 9pt

Leading: 2, Bold: font, Underline: font

□ Show bold as font
☑ Show bold as colour
□ Allow blinking

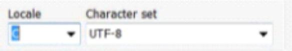
Font smoothing: ● Default ○ None ○ Partial ○ Full

Locale　Character set

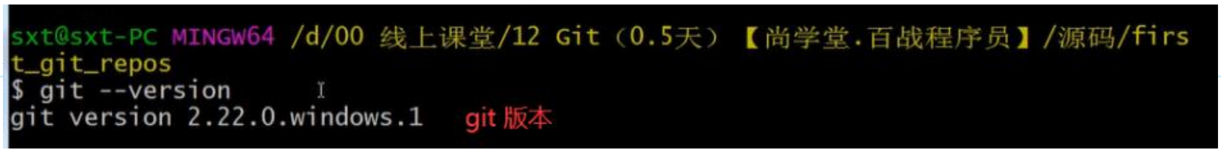
bash here

设置文字大小

utf-8



Locale: C　Character set: UTF-8



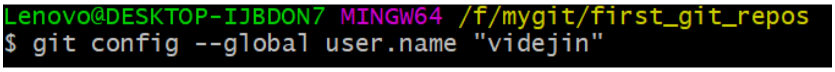
```
sxt@sxt-PC MINGW64 /d/00 线上课堂/12 Git（0.5天）【尚学堂.百战程序员】/源码/firs
t_git_repos
$ git --version
git version 2.22.0.windows.1
```

git 版本

✦git内核遵从Linux，没有消息就是好消息

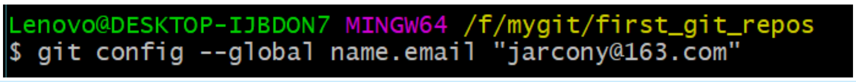
✧Git config（做配置，为git后续命令增加一些默认的本地配置心息）　--global（公共配置，全局的）

✦个人信息



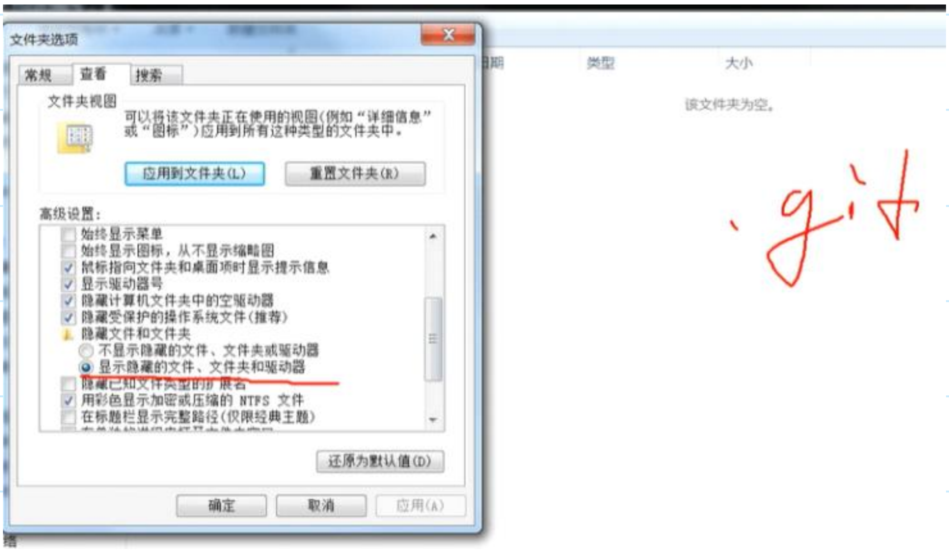
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos
$ git config --global user.name "videjin"
```

✦个人邮箱地址



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos
$ git config --global name.email "jarcony@163.com"
```

✦目录中有.git则是git的本地仓库，git bash可识别



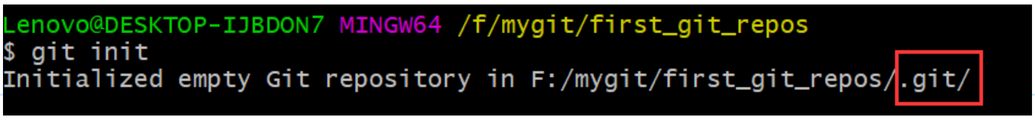
文件夹视图

可以将该文件夹正在使用的视图(例如“详细信息”或“图标”)应用到所有这种类型的文件夹中。

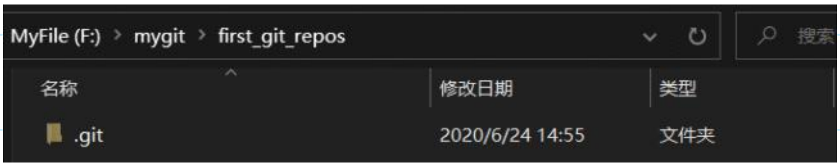
高级设置:

- □ 始终显示菜单
- □ 始终显示图标，从不显示缩略图
- ☑ 鼠标指向文件夹和桌面项时显示提示信息
- ☑ 显示驱动器号
- ☑ 隐藏计算机文件夹中的空驱动器
- ☑ 隐藏受保护的操作系统文件(推荐)
- 隐藏文件和文件夹
  - ○ 不显示隐藏的文件、文件夹或驱动器
  - ● 显示隐藏的文件、文件夹和驱动器
- □ 隐藏已知文件类型的扩展名
- ☑ 用彩色显示加密或压缩的 NTFS 文件
- □ 在标题栏显示完整路径(仅限经典主题)

◆初始化git本地仓库



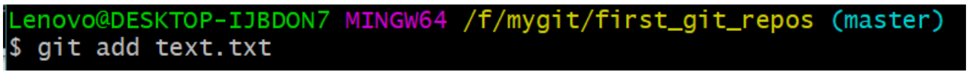
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos
$ git init
Initialized empty Git repository in F:/mygit/first_git_repos/.git/
```



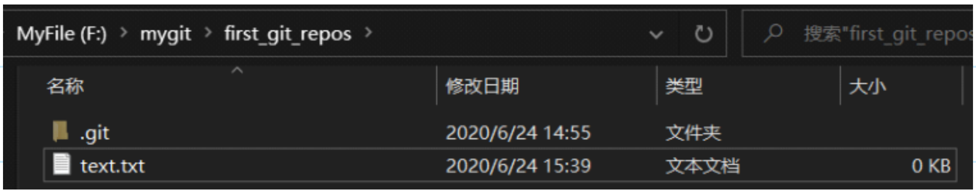
名称	修改日期	类型
.git	2020/6/24 14:55	文件夹

◆对文件的处理

git add 文件名

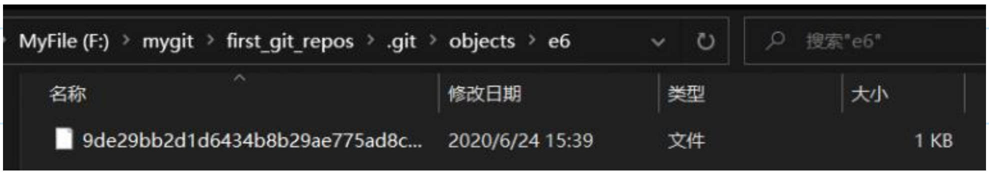


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git add text.txt
```



名称	修改日期	类型	大小
.git	2020/6/24 14:55	文件夹	
text.txt	2020/6/24 15:39	文本文档	0 KB

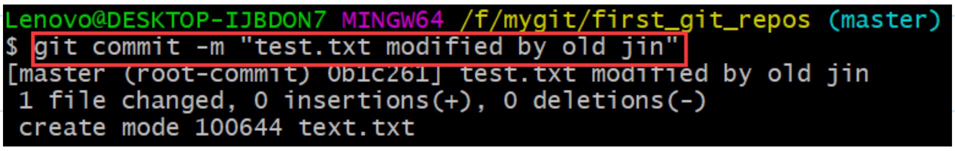
文件没有区别，但是在.git中增加了文件（不可解析，git可以）



名称	修改日期	类型	大小
9de29bb2d1d6434b8b29ae775ad8c...	2020/6/24 15:39	文件	1 KB

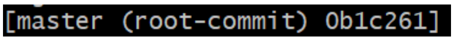
直接打开文件修改，但git不知道，需要git commit提交到git后加描述

git commit –m（描述）”test”



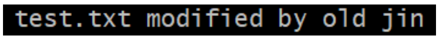
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git commit -m "test.txt modified by old jin"
[master (root-commit) 0b1c261] test.txt modified by old jin
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 text.txt
```

master (root-commit) 0b1c261是提交了一个文件



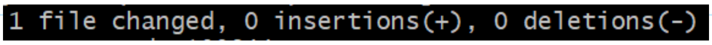
```
[master (root-commit) 0b1c261]
```

文件描述



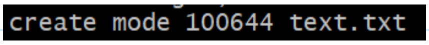
```
test.txt modified by old jin
```

一个文件修改了，没有新增的，没有删除的



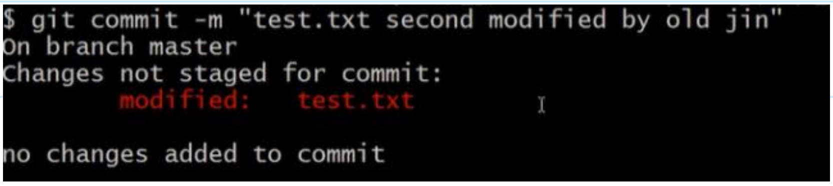
```
1 file changed, 0 insertions(+), 0 deletions(-)
```

文件创建好了



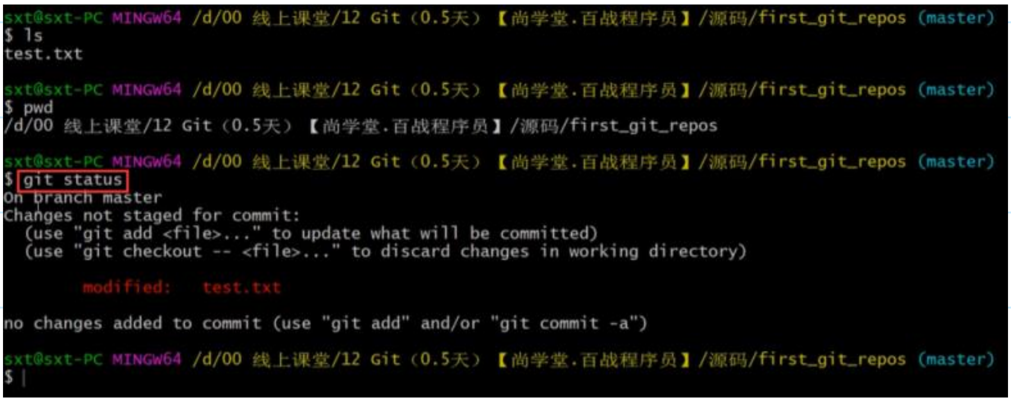
```
create mode 100644 text.txt
```

二次提交，文件没变化



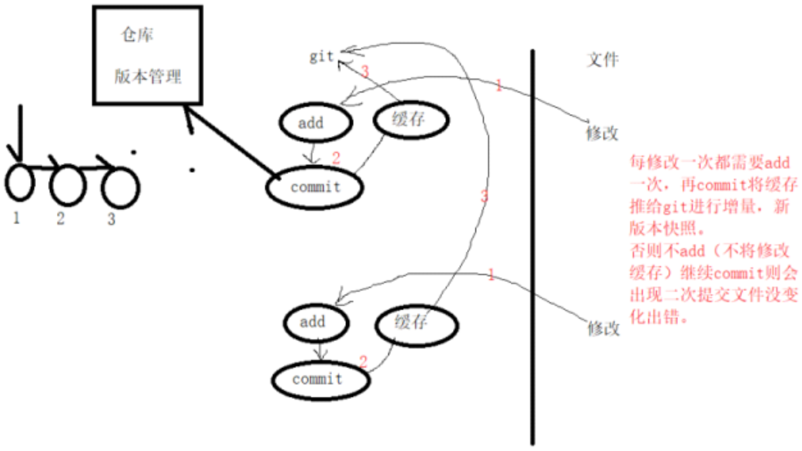
```
$ git commit -m "test.txt second modified by old jin"
On branch master
Changes not staged for commit:
        modified:   test.txt

no changes added to commit
```

修改的状态没有提交



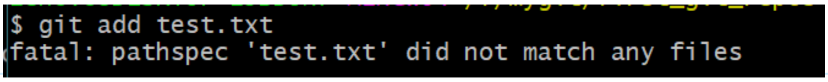
```
sxt@sxt-PC MINGW64 /d/00 线上课堂/12 Git (0.5天) 【尚学堂.百战程序员】/源码/first_git_repos (master)
$ ls
test.txt

sxt@sxt-PC MINGW64 /d/00 线上课堂/12 Git (0.5天) 【尚学堂.百战程序员】/源码/first_git_repos (master)
$ pwd
/d/00 线上课堂/12 Git (0.5天) 【尚学堂.百战程序员】/源码/first_git_repos

sxt@sxt-PC MINGW64 /d/00 线上课堂/12 Git (0.5天) 【尚学堂.百战程序员】/源码/first_git_repos (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")

sxt@sxt-PC MINGW64 /d/00 线上课堂/12 Git (0.5天) 【尚学堂.百战程序员】/源码/first_git_repos (master)
$ |
```

错误解决：



每修改一次都需要add一次，再commit将缓存推给git进行增量，新版本快照。
否则不add（不将修改缓存）继续commit则会出现二次提交文件没变化出错。

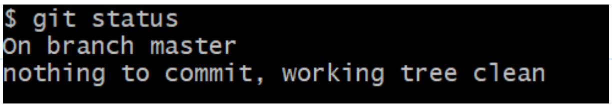
名字不对：



```
$ git add test.txt
fatal: pathspec 'test.txt' did not match any files
```

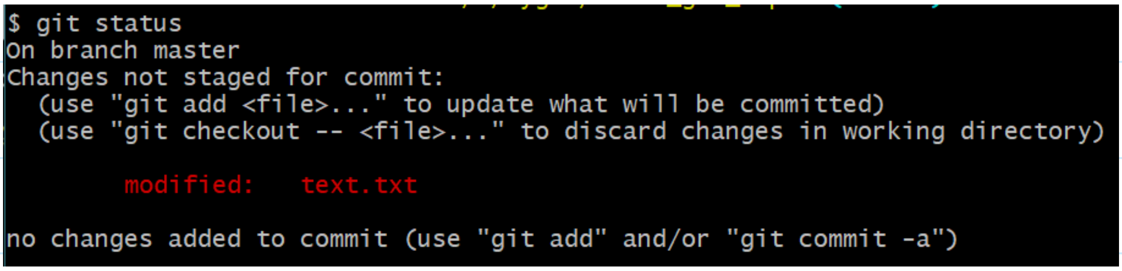
git status 状态

没有工作树，没有缓存，最新版本



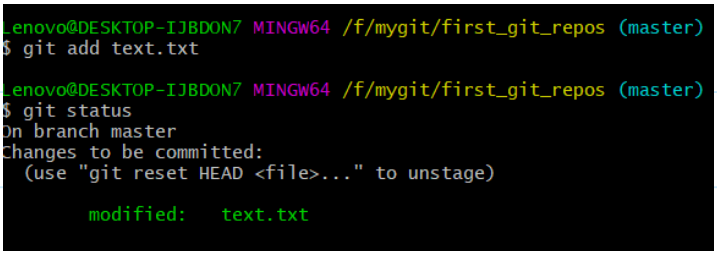
```
$ git status
On branch master
nothing to commit, working tree clean
```

继续修改text.txt，没有add则状态会出现

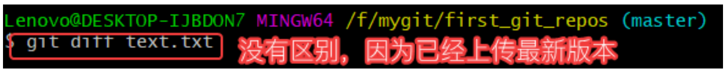


```
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   text.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

安装成功



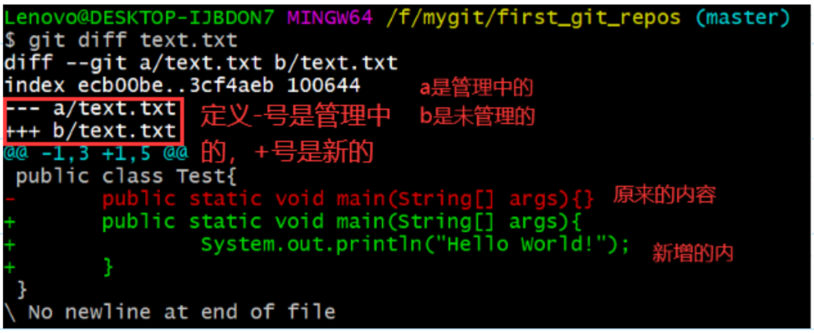
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git add text.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   text.txt
```



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git diff text.txt
```

没有区别，因为已经上传最新版本

2. git存储流程
    a. 代码工作区-->执行git add--->暂存区（临时存储）--->执行git commit--->本地库（历史版本）
    b. 工作区、暂存区和本地仓库，逻辑上是本地计算机。当我们新建一个文件时，文件位于工作区，处于已修改（modified）状态，表明文件已进行了修改，但还没有提交保存；通过命令git add将其添加到暂存区，文件是已暂存（staged）状态，表明把已修改的文件放到下次提交时要保存的清单中；通过命令git commit将文件放入本地仓库，文件为已提交（commited）状态，表示该文件已经被安全地保存在本地数据库中，到这一步可以说是成功生成了一个新的版本，旧版本进行快照。
    c. 远程仓库用来将本地仓库上传到网络，实现备份、共享和合作。
3. git常用命令
    a. 添加文件
        i. git add 文件名
            1）将文件添加到暂存区
    b. 提交文件
        i. git commit -m "提交注释"
            1）将文件提交到本地仓库
    c. 查看状态
        i. git status
            1）查看目录中是否还有文件未提交
    d. 查看文件修改对比差异
        i. git diff
            1）比较文件差异
            2）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git diff text.txt
diff --git a/text.txt b/text.txt
index ecb00be..3cf4aeb 100644
--- a/text.txt
+++ b/text.txt
@@ -1,3 +1,5 @@
 public class Test{
-       public static void main(String[] args){}
+       public static void main(String[] args){
+               System.out.println("Hello World!");
+       }
 }
\ No newline at end of file
```

a是管理中的

b是未管理的

定义-号是管理中的，+号是新的

原来的内容

新增的内

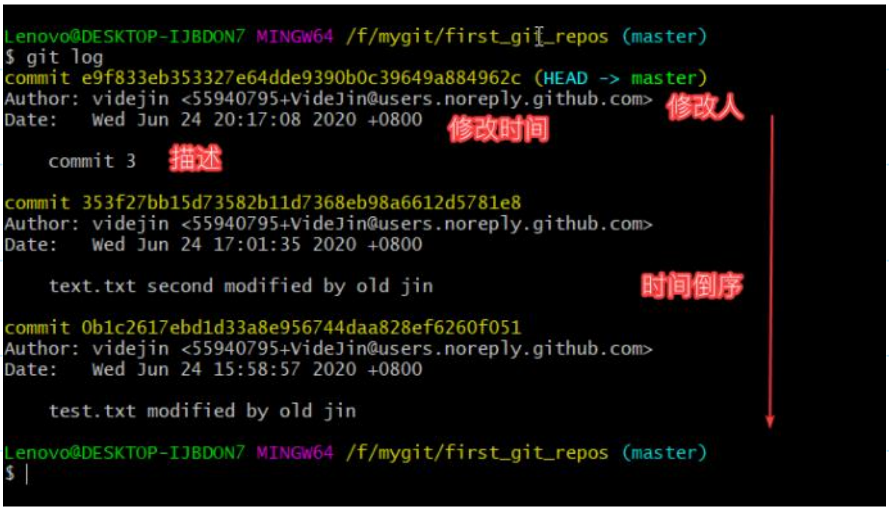
        ii. git diff[文件名] |
            1）将工作区中的文件和暂存区进行比较

e. 查看日志

i. git log

1）查看历史记录，显示从最近到最远的日志

2)



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git log
commit e9f833eb353327e64dde9390b0c39649a884962c (HEAD -> master)
Author: videjin <55940795+VideJin@users.noreply.github.com>
Date:   Wed Jun 24 20:17:08 2020 +0800

    commit 3

commit 353f27bb15d73582b11d7368eb98a6612d5781e8
Author: videjin <55940795+VideJin@users.noreply.github.com>
Date:   Wed Jun 24 17:01:35 2020 +0800

    text.txt second modified by old jin

commit 0b1c2617ebd1d33a8e956744daa828ef6260f051
Author: videjin <55940795+VideJin@users.noreply.github.com>
Date:   Wed Jun 24 15:58:57 2020 +0800

    test.txt modified by old jin

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ |
```

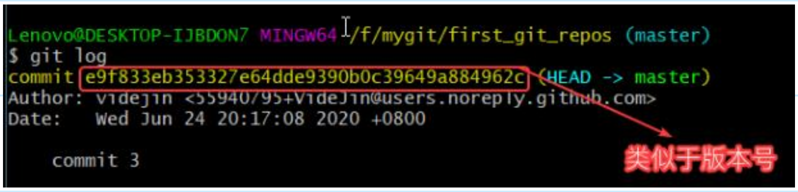
修改人

修改时间

描述

时间倒序

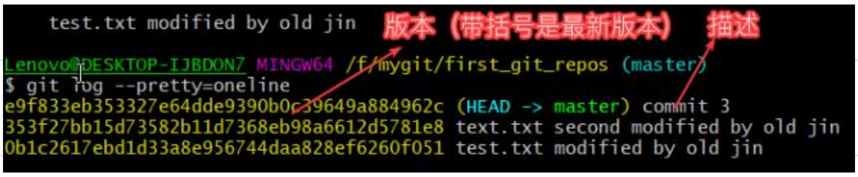
3)



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git log
commit e9f833eb353327e64dde9390b0c39649a884962c (HEAD -> master)
Author: videjin <55940795+VideJin@users.noreply.github.com>
Date:   Wed Jun 24 20:17:08 2020 +0800

    commit 3
```

类似于版本号

4)

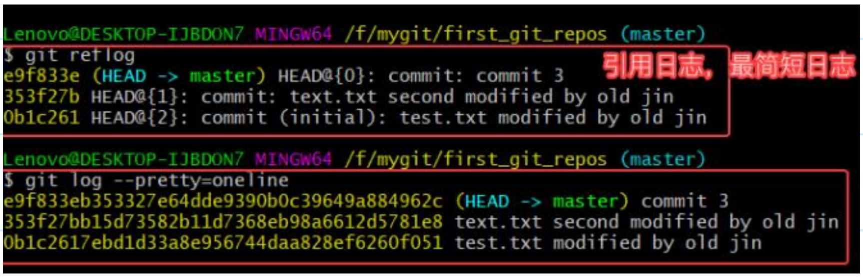


```
    test.txt modified by old jin

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git log --pretty=oneline
e9f833eb353327e64dde9390b0c39649a884962c (HEAD -> master) commit 3
353f27bb15d73582b11d7368eb98a6612d5781e8 text.txt second modified by old jin
0b1c2617ebd1d33a8e956744daa828ef6260f051 test.txt modified by old jin
```

版本（带括号是最新版本）

描述

5)

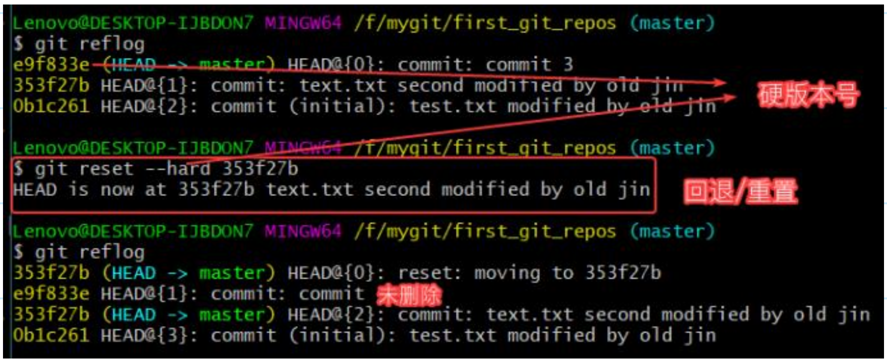


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git reflog
e9f833e (HEAD -> master) HEAD@{0}: commit: commit 3
353f27b HEAD@{1}: commit: text.txt second modified by old jin
0b1c261 HEAD@{2}: commit (initial): test.txt modified by old jin

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git log --pretty=oneline
e9f833eb353327e64dde9390b0c39649a884962c (HEAD -> master) commit 3
353f27bb15d73582b11d7368eb98a6612d5781e8 text.txt second modified by old jin
0b1c2617ebd1d33a8e956744daa828ef6260f051 test.txt modified by old jin
```

引用日志，最简短日志

6）引用日志一般配合其他命令，比如版本变化的历史操作，可以让本地文件变化

a)



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git reflog
e9f833e (HEAD -> master) HEAD@{0}: commit: commit 3
353f27b HEAD@{1}: commit: text.txt second modified by old jin
0b1c261 HEAD@{2}: commit (initial): test.txt modified by old jin

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git reset --hard 353f27b
HEAD is now at 353f27b text.txt second modified by old jin

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git reflog
353f27b (HEAD -> master) HEAD@{0}: reset: moving to 353f27b
e9f833e HEAD@{1}: commit: commit
353f27b (HEAD -> master) HEAD@{2}: commit: text.txt second modified by old jin
0b1c261 HEAD@{3}: commit (initial): test.txt modified by old jin
```

硬版本号

回退/重置

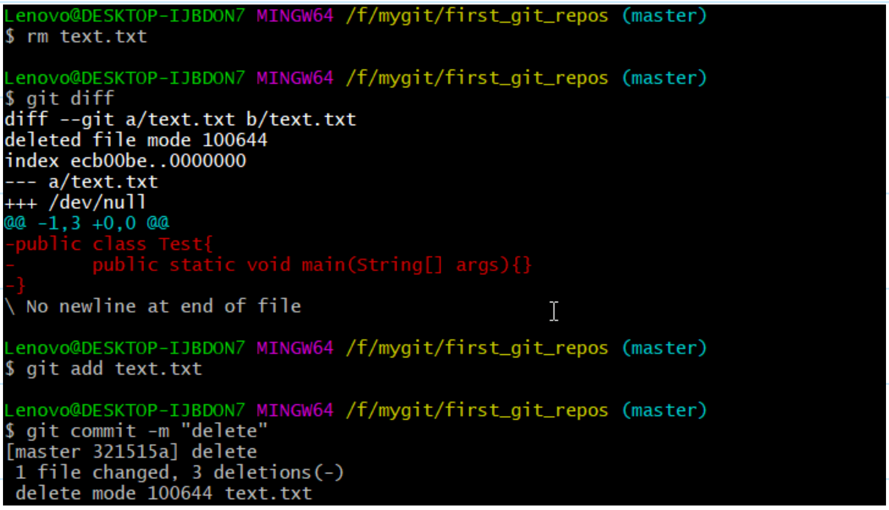
未删除

f. 删除文件

i. 兼容Linux，rm,cat查看

ii. 删除文件并且上传

iii.

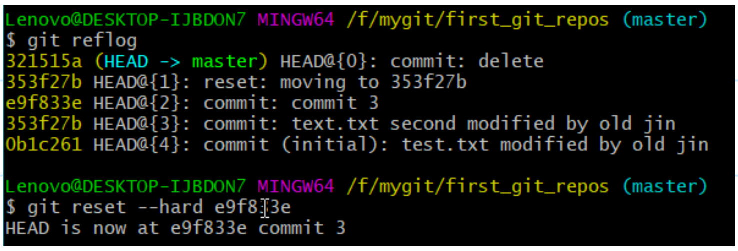


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ rm text.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git diff
diff --git a/text.txt b/text.txt
deleted file mode 100644
index ecb00be..0000000
--- a/text.txt
+++ /dev/null
@@ -1,3 +0,0 @@
-public class Test{
-       public static void main(String[] args){}
-}
\ No newline at end of file

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git add text.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git commit -m "delete"
[master 321515a] delete
 1 file changed, 3 deletions(-)
 delete mode 100644 text.txt
```

iv. 删除后找回方式

1） 历史版本

a)

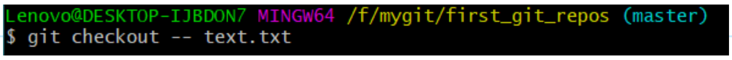


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git reflog
321515a (HEAD -> master) HEAD@{0}: commit: delete
353f27b HEAD@{1}: reset: moving to 353f27b
e9f833e HEAD@{2}: commit: commit 3
353f27b HEAD@{3}: commit: text.txt second modified by old jin
0b1c261 HEAD@{4}: commit (initial): test.txt modified by old jin

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git reset --hard e9f833e
HEAD is now at e9f833e commit 3
```

2） 用git checkout -- 文件名，检出文件

a） 删除后未提交可以用(未提交删除文件时)

b)

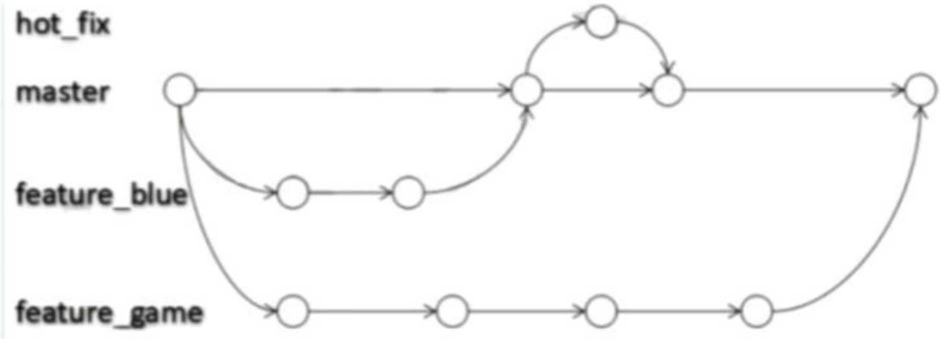


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git checkout -- text.txt
```

g. 分支操作

i. 在版本回退那里，已经知道每次提交，git都会把他们串成一条时间线，这条时间线就是一个分支。截止到目前，只有一条时间线，在git里，这个分支叫主分支，即master。切换分支即可以理解为切换时间线。

1)



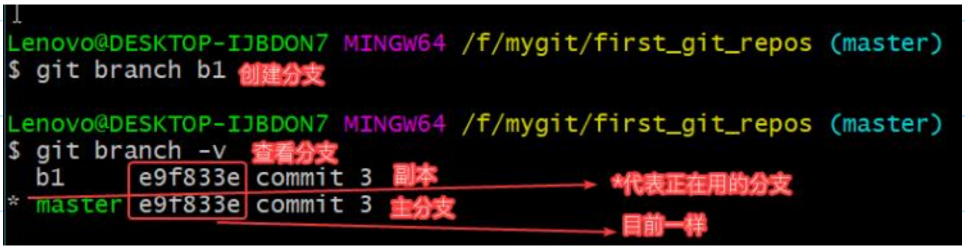
ii. 在版本控制过程中，使用多条线同时推进多个任务。每条线成为一个分支。

1） 例如

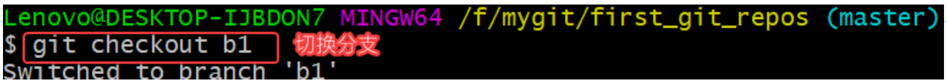
iii. 创建分支

1） git branch[分支名]

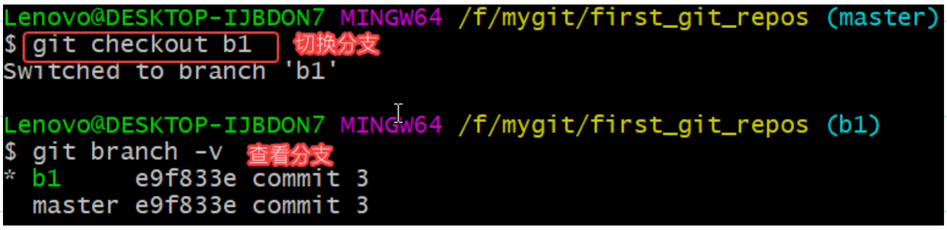
2)



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git branch b1 创建分支

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git branch -v  查看分支
  b1     e9f833e commit 3  副本        *代表正在用的分支
* master e9f833e commit 3  主分支      目前一样
```



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git checkout b1   切换分支
Switched to branch 'b1'
```

3）

```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git checkout b1
Switched to branch 'b1'

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (b1)
$ git branch -v
* b1     e9f833e commit 3
  master e9f833e commit 3
```

切换分支

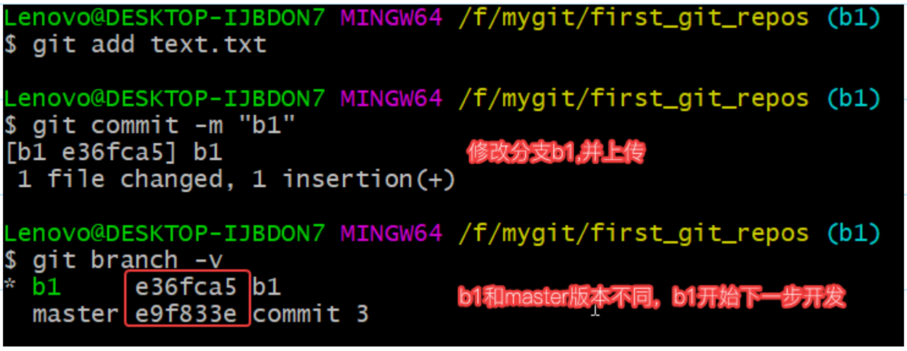
查看分支

iv. 查看分支

1）git branch -v

v. 切换分支

1）git checkout 分支

2）

```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (b1)
$ git add text.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (b1)
$ git commit -m "b1"
[b1 e36fca5] b1
 1 file changed, 1 insertion(+)

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (b1)
$ git branch -v
* b1     e36fca5 b1
  master e9f833e commit 3
```

修改分支b1,并上传

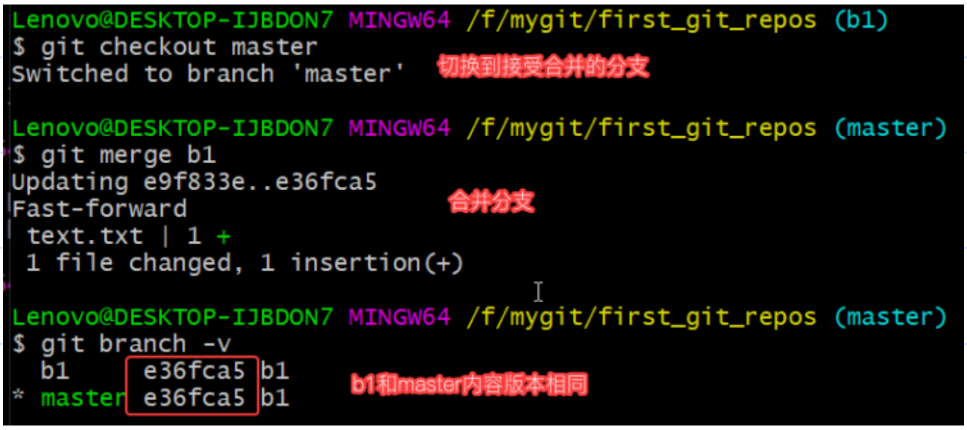
b1和master版本不同，b1开始下一步开发

vi. 合并分支

1）切换到接受合并的分支

2）git checkout 接受合并的分支

3）git merge 要合并的分支

4）

```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (b1)
$ git checkout master
Switched to branch 'master'

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git merge b1
Updating e9f833e..e36fca5
Fast-forward
 text.txt | 1 +
 1 file changed, 1 insertion(+)

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git branch -v
  b1     e36fca5 b1
* master e36fca5 b1
```

切换到接受合并的分支

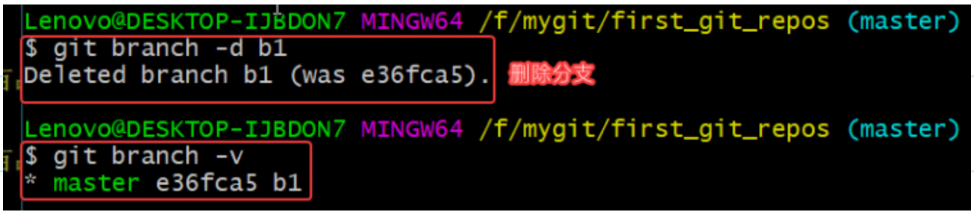
合并分支

b1和master内容版本相同

vii. 删除分支

1）git branch –d 分支名

2）不能在所属分支删除分支

3）

```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git branch -d b1
Deleted branch b1 (was e36fca5).

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ git branch -v
* master e36fca5 b1
```

删除分支

远程仓库操作

2020年6月22日 10:50

1.中国现在多用gitee（中国的）

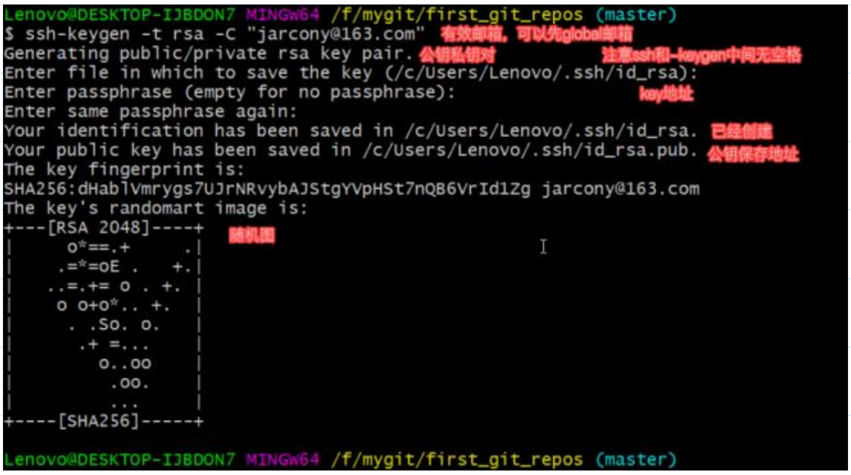
地址：http://gitee.com/

2. 创建SSH Key

a. 因为数据保存在远程服务器，服务器需要对你的身份进行识别，SSH Key可以让你的电脑和码云之间建立安全的加密连接。

b. 运行命令ssh-keygen -t rsa -C ”你的邮箱”，会有三次提示输入，直接回车即可。

i.



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ ssh-keygen -t rsa -C "jarcony@163.com"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/Lenovo/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Lenovo/.ssh/id_rsa.
Your public key has been saved in /c/Users/Lenovo/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:dHablVmrygs7UJrNRvybAJStgYVpHSt7nQB6VrId1Zg jarcony@163.com
The key's randomart image is:
+---[RSA 2048]----+
|      o*==.+    .|
|     .=*=oE .  +.|
|    ..=.+= o . +. |
|     o o+o*.. +. |
|      . .So. o.  |
|       .+ =...   |
|         o..oo   |
|          .oo.   |
|           ...   |
+----[SHA256]-----+

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
```

有效邮箱，可以先global邮箱

公钥私钥对

注意ssh和-keygen中间无空格

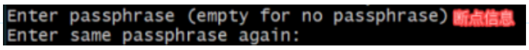
key地址

已经创建

公钥保存地址

随机图

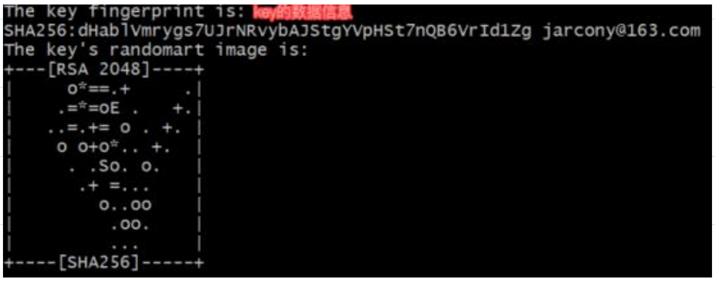
ii. 过滤的信息



```
Enter passphrase (empty for no passphrase)
Enter same passphrase again:
```

断点信息

iii.

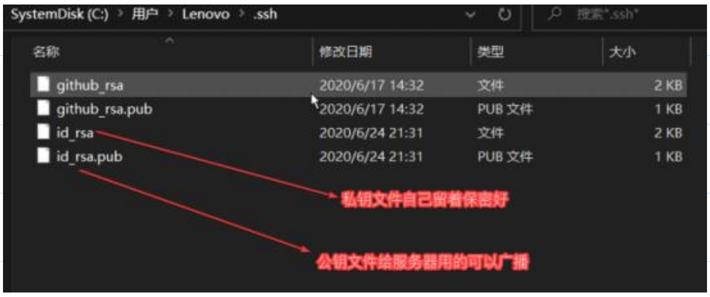


```
The key fingerprint is:
SHA256:dHablVmrygs7UJrNRvybAJStgYVpHSt7nQB6VrId1Zg jarcony@163.com
The key's randomart image is:
+---[RSA 2048]----+
|      o*==.+    .|
|     .=*=oE .  +.|
|    ..=.+= o . +. |
|     o o+o*.. +. |
|      . .So. o.  |
|       .+ =...   |
|         o..oo   |
|          .oo.   |
|           ...   |
+----[SHA256]-----+
```

key的数据信息

c. 在用户目录下生成.ssh目录，里面有一个id_rsa.pub文件，保存的就是公钥。

i.

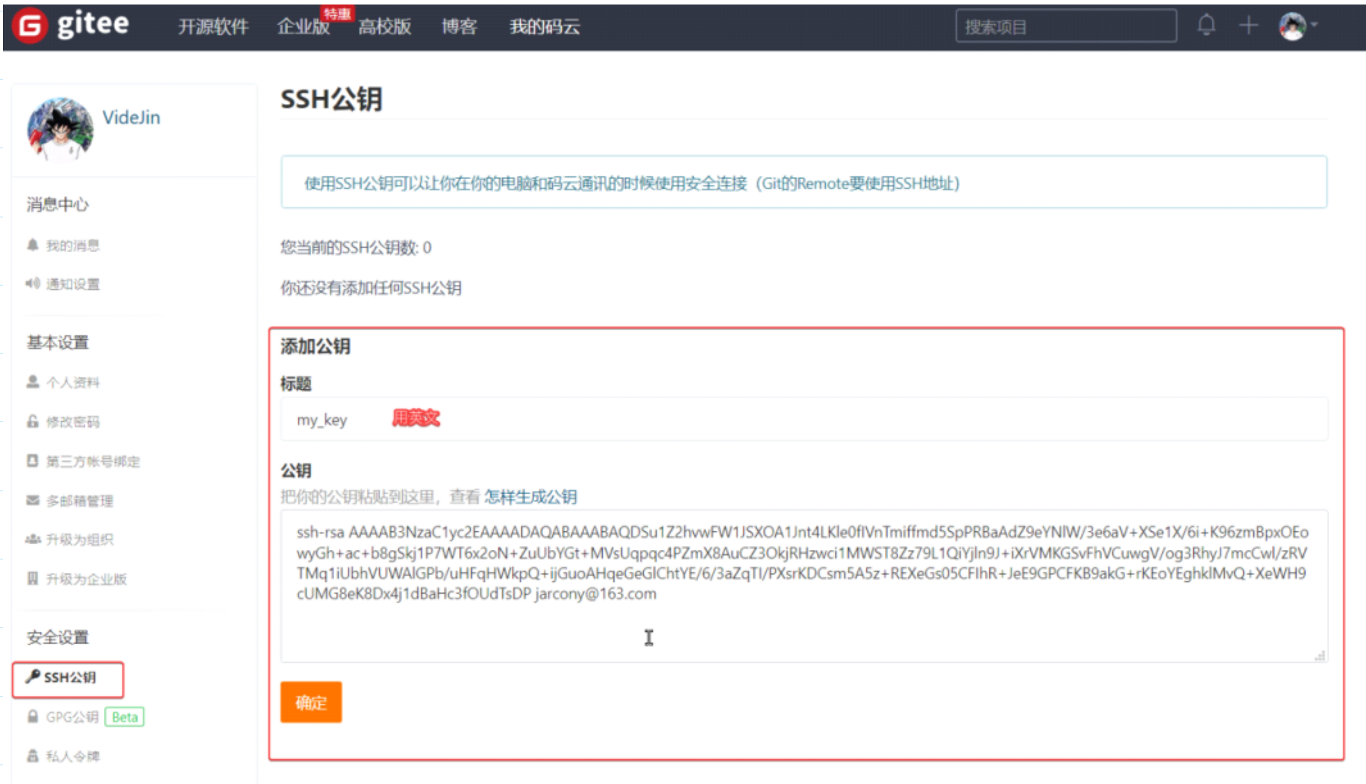


名称	修改日期	类型	大小
github_rsa	2020/6/17 14:32	文件	2 KB
github_rsa.pub	2020/6/17 14:32	PUB 文件	1 KB
id_rsa	2020/6/24 21:31	文件	2 KB
id_rsa.pub	2020/6/24 21:31	PUB 文件	1 KB

私钥文件自己留着保密好

公钥文件给服务器用的可以广播

d. 登录码云，在SSH公钥文本框里粘贴id_rsa.pub文件的内容

i. 

SSH公钥

使用SSH公钥可以让你在你的电脑和码云通讯的时候使用安全连接（Git的Remote要使用SSH地址）

您当前的SSH公钥数: 0

你还没有添加任何SSH公钥

添加公钥

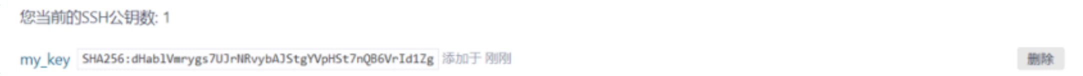
标题

my_key

公钥

把你的公钥粘贴到这里，查看 怎样生成公钥

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDSu1Z2hvwFW1JSXOA1Jnt4LKle0flVnTmiffmd5SpPRBaAdZ9eYNlW/3e6aV+XSe1X/6i+K96zmBpxOEo
wyGh+ac+b8gSkj1P7WT6x2oN+ZuUbYGt+MVsUqpqc4PZmX8AuCZ3OkjRHzwci1MWST8Zz79L1QiYjln9J+iXrVMKGSvFhVCuwgV/og3RhyJ7mcCwl/zRV
TMq1iUbhVUWAlGPb/uHFqHWkpQ+ijGuoAHqeGeGlChtYE/6/3aZqTl/PXsrKDCsm5A5z+REXeGs05CFlhR+JeE9GPCFKB9akG+rKEoYEghklMvQ+XeWH9
cUMG8eK8Dx4j1dBaHc3fOUdTsDP jarcony@163.com
```

ii. 

您当前的SSH公钥数: 1

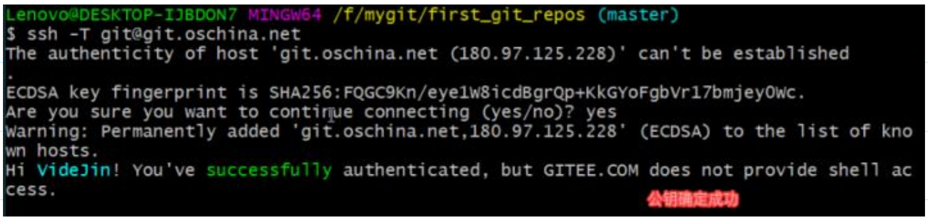
my_key SHA256:dHablVmrygs7UJrNRvybAJStgYVpHSt7nQB6VrId1Zg 添加于 刚刚 删除

e. 验证密钥是否添加成功

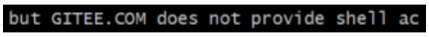
i. 执行命令 ssh -T git@git.oschina.net

ii. git使用SSH连接第一次验证服务器的key时，需要你进行确认，此时输入yes回车即可。

iii. 再次执行 ssh -T git@git.oschina.net

iv. 

```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/first_git_repos (master)
$ ssh -T git@git.oschina.net
The authenticity of host 'git.oschina.net (180.97.125.228)' can't be established
.
ECDSA key fingerprint is SHA256:FQGC9Kn/eye1W8icdBgrQp+KkGYoFgbVr17bmjeyOWc.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'git.oschina.net,180.97.125.228' (ECDSA) to the list of kno
wn hosts.
Hi VideJin! You've successfully authenticated, but GITEE.COM does not provide shell ac
cess.
```

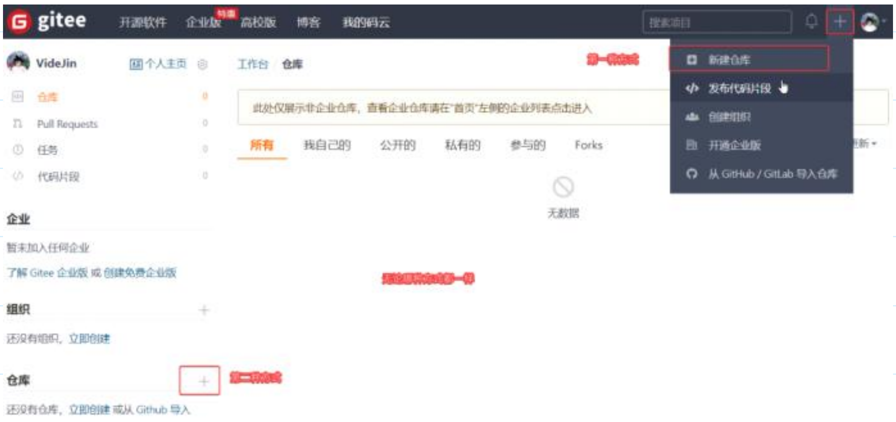
v. 

```
but GITEE.COM does not provide shell ac
```

1）但不支持加密协议

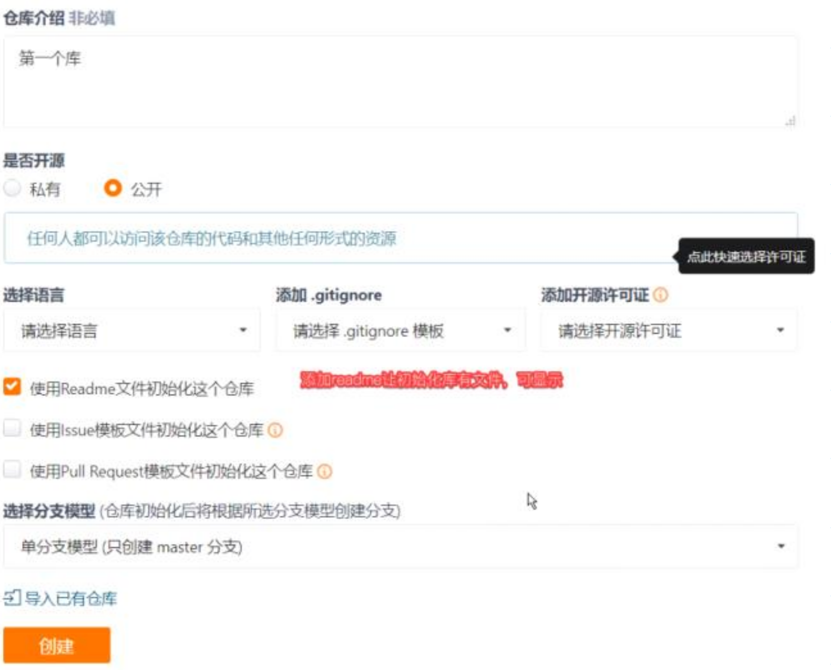
3、创建远程仓库

a. 在码云中创建仓库

i. 

此处仅展示非企业仓库，查看企业仓库请在"首页"左侧的企业列表点击进入

ii.



仓库介绍 非必填

第一个库

是否开源

○ 私有 ● 公开

任何人都可以访问该仓库的代码和其他任何形式的资源

选择语言: 请选择语言

添加 .gitignore: 请选择 .gitignore 模板

添加开源许可证: 请选择开源许可证

☑ 使用Readme文件初始化这个仓库

添加readme让初始化库有文件，可显示

□ 使用Issue模板文件初始化这个仓库

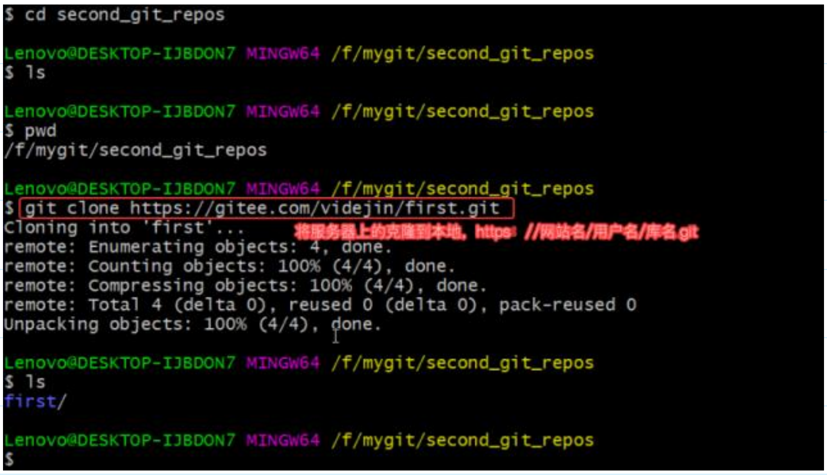
□ 使用Pull Request模板文件初始化这个仓库

选择分支模型 (仓库初始化后将根据所选分支模型创建分支)

单分支模型 (只创建 master 分支)

b. 克隆到本地

i.

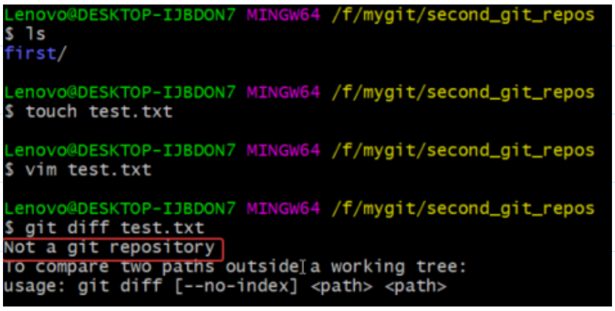


```
$ cd second_git_repos

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ ls

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ pwd
/f/mygit/second_git_repos

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ git clone https://gitee.com/videjin/first.git
Cloning into 'first'...
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
Unpacking objects: 100% (4/4), done.

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ ls
first/

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$
```

将服务器上的克隆到本地，https：//网站名/用户名/库名.git

ii. 新建的文件必须add到git，才能够被git知道有这个文件，并记录版本

1）

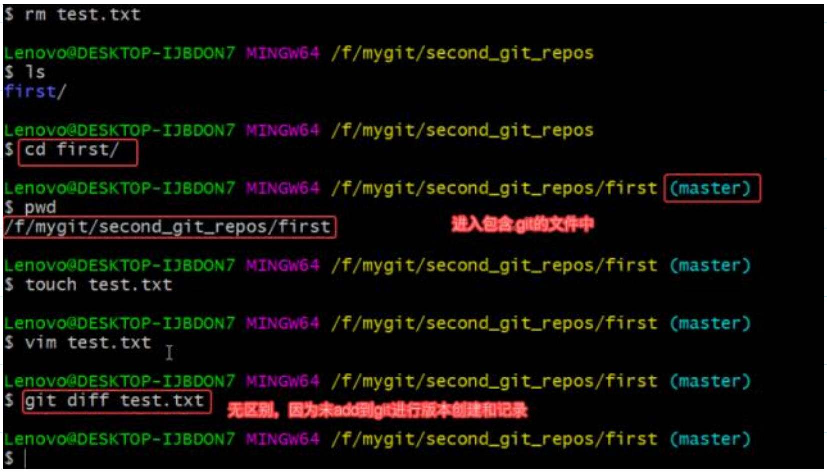


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ ls
first/

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ touch test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ vim test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ git diff test.txt
Not a git repository
To compare two paths outside a working tree:
usage: git diff [--no-index] <path> <path>
```

a）这不是在.git文件中创建的故出现这个问题

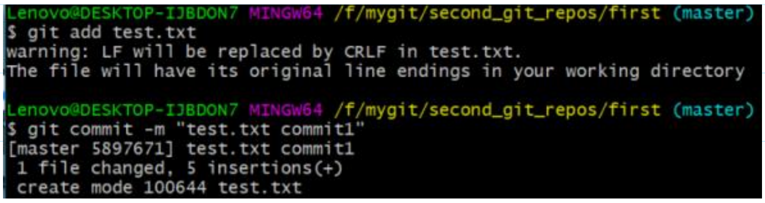
b）解决是打开first（有.git的文件），再创建

i)



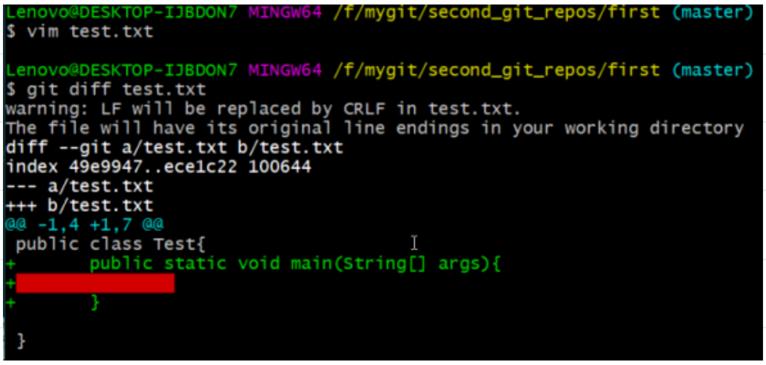
```
$ rm test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ ls
first/

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos
$ cd first/

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ pwd
/f/mygit/second_git_repos/first                进入包含git的文件中

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ touch test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ vim test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git diff test.txt   无区别，因为未add到git进行版本创建和记录

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ |
```

ii)



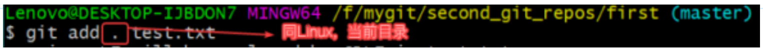
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git add test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git commit -m "test.txt commit1"
[master 5897671] test.txt commit1
 1 file changed, 5 insertions(+)
 create mode 100644 test.txt
```

iii)



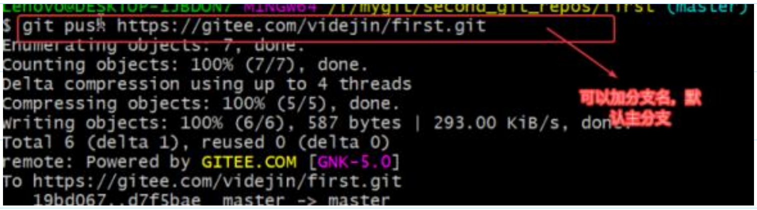
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ vim test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git diff test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory
diff --git a/test.txt b/test.txt
index 49e9947..ece1c22 100644
--- a/test.txt
+++ b/test.txt
@@ -1,4 +1,7 @@
 public class Test{
+       public static void main(String[] args){
+[redacted]
+       }

 }
```

iv)



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git add . test.txt  同Linux，当前目录
```

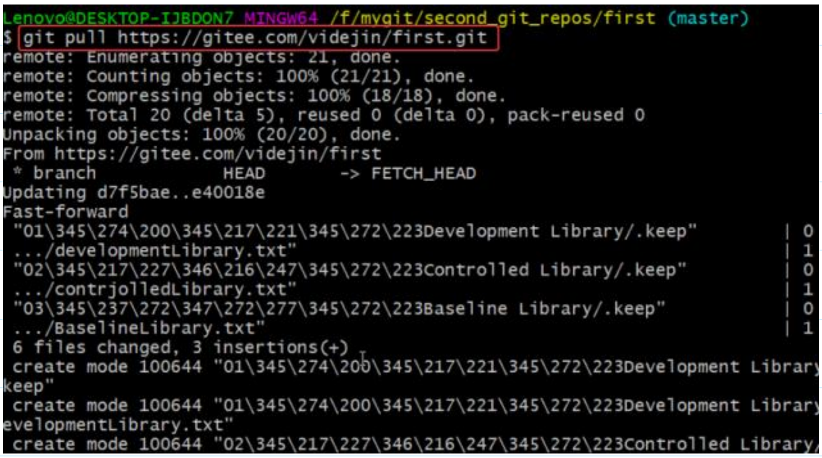
c. 上传，同步到服务器，push->推送



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git push https://gitee.com/videjin/first.git
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads               可以加分支名，默
Compressing objects: 100% (5/5), done.                认主分支
Writing objects: 100% (6/6), 587 bytes | 293.00 KiB/s, done.
Total 6 (delta 1), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-5.0]
To https://gitee.com/videjin/first.git
   19bd067..d7f5bae  master -> master
```

d. 从服务区拉取到本地，这是有库了不用clone，用pull->拉取

i.

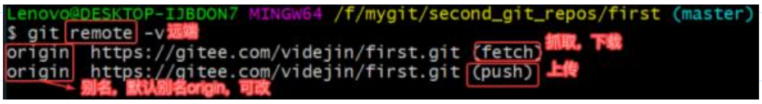


```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git pull https://gitee.com/videjin/first.git
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (18/18), done.
remote: Total 20 (delta 5), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (20/20), done.
From https://gitee.com/videjin/first
 * branch            HEAD       -> FETCH_HEAD
Updating d7f5bae..e40018e
Fast-forward
 "01\345\274\200\345\217\221\345\272\223Development Library/.keep"       | 0
 .../developmentLibrary.txt"                                              | 1
 "02\345\217\227\346\216\247\345\272\223Controlled Library/.keep"        | 0
 .../contrjolledLibrary.txt"                                              | 1
 "03\345\237\272\347\272\277\345\272\223Baseline Library/.keep"          | 0
 .../BaselineLibrary.txt"                                                 | 1
 6 files changed, 3 insertions(+)
 create mode 100644 "01\345\274\200\345\217\221\345\272\223Development Library
keep"
 create mode 100644 "01\345\274\200\345\217\221\345\272\223Development Library
evelopmentLibrary.txt"
 create mode 100644 "02\345\217\227\346\216\247\345\272\223Controlled Library
```

4. 创建远程库地址别名（用的少）

i. git remote -v 查看当前所有远程地址别名

1）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git remote -v 远端
origin  https://gitee.com/videjin/first.git (fetch) 抓取，下载
origin  https://gitee.com/videjin/first.git (push)  上传
  别名，默认别名origin，可改
```

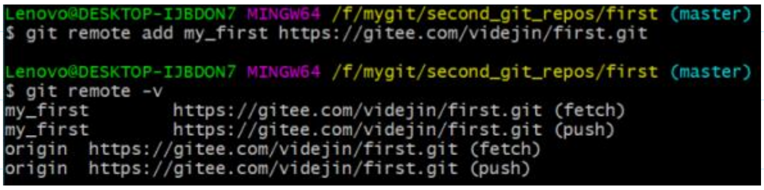
2）原来推送的时候是

a） git push https://gitee.com/videjin/first.git master

b） 改了别名之后 别名可以代替路径地址

ii. git remote add 别名 远程地址 （添加别名）

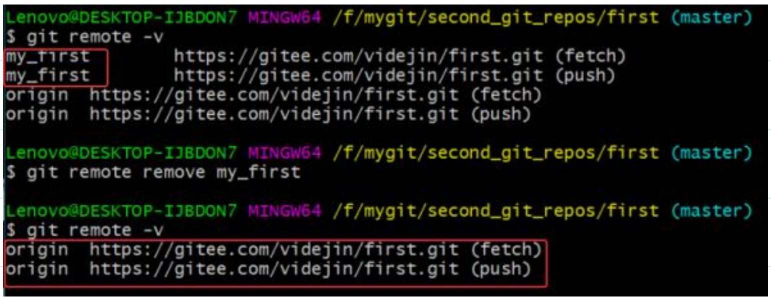
a）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git remote add my_first https://gitee.com/videjin/first.git

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git remote -v
my_first        https://gitee.com/videjin/first.git (fetch)
my_first        https://gitee.com/videjin/first.git (push)
origin  https://gitee.com/videjin/first.git (fetch)
origin  https://gitee.com/videjin/first.git (push)
```

iii. git remote remove 别名 （删除别名）

1）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git remote -v
my_first        https://gitee.com/videjin/first.git (fetch)
my_first        https://gitee.com/videjin/first.git (push)
origin  https://gitee.com/videjin/first.git (fetch)
origin  https://gitee.com/videjin/first.git (push)

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git remote remove my_first

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git remote -v
origin  https://gitee.com/videjin/first.git (fetch)
origin  https://gitee.com/videjin/first.git (push)
```

5. 解决冲突

a. 要点

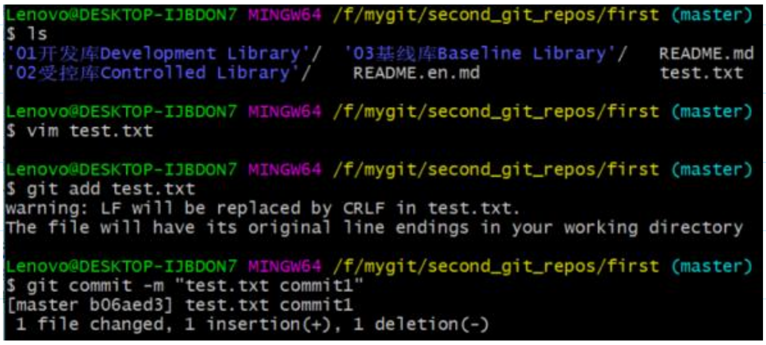
i. 如果不是基于GitHub远程库的最新版所做的修改，不能推送，必须先拉取。

ii. 拉取下来后如果进入冲突状态，则修改->add->commit即可。

b. 问题

i. 本地进行修改并更新版本，但不上传到服务器

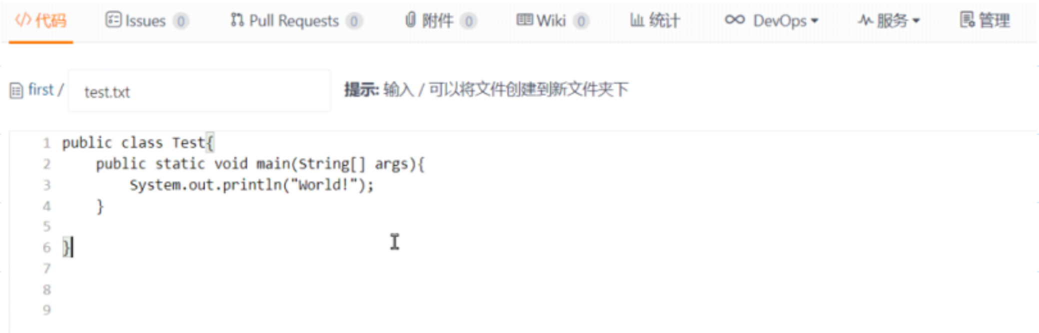
1）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ ls
'01开发库Development Library'/   '03基线库Baseline Library'/   README.md
'02受控库Controlled Library'/    README.en.md                 test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ vim test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git add test.txt
warning: LF will be replaced by CRLF in test.txt.
The file will have its original line endings in your working directory

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git commit -m "test.txt commit1"
[master b06aed3] test.txt commit1
 1 file changed, 1 insertion(+), 1 deletion(-)
```

ii. 服务器端进行修改

1）



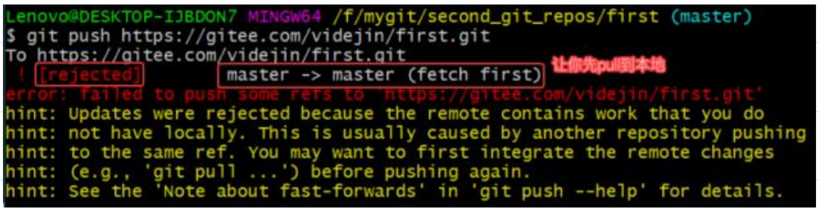
first / test.txt

提示: 输入 / 可以将文件创建到新文件夹下

```
public class Test{
    public static void main(String[] args){
        System.out.println("World!");
    }

}
```

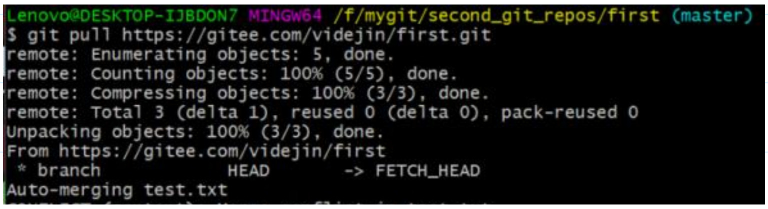
iii. 本地和服务器版本冲突

1）



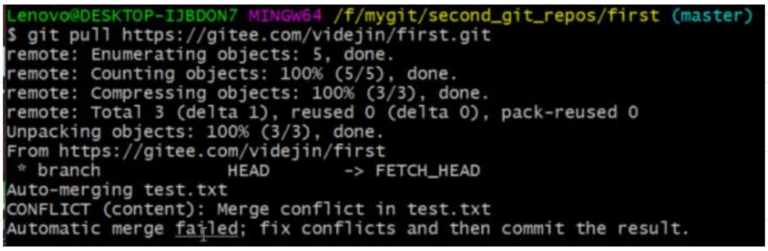
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git push https://gitee.com/videjin/first.git
To https://gitee.com/videjin/first.git
 ! [rejected]        master -> master (fetch first)   让你先pull到本地
error: failed to push some refs to 'https://gitee.com/videjin/first.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

2）



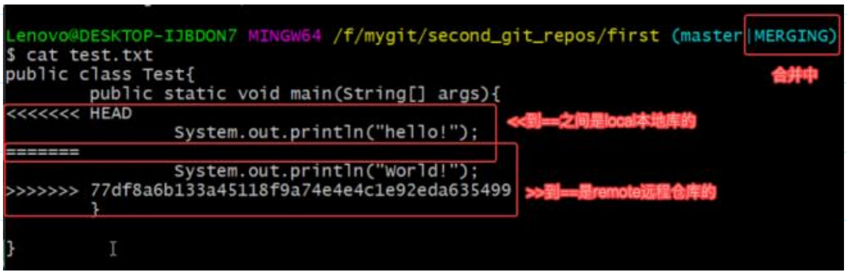
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git pull https://gitee.com/videjin/first.git
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://gitee.com/videjin/first
 * branch            HEAD       -> FETCH_HEAD
Auto-merging test.txt
```

2）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git pull https://gitee.com/videjin/first.git
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://gitee.com/videjin/first
 * branch            HEAD       -> FETCH_HEAD
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.
```

3）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master|MERGING)
$ cat test.txt
public class Test{
        public static void main(String[] args){
<<<<<<< HEAD
                System.out.println("hello!");
=======
                System.out.println("world!");
>>>>>>> 77df8a6b133a45118f9a74e4e4c1e92eda635499
        }

}
```

合并中

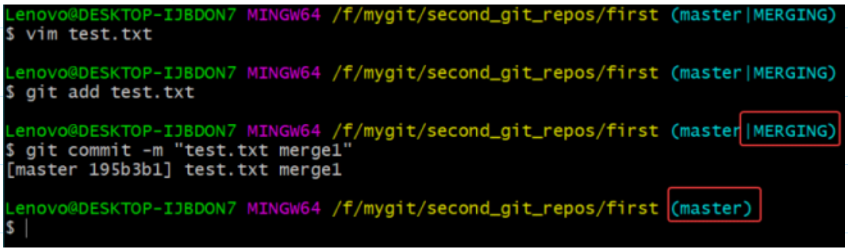
<<到==之间是local本地库的

>>到==是remote远程仓库的

4）要进行合并，直接自己手动改成认为对的，删去另一部分，再上传就可以

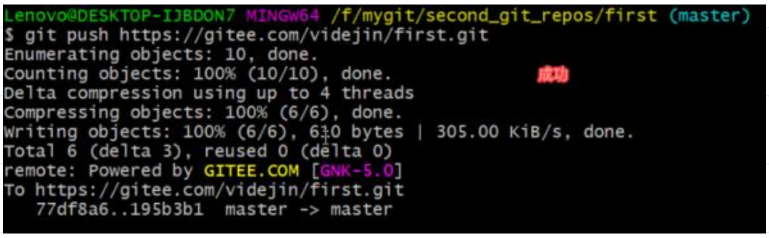
5）merging是创建了一个分支，等你合并之后就回到冲突分支（这里是主分支）

a）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master|MERGING)
$ vim test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master|MERGING)
$ git add test.txt

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master|MERGING)
$ git commit -m "test.txt merge1"
[master 195b3b1] test.txt merge1

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ |
```

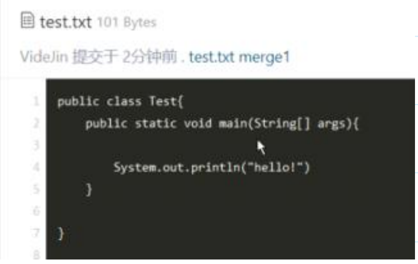
b）



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git push https://gitee.com/videjin/first.git
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 610 bytes | 305.00 KiB/s, done.
Total 6 (delta 3), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-5.0]
To https://gitee.com/videjin/first.git
   77df8a6..195b3b1  master -> master
```

成功

c）

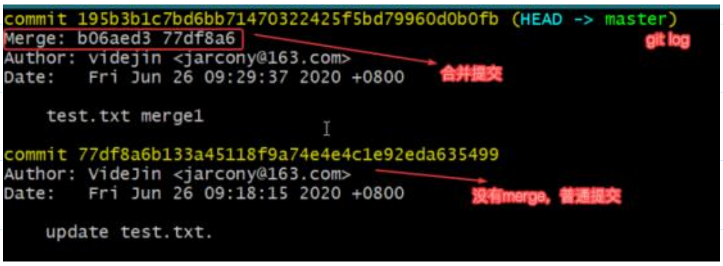


test.txt 101 Bytes

VideJin 提交于 2分钟前 . test.txt merge1

```
public class Test{
    public static void main(String[] args){

        System.out.println("hello!")
    }

}
```

6）



```
commit 195b3b1c7bd6bb71470322425f5bd79960d0b0fb (HEAD -> master)
Merge: b06aed3 77df8a6
Author: videjin <jarcony@163.com>
Date:   Fri Jun 26 09:29:37 2020 +0800

    test.txt merge1

commit 77df8a6b133a45118f9a74e4e4c1e92eda635499
Author: VideJin <jarcony@163.com>
Date:   Fri Jun 26 09:18:15 2020 +0800

    update test.txt.
```

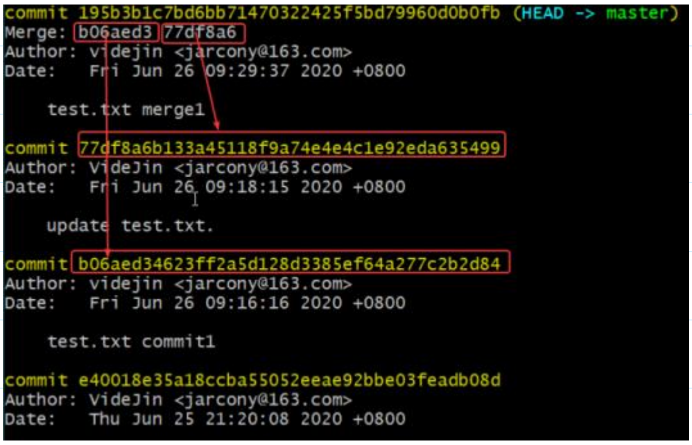
git log

合并提交

没有merge，普通提交

a）但凡合并都开了临时分支，一但合并就回到原来分支上

b）合并分支将下边的分支进行合并（从log中体现）



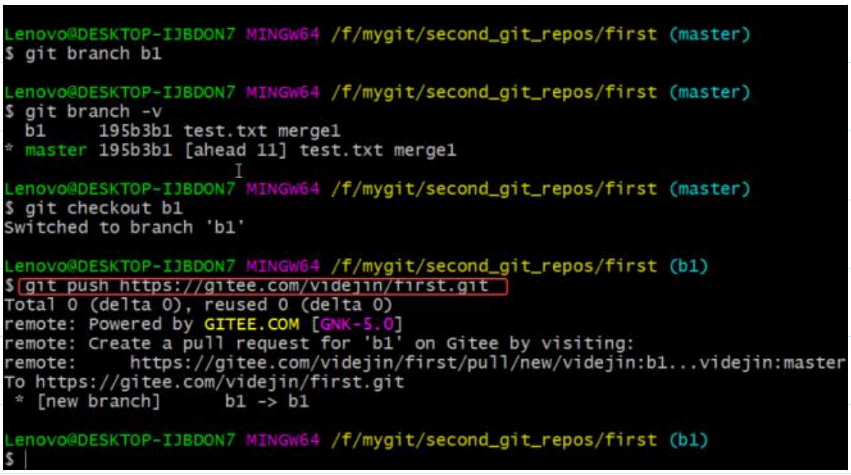
```
commit 195b3b1c7bd6bb71470322425f5bd79960d0b0fb (HEAD -> master)
Merge: b06aed3 77df8a6
Author: videjin <jarcony@163.com>
Date:   Fri Jun 26 09:29:37 2020 +0800

    test.txt merge1

commit 77df8a6b133a45118f9a74e4e4c1e92eda635499
Author: VideJin <jarcony@163.com>
Date:   Fri Jun 26 09:18:15 2020 +0800

    update test.txt.

commit b06aed34623ff2a5d128d3385ef64a277c2b2d84
Author: videjin <jarcony@163.com>
Date:   Fri Jun 26 09:16:16 2020 +0800

    test.txt commit1

commit e40018e35a18ccba55052eeae92bbe03feadb08d
Author: VideJin <jarcony@163.com>
Date:   Thu Jun 25 21:20:08 2020 +0800
```

7)

6. 远程仓库创建分支

a. 可在本地创建分支，然后push到服务器端，服务器端的分支与主分支相同，因为相当于副本

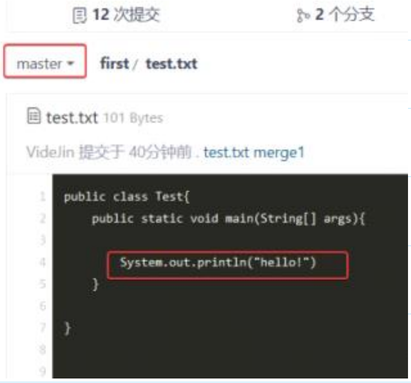
i.



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git branch b1

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git branch -v
  b1     195b3b1 test.txt merge1
* master 195b3b1 [ahead 11] test.txt merge1

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git checkout b1
Switched to branch 'b1'

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (b1)
$ git push https://gitee.com/videjin/first.git
Total 0 (delta 0), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-5.0]
remote: Create a pull request for 'b1' on Gitee by visiting:
remote:     https://gitee.com/videjin/first/pull/new/videjin:b1...videjin:master
To https://gitee.com/videjin/first.git
 * [new branch]      b1 -> b1

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (b1)
$ 
```

b. 分支文件合并

i.

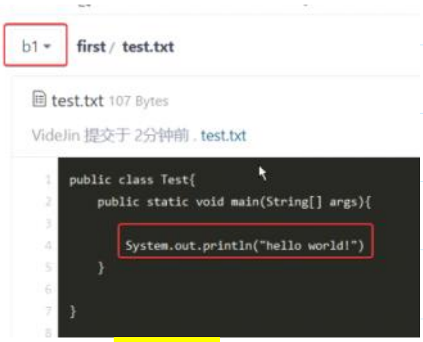


test.txt 101 Bytes

VideJin 提交于 40分钟前 . test.txt merge1

```
public class Test{
    public static void main(String[] args){

        System.out.println("hello!")
    }

}
```

ii.



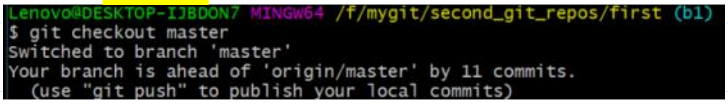
test.txt 107 Bytes

VideJin 提交于 2分钟前 . test.txt

```
public class Test{
    public static void main(String[] args){

        System.out.println("hello world!")
    }

}
```

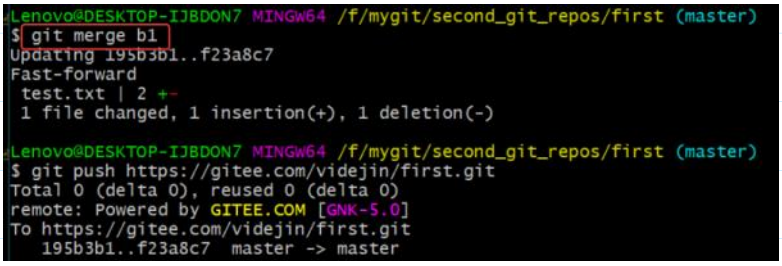
iii. 切换到需要的分支git checkout master

1)



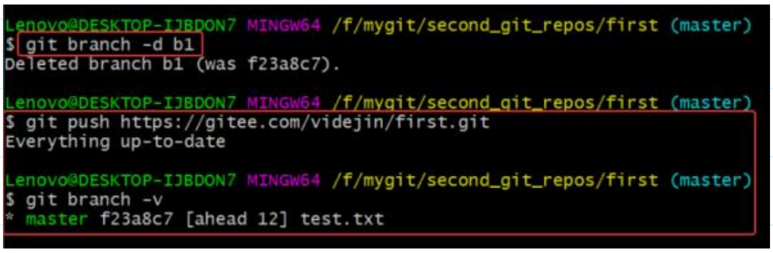
```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (b1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 11 commits.
  (use "git push" to publish your local commits)
```

iv. 然后进行合并



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git merge b1
Updating 195b3b1..f23a8c7
Fast-forward
 test.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git push https://gitee.com/videjin/first.git
Total 0 (delta 0), reused 0 (delta 0)
remote: Powered by GITEE.COM [GNK-5.0]
To https://gitee.com/videjin/first.git
   195b3b1..f23a8c7  master -> master
```

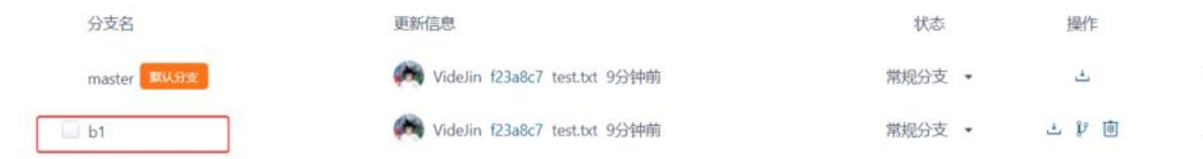
v. 合并完之后删除分支



```
Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git branch -d b1
Deleted branch b1 (was f23a8c7).

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git push https://gitee.com/videjin/first.git
Everything up-to-date

Lenovo@DESKTOP-IJBDON7 MINGW64 /f/mygit/second_git_repos/first (master)
$ git branch -v
* master f23a8c7 [ahead 12] test.txt
```

1） 这样只是删除了本地的分支，远程仓库的分支还在

a）



分支名	更新信息	状态	操作
master 默认分支	VideJin f23a8c7 test.txt 9分钟前	常规分支 ▾	
□ b1	VideJin f23a8c7 test.txt 9分钟前	常规分支 ▾	

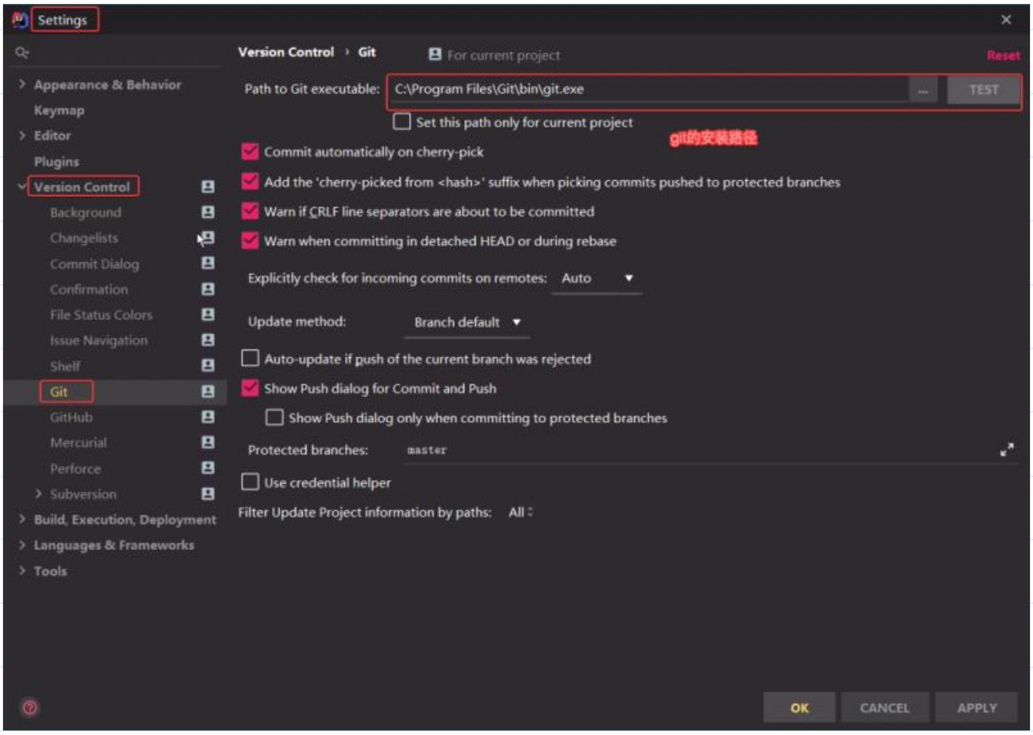
2） 不能本地删除远程的分支，只能在远程服务端进行删除

IDEA中操作git

2020年6月22日 10:50

1. 服务器端创建一个新的仓库
2. 在idea中配置git

a.



Path to Git executable: C:\Program Files\Git\bin\git.exe

git的安装路径

□ Set this path only for current project

☑ Commit automatically on cherry-pick

☑ Add the 'cherry-picked from <hash>' suffix when picking commits pushed to protected branches

☑ Warn if CRLF line separators are about to be committed

☑ Warn when committing in detached HEAD or during rebase

Explicitly check for incoming commits on remotes: Auto

Update method: Branch default

□ Auto-update if push of the current branch was rejected

☑ Show Push dialog for Commit and Push

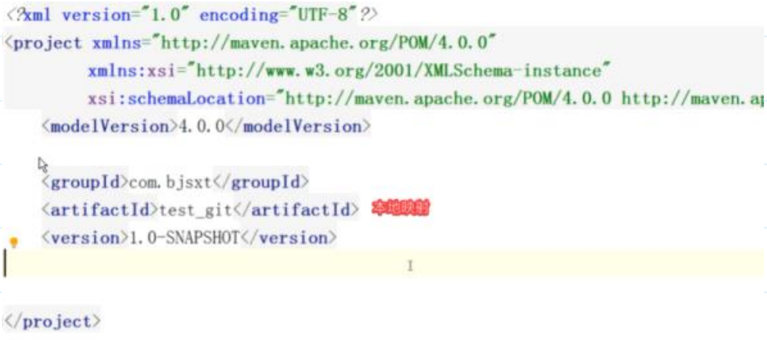
□ Show Push dialog only when committing to protected branches

Protected branches: master

□ Use credential helper

Filter Update Project information by paths: All

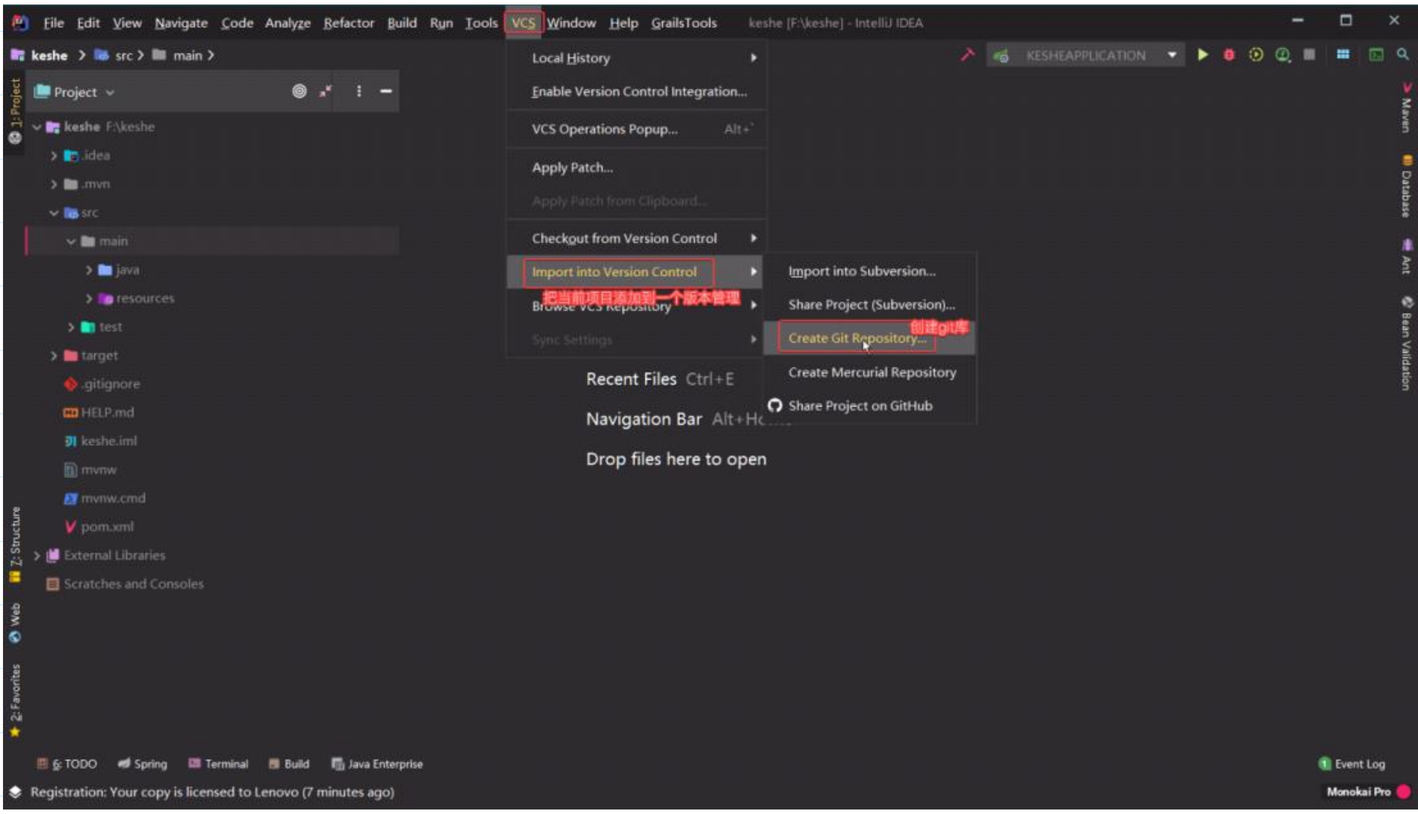
b.



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.a
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.bjsxt</groupId>
    <artifactId>test_git</artifactId>
    <version>1.0-SNAPSHOT</version>


</project>
```

本地映射

c.



把当前项目添加到一个版本管理

创建git库

Registration: Your copy is licensed to Lenovo (7 minutes ago)

i、相当于git init，本地仓库的初始化

d.

e、是