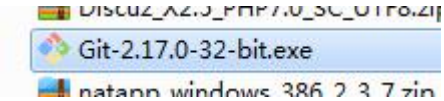
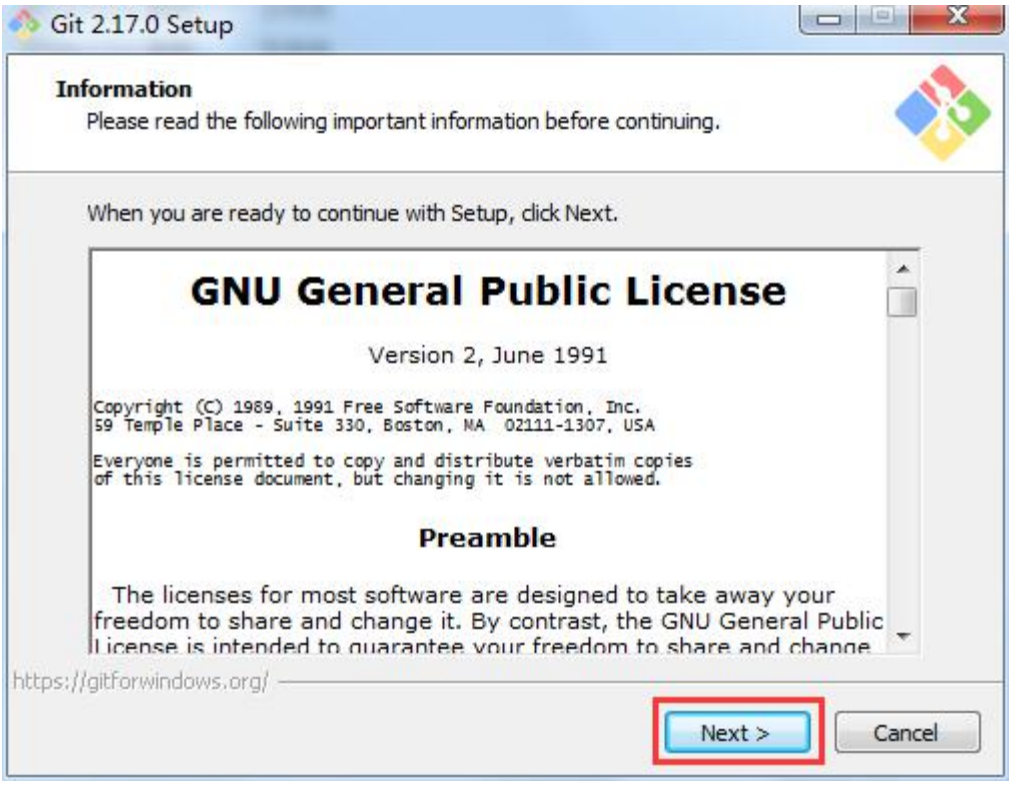
1. **GitHub 是一个面向开源及私有软件项目的托管平台，因为只支持 git 作为唯一的版本库格式进行托管，故名 GitHub。**

Git 分布式版本控制软件；
Svn \hg \git
SVN 集中式版本控制软件

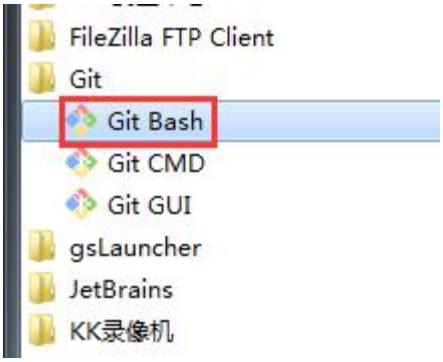
2. 下载 Windows 版 Git

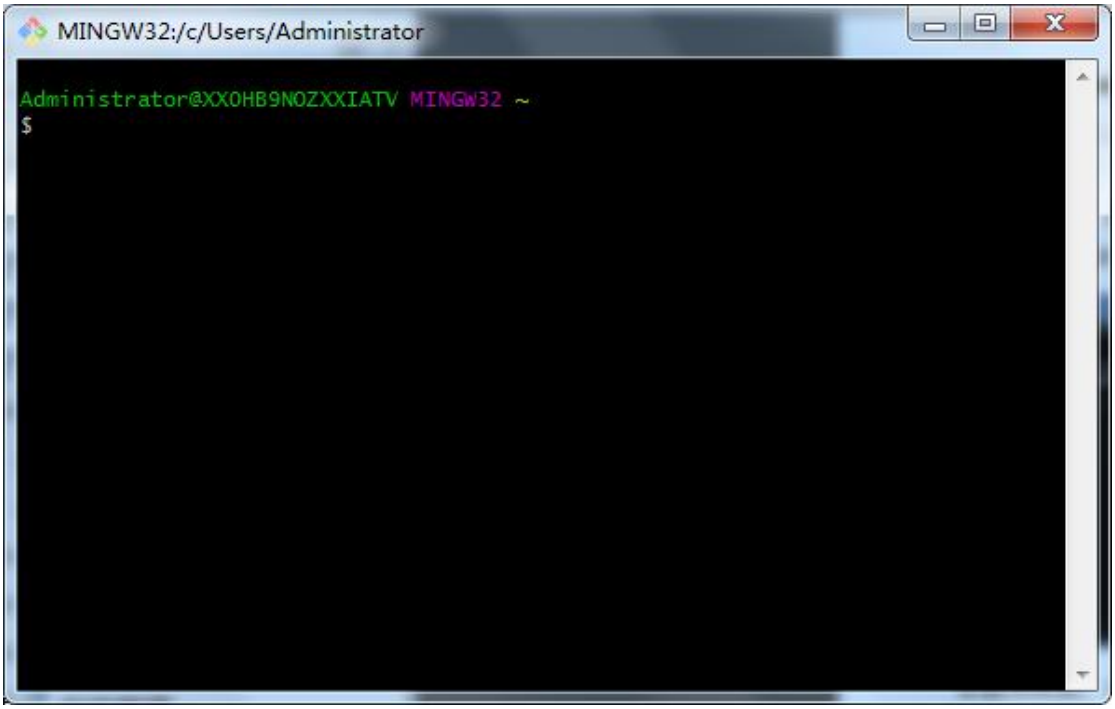


3. 双击开始安装

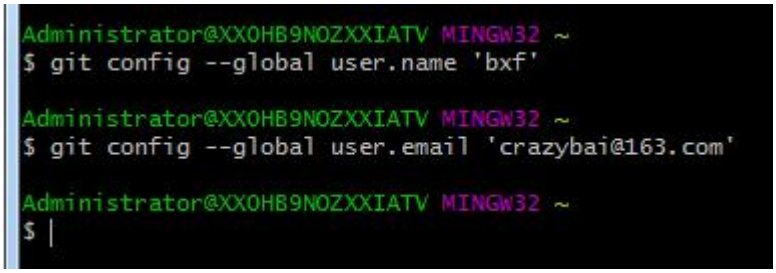


4. 打开 Git Bash





5. 创建用户（指定用户名、邮箱）



6. 创建版本库

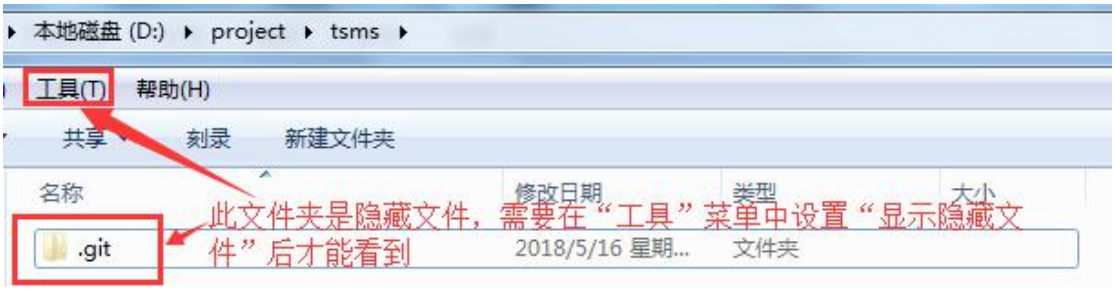
1) 创建空目录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /e
$ cd d:

Administrator@XXOHB9NOZXXIATV MINGW32 /d
$ cd project

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project
$ mkdir tsms

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project
$ cd tsms

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms
$
```

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms
$ pwd
/d/project/tsms
```

2) 使用 git init 命令将上述目录变成 git 可管理的仓库

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms
$ git init
Initialized empty Git repository in D:/project/tsms/.git/
```



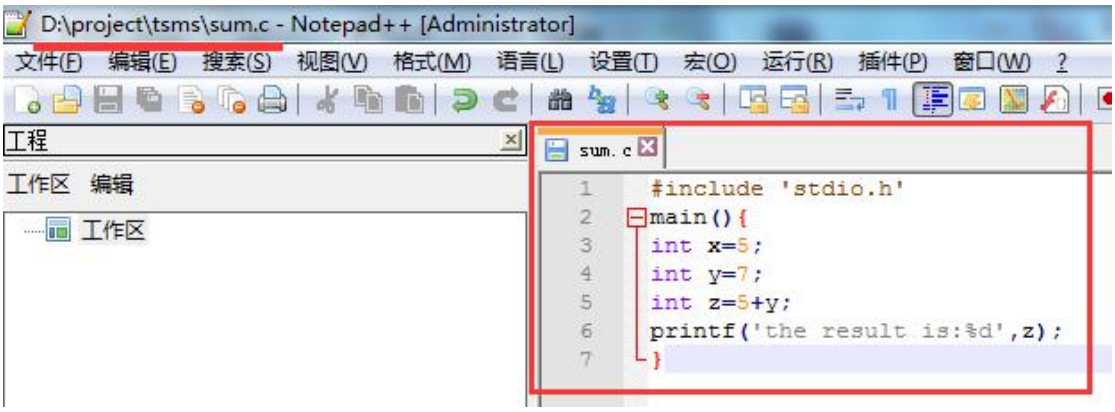
.git 目录是隐藏目录，使用 ls –ah 命令可以看到此目录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ ls

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ ls -ah
./  ../  .git/
```

7. 将文件添加到版本库

1) 在仓库中创建文件

在 d:/project/tsms 目录中（此目录是 Git 仓库，在仓库中的文件，才能由 Git 管理）创建 sum.c 文件。



注意，不要使用 Windows 的记事本创建文件，记事本创建的文件会出现格式或文本混乱的问题，可以使用其他编辑器创建文件，如 NotePad++、sublime 等编辑器。

2) 将文件添加到暂存区

3) 提交文件

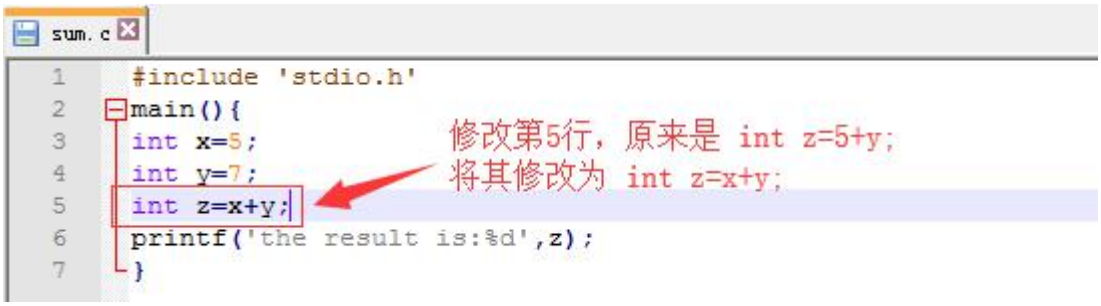
使用 git commit 命令通知 Git，将文件提交到仓库。

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git commit -m 'this program is a C program'
[master (root-commit) 2ae02ec] this program is a C program
 1 file changed, 7 insertions(+)
 create mode 100644 sum.c

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

命令执行完毕后，提示 1 个文件被改动，插入了 7 行内容

8. 修改文件、查看修改前后文件差异、查看仓库状态、提交修改

1) 修改 sum.c 文件内容



2) 查看仓库当前状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (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:   sum.c

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

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

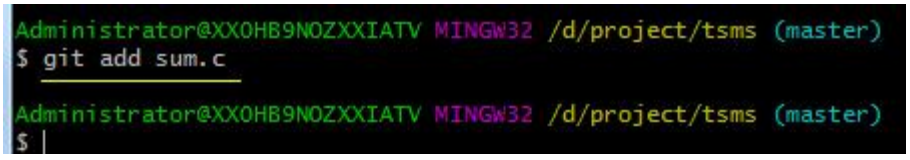
命令执行完毕后，可以看到仓库状态信息：
在 master 主分支修改了文件，
但修改内容并未提交，
修改的文件是：sum.c

3) **查看文件修改前后的差异**

4) **提交修改**

提交修改 和提交新文件步骤一样，执行 git add 命令和 git commit 命令。

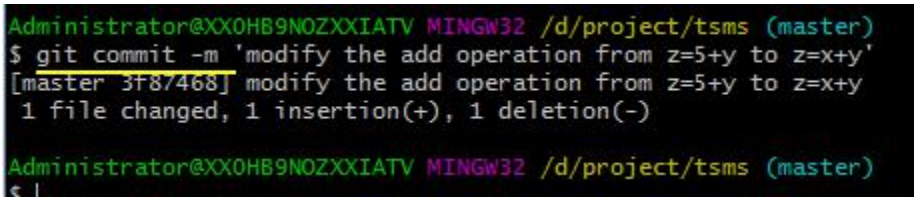
a) 将修改的文件添加到仓库



b) 查看添加到仓库后的状态（提交前）

git status 命令执行完毕后，提示我们“将要被提交的修改（还未提交）”是 sum.c 文件。

c) 提交文件



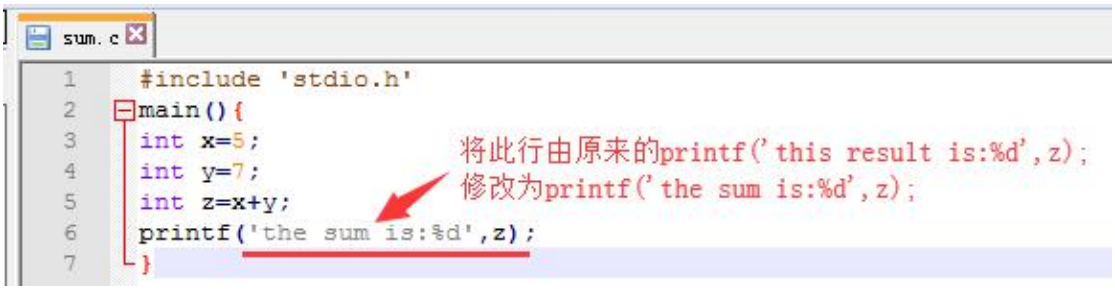
d) 查看提交后的状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```

提示，当前没有需要提交的修改，工作目录是干净的。

9. 版本回退

1) 再次修改文件并提交



a) 添加

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git add sum.c
```

b) 查看仓库状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
```

c) 提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git commit -m 'modify code from the result is... to the sum is ...'
[master 38ccf7e] modify code from the result is... to the sum is ...
 1 file changed, 1 insertion(+), 1 deletion(-)
```

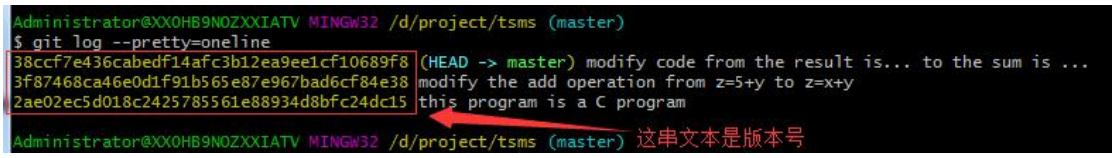
d) 查看仓库状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```

2) 查看历史修改记录

使用 git　log 命令查看历史修改记录。

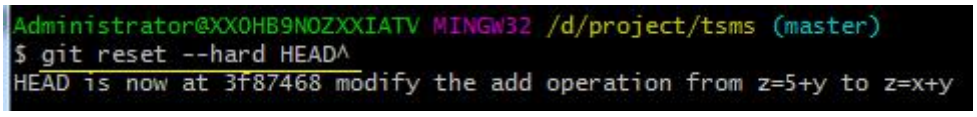
可以在 git log 命令后加参数 --pretty=oneline 简化输出信息

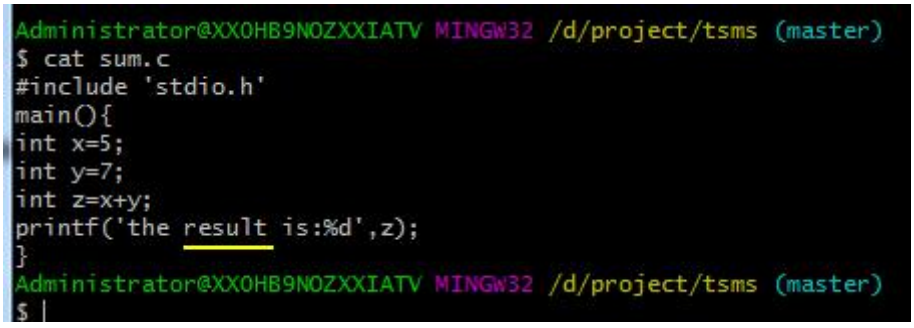


3) **回退到上一版本**

a) 使用 cat 命令查看当前文件内容

b) 回退到上一版本“from z=x+y”





sum.c

```
1  #include 'stdio.h'
2  main(){
3  int x=5;
4  int y=7;
5  int z=x+y;
6  printf('the result is:%d',z);
7  }
```

HEAD 表示当前版本，上一个版本是 HEAD^，再上一个版本是 HEAD^^，HEAD~50 表示上 50 个版本。

c) 查看当前版本库状态

可以看到最新的版本库已经不见了

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git log
commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38 (HEAD -> master)
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

d) 回到未来的某个版本（上面消失不见的之前修改的最近的版本库）

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git reset --hard 38ccf7
HEAD is now at 38ccf7e modify code from the result is... to the sum is ...

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

使用版本号（commit id）回到未来的版本，版本号无需写全，只需写前几位即可（能够与其他版本号区别开来即可）。

查看文件内容，可以看到版本已经恢复了

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ cat sum.c
#include 'stdio.h'
main(){
int x=5;
int y=7;
int z=x+y;
printf('the sum is:%d',z);
}
```

sum.c

```
1  #include 'stdio.h'
2  main(){
3  int x=5;
4  int y=7;
5  int z=x+y;
6  printf('the sum is:%d',z);
7  }
```

e) 查看历史修改记录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git log
commit 38ccf7e436cabedf14afc3b12ea9ee1cf10689f8 (HEAD -> master)
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:54:01 2018 +0800

    modify code from the result is... to the sum is ...

commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

4) **使用 git reflog 命令查看历次的操作命令历史记录**

当回退到某个版本后，关闭了 Git Bash，此时就看不见了之前的版本号了，我们可以使用 git reflog 命令查看历次操作记录, 在这个操作记录中可以看到版本号, 这样就可以使用“git reset –hard 版本号” 命令恢复到未来的版本库了。

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git reflog
38ccf7e (HEAD -> master) HEAD@{0}: reset: moving to 38ccf7
3f87468 HEAD@{1}: reset: moving to HEAD^
38ccf7e (HEAD -> master) HEAD@{2}: commit: modify code from the result is... to the sum is ...
3f87468 HEAD@{3}: commit: modify the add operation from z=5+y to z=x+y
2ae02ec HEAD@{4}: commit (initial): this program is a C program
        版本号
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
```

10. 工作区和暂存区

- 工作区（Working Directory）

工作区就是在电脑里能看到的，我们前面创建的 d:/project/tsms 就是工作区。

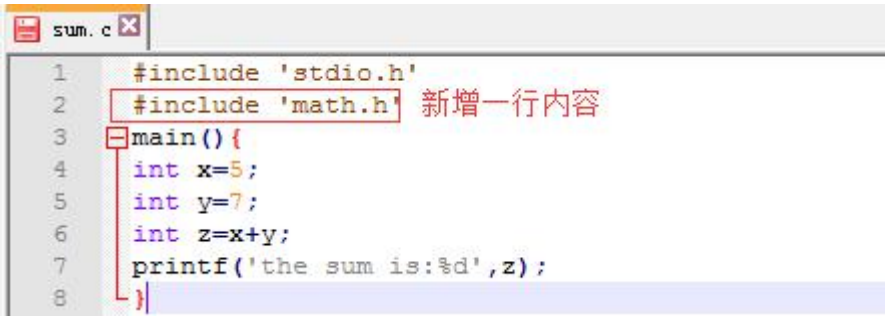
- 版本库（Repository）

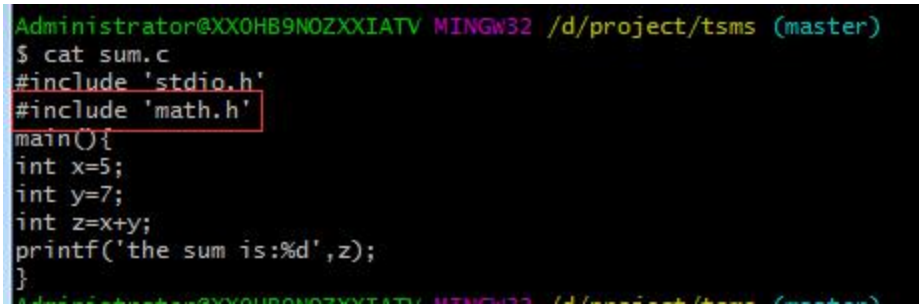
工作区有一个隐藏目录 .git，这个不算工作区，是 Git 的版本库。

- 暂存区

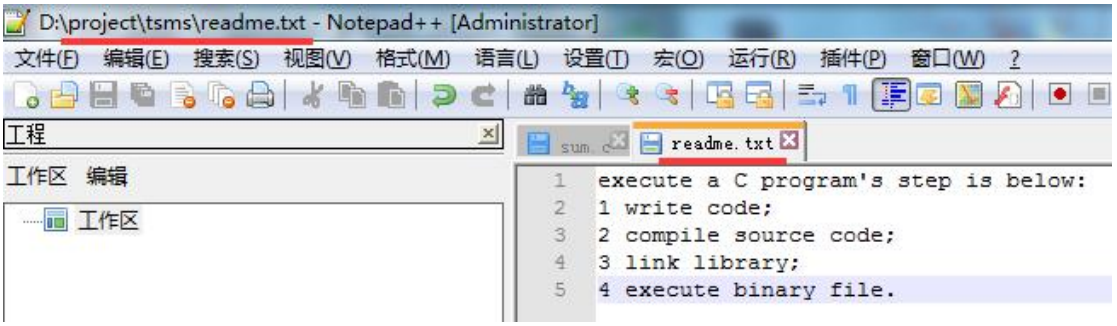
版本库中有一个称为 stage 的暂存区，还有 Git 自动创建的第一个分支 master，以及指向 master 的一个指针 HEAD。
执行 git add 命令添加文件，实际上就是将文件添加到暂存区。
执行 git commit 命令提交文件，实际上是把暂存区的所有内容提交到当前分支。
通过练习理解工作区、暂存区的功能和意义

1) **修改 sum.c 文件内容**





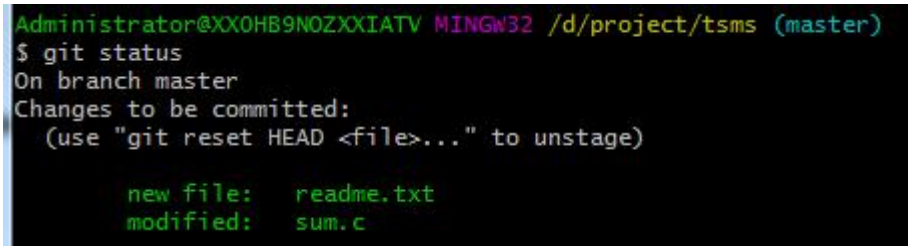
2) **在工作区新增一个 readme.txt 文件**



3) 查看仓库状态

4) 添加 sum.c 和 readme.txt 文件到仓库

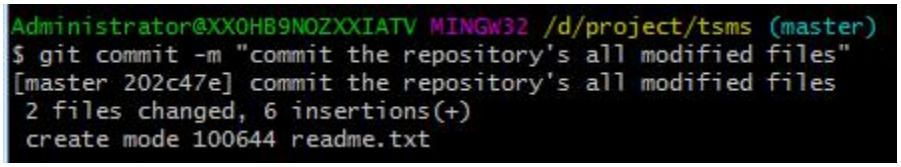
5) 查看仓库状态



执行 git add 命令后，sum.c 和 readme.txt 两个文件已进入暂存区，等待提交。

6) 提交修改

执行 git commit 命令一次性把暂存区中的所有修改提交到分支。



7) 查看仓库状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```

8) 查看历史修改记录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git log
commit 202c47e97219accd2458f84fb44403fd84f085ef (HEAD -> master)
Author: bxf <crazybai@163.com>
Date:   Wed May 16 23:53:12 2018 +0800

    commit the repository's all modified files

commit 38ccf7e436cabedf14afc3b12ea9ee1cf10689f8
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:54:01 2018 +0800

    modify code from the result is... to the sum is ...

commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
```

11. 撤销修改

情况一：修改未被添加到暂存区，需要撤销修改

1) 在 readme.txt 文件中新增一行

sum.c | readme.txt

```
1 execute a C program's step is below:
2 1 write code;
3 2 compile source code;
4 3 link library;
5 4 execute binary file.
6 5 this is php file operation step.
```

新增一行

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
5 this is php file operation step.
```

2) **查看仓库状态**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (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:   readme.txt  使用git checkout -- 文件名 命令丢弃工作区中的修改

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

Git 提示使用 git checkout --file 命令可以丢弃工作区中的修改。

3) **执行 git checkout 命令，将工作区的修改全部撤销**

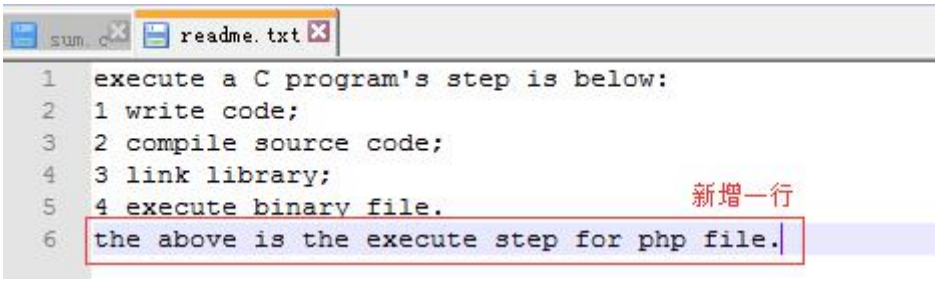
```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git checkout -- readme.txt
```

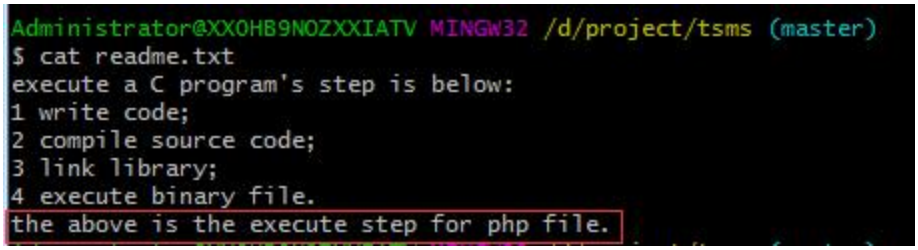
4) **再次查看当前状态**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
```

情况二：修改已经添加到暂存区，需要撤销

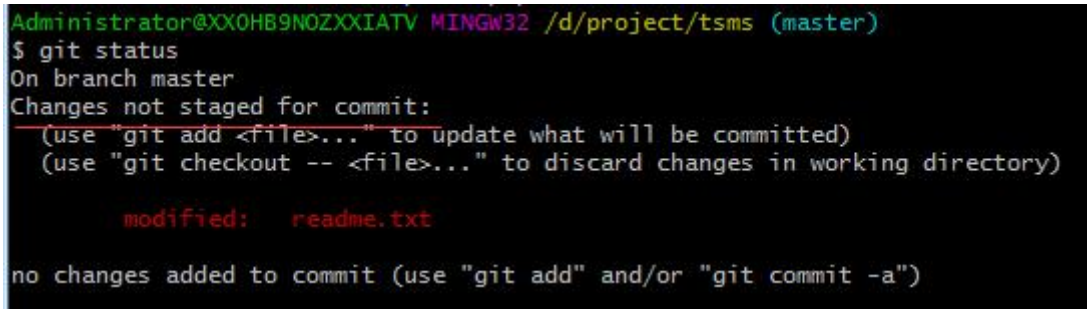
1) 在 readme.txt 文件中新增一行





2) 查看状态

修改未添加到暂存区



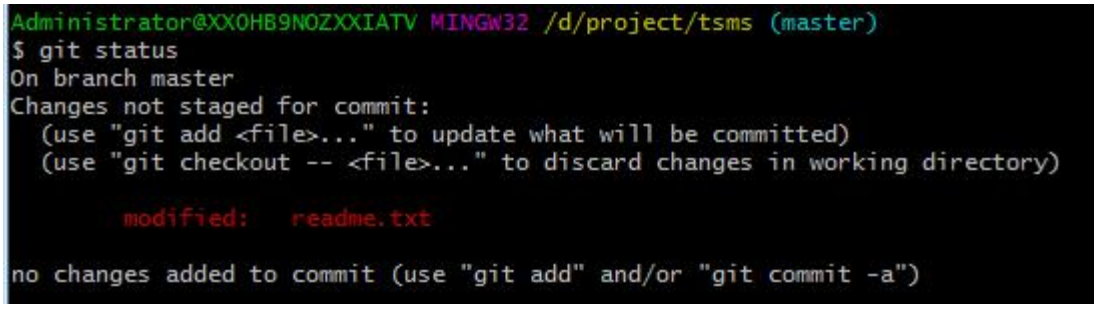
3) 将修改添加到暂存区

4) 查看状态

Git 提示，可以使用 git reset HEAD file 命令，将暂存区的修改撤销掉（将修改回退到工作区）。

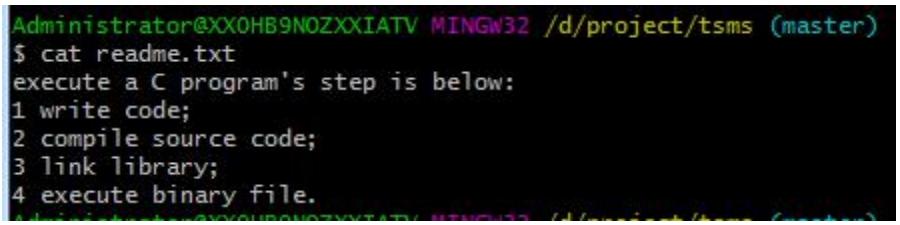
5) 执行 get reset 命令

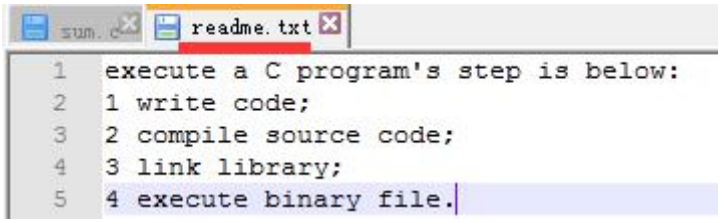
6) 查看状态



7) 执行 git checkout -- readme.txt 命令，丢弃工作区的修改

8) 查看文件内容





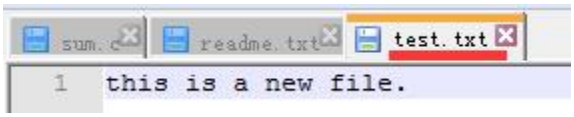
12. 删除文件

场景一：

手动删除工作中的 test.txt 文件，还未执行 git commit 命令，此时版本库中还存在 test.txt

文件，此时可以使用 git　checkout 命令将版本库中的 test.txt 文件恢复到工作区中。

1) **新增一个 test.txt 文件**



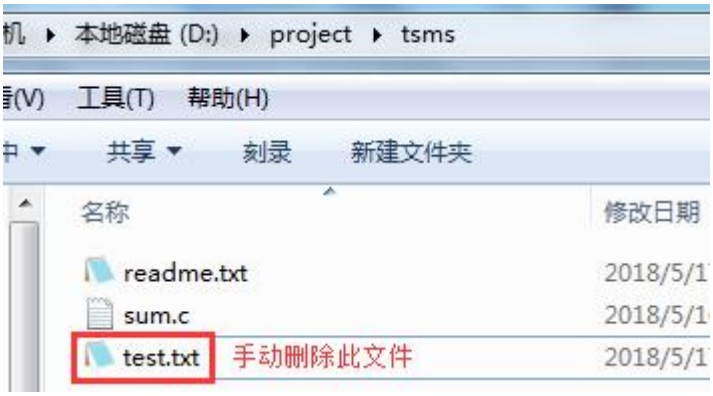
2) **查看状态**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
```

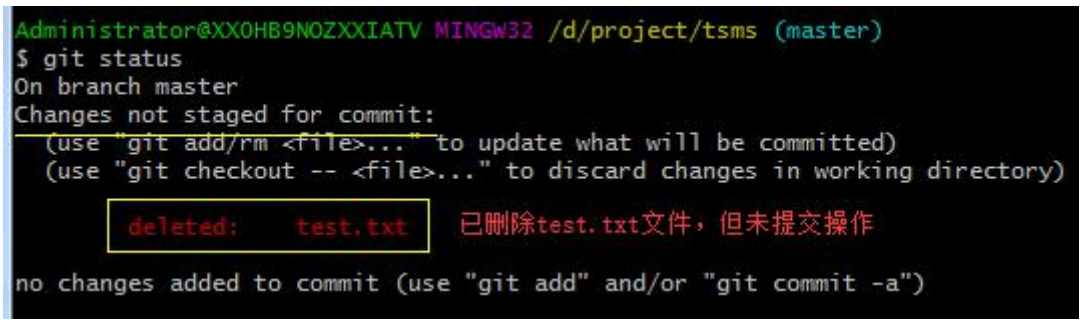
3) **添加并提交文件**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git add test.txt

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   test.txt


Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git commit -m 'add test.txt'
[master c2a8727] add test.txt
 1 file changed, 1 insertion(+)
 create mode 100644 test.txt

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```

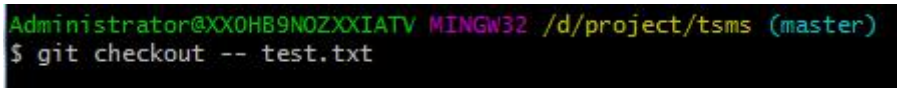
4) 在目录中手动删除 test.txt 文件



5) 查看状态

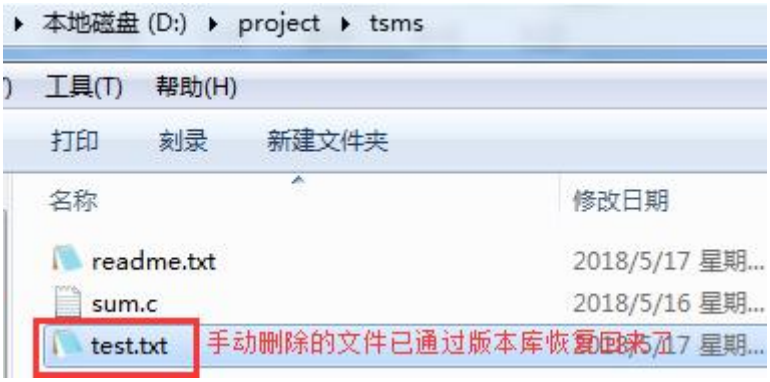


6) 使用版本库恢复删除的 test.txt 文件到工作区



7) **查看状态**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```



场景二：

手动删除工作区中的 test.txt 文件，此时版本库中还存在 test.txt 文件，版本库与工作区不一致，此时可执行 git　rm 命令将版本库中的 test.txt 文件删除。

1) **查看状态**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```

2) **删除工作区中的 test.txt 文件**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ ls
readme.txt  sum.c  test.txt

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ rm test.txt

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ ls
readme.txt  sum.c
```

3) 查看状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        deleted:    test.txt
```

4) 删除版本库中的 test.txt 文件

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git rm test.txt
rm 'test.txt'
```

5) 查看状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        deleted:    test.txt
```

6) 提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git commit -m 'delete test.txt from repository'
[master 9555a09] delete test.txt from repository
 1 file changed, 1 deletion(-)
 delete mode 100644 test.txt
```

7) 查看状态

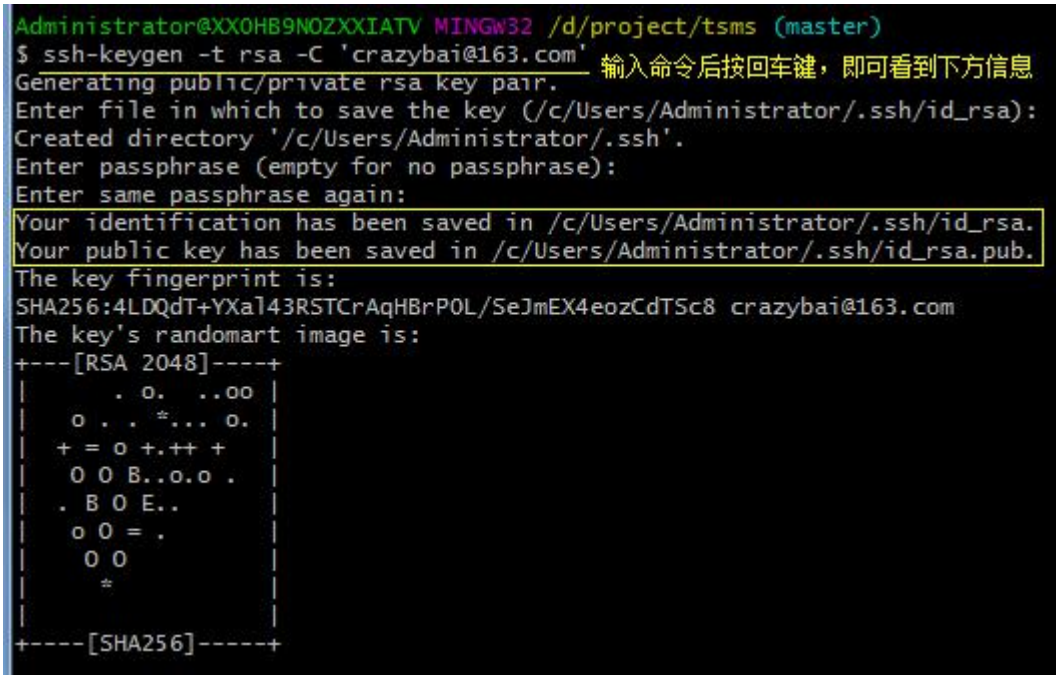
```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git status
On branch master
nothing to commit, working tree clean
```

8) 查看日志

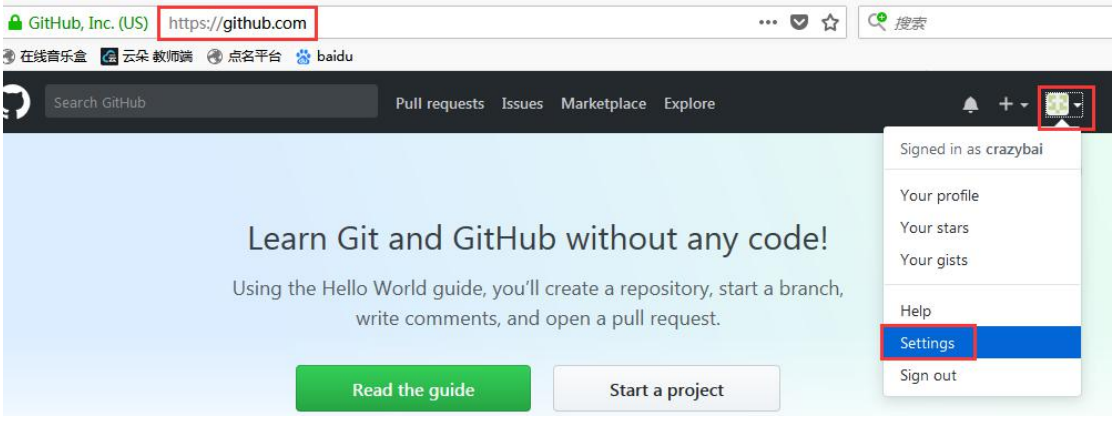
```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git log
commit 9555a09461f9f0b9a7520e2387c77712e0b9ecaa (HEAD -> master)
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:19:56 2018 +0800

    delete test.txt from repository

commit c2a87277d78ec15f9fed29e0496ba096d8e853d1
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:06:16 2018 +0800

    add test.txt

commit 202c47e97219accd2458f84fb44403fd84f085ef
Author: bxf <crazybai@163.com>
Date:   Wed May 16 23:53:12 2018 +0800

    commit the repository's all modified files

commit 38ccf7e436cabedf14afc3b12ea9ee1cf10689f8
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:54:01 2018 +0800

    modify code from the result is... to the sum is ...

commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program
```

13. 远程仓库

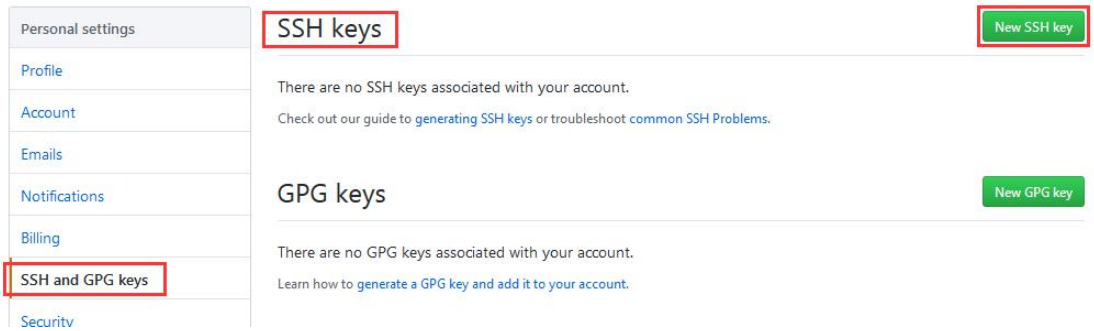
1) 创建 SSH Key



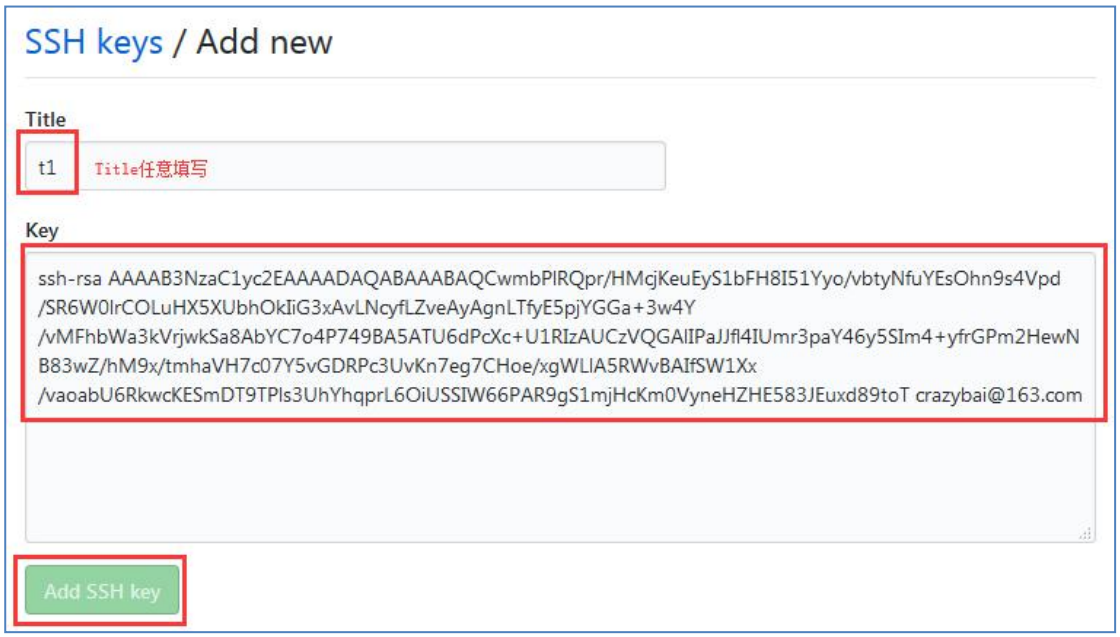
2) 登陆 GitHub，打开“ settings”页面



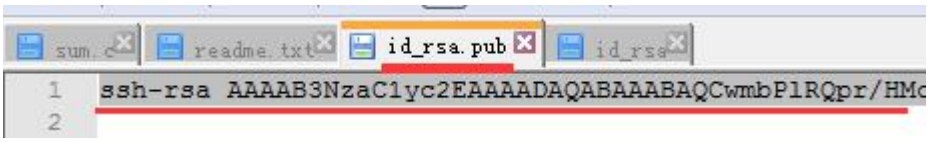
3) 选择“SSH and GPG Keys”项，单击右上角“New SSH Key”按钮



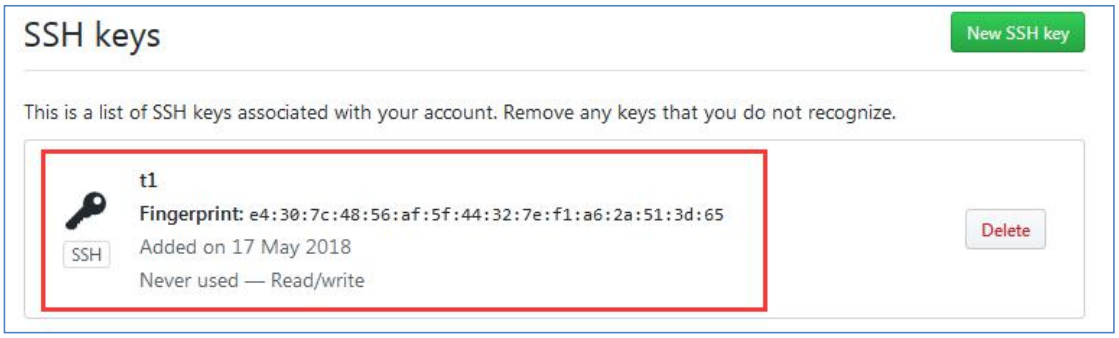
4) 输入 Titke 和 Key



复制 id_rsa.pub 文件中内容到上图中的 Key 文本域中

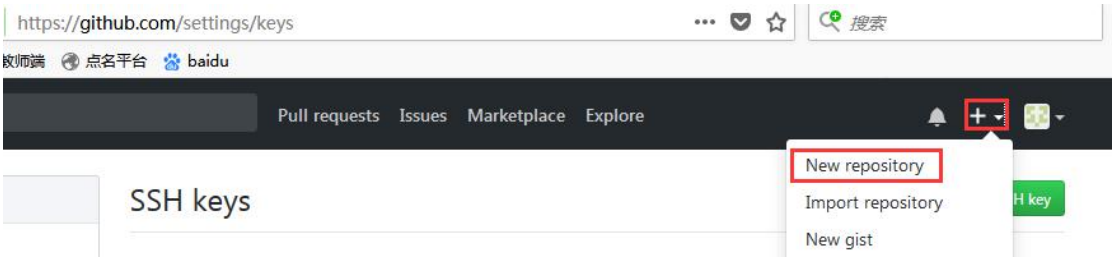


5) 单击“Add SSH Key”按钮后完成操作，效果如下

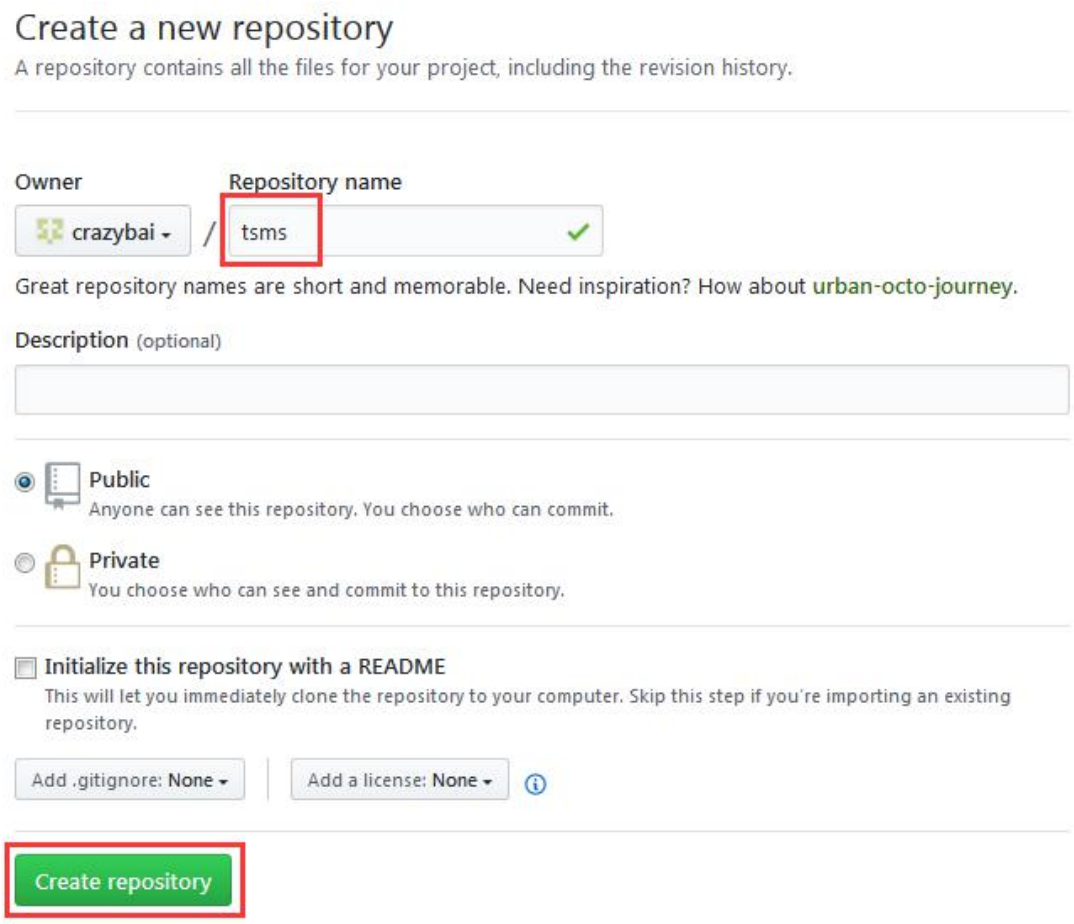


14. 添加远程库

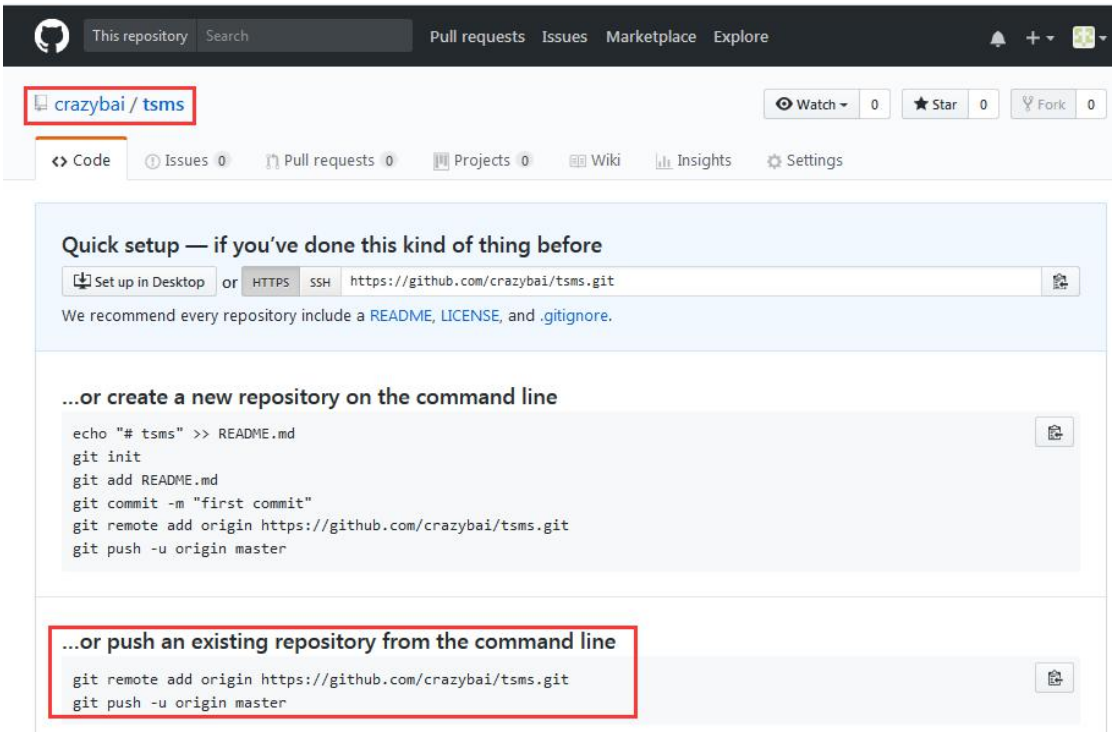
1) 在 GitHub 上创建新的仓库



2) 输入仓库名称


Create a new repository
A repository contains all the files for your project, including the revision history.
Owner
Repository name
crazybai
/
tsms
Great repository names are short and memorable. Need inspiration? How about urban-octo-journey.
Description (optional)
Public
Anyone can see this repository. You choose who can commit.
Private
You choose who can see and commit to this repository.
Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.
Add .gitignore: None
Add a license: None
Create repository

3) 仓库创建完成


This repository
Search
Pull requests
Issues
Marketplace
Explore
crazybai / tsms
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 0
Projects 0
Wiki
Insights
Settings
Quick setup — if you've done this kind of thing before
Set up in Desktop
or
HTTPS
SSH
https://github.com/crazybai/tsms.git
We recommend every repository include a README, LICENSE, and .gitignore.
...or create a new repository on the command line
echo "# tsms" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/crazybai/tsms.git
git push -u origin master
...or push an existing repository from the command line
git remote add origin https://github.com/crazybai/tsms.git
git push -u origin master

4) **本地关联远程库**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git remote add origin https://github.com/crazybai/tsms.git
```

5) **将本地仓库内容推送到 GitHub 上的远程仓库**

a) 要求输入 GitHub 站点的用户名

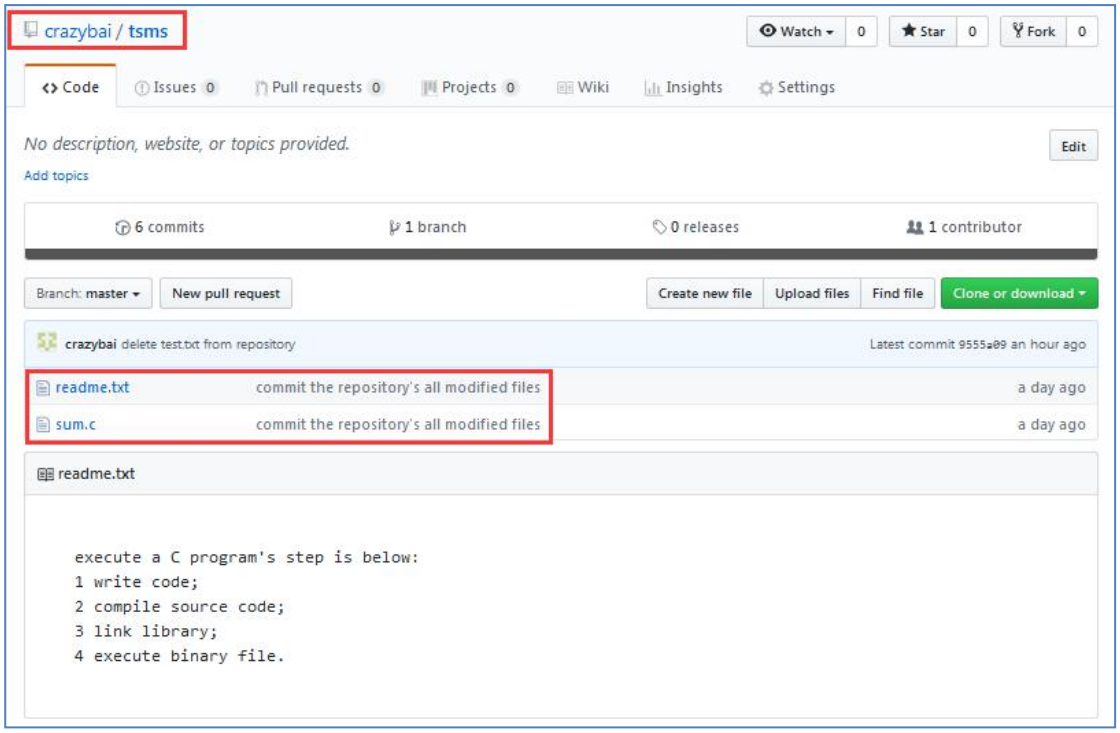
```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git push -u origin master
Username for 'https://github.com': crazybai@163.com   输入GitHub的用户名
```

b) 要求输入 GitHub 站点的用户密码

c) 推送成功

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git push -u origin master
Username for 'https://github.com': crazybai@163.com
Counting objects: 17, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (13/13), done.
Writing objects: 100% (17/17), 1.61 KiB | 205.00 KiB/s, done.
Total 17 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/crazybai/tsms.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

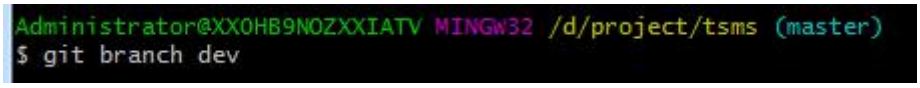
6) 查看远程仓库内容



15. 分支管理

1) 查看分支

2) 创建 dev 分支



3) 查看分支

4) **切换到 dev 分支**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git checkout dev
Switched to branch 'dev'
```

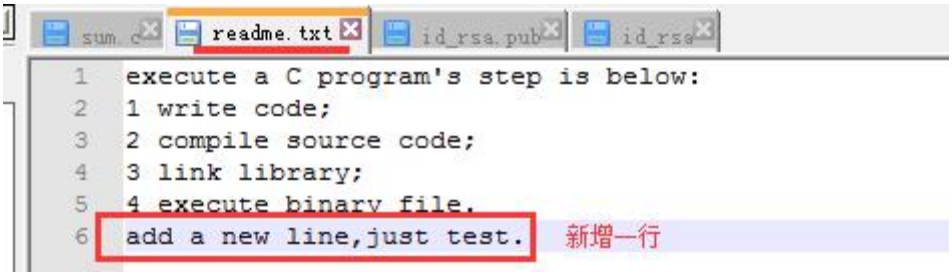
5) **查看分支**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git branch
* dev    当前分支切换为dev
  master
```

注意：可以使用“git checkout -b dev”实现创建 dev 分支，并切换到 dev 分支（即将当前分支设置为 dev）

6) **在 dev 分支上修改 readme.txt 文件并提交**

a) 修改 readme.txt 文件，新增一行内容



b) 查看状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git status
On branch dev
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:   readme.txt

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

c) 添加并提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git add readme.txt

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git status
On branch dev
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   readme.txt


Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git commit -m 'add a new line in readme.txt file'
[dev 5e25de1] add a new line in readme.txt file
 1 file changed, 2 insertions(+), 1 deletion(-)

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git status
On branch dev
nothing to commit, working tree clean

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$
```

d) 查看文件内容

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
add a new line,just test.
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$
```

e) 查看历史记录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git log
commit 5e25de1106954684ac4f139ec0b5a0a2bf1ba352 (HEAD -> dev)
Author: bxf <crazybai@163.com>
Date:   Fri May 18 00:42:07 2018 +0800

    add a new line in readme.txt file

commit 9555a09461f9f0b9a7520e2387c77712e0b9ecaa (origin/master, master)
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:19:56 2018 +0800

    delete test.txt from repository

commit c2a87277d78ec15f9fed29e0496ba096d8e853d1
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:06:16 2018 +0800

    add test.txt

commit 202c47e97219accd2458f84fb44403fd84f085ef
Author: bxf <crazybai@163.com>
Date:   Wed May 16 23:53:12 2018 +0800

    commit the repository's all modified files

commit 38ccf7e436cabedf14afc3b12ea9ee1cf10689f8
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:54:01 2018 +0800

    modify code from the result is... to the sum is ...

commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$
```

7) 切换回 master 分支

a) 切换分支

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (dev)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

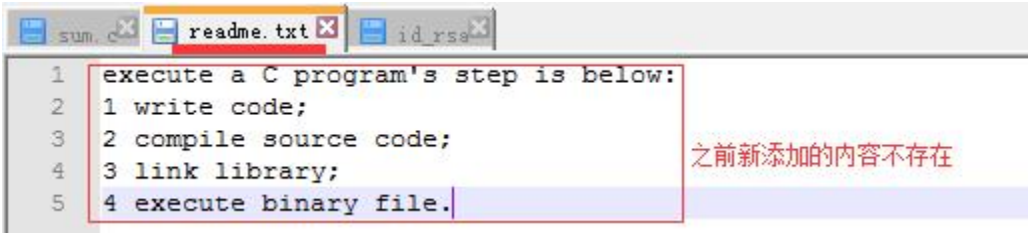
b) 查看分支

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git branch
  dev
* master
```

c) 查看文件内容

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
```

之前新添加的内容不存在



之前新添加的内容不存在

d) 查看历史记录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git log
commit 9555a09461f9f0b9a7520e2387c77712e0b9ecaa (HEAD -> master, origin/master)
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:19:56 2018 +0800

    delete test.txt from repository

commit c2a87277d78ec15f9fed29e0496ba096d8e853d1
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:06:16 2018 +0800

    add test.txt

commit 202c47e97219accd2458f84fb44403fd84f085ef
Author: bxf <crazybai@163.com>
Date:   Wed May 16 23:53:12 2018 +0800

    commit the repository's all modified files

commit 38ccf7e436cabedf14afc3b12ea9ee1cf10689f8
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:54:01 2018 +0800

    modify code from the result is... to the sum is ...

commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
```

8) 将 dev 分支合并到 master 分支上

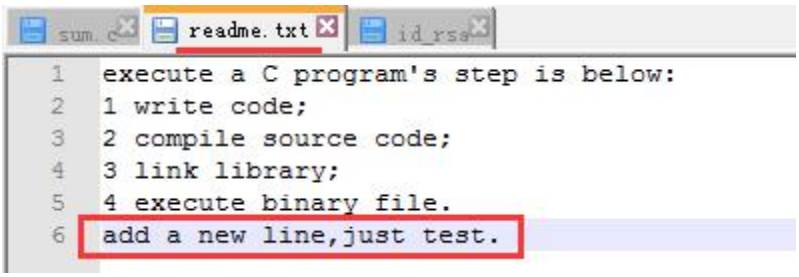
a) 合并分支

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git merge dev
Updating 9555a09..5e25de1
Fast-forward
 readme.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

b) 查看内容

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
add a new line,just test.
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

出现了之前的新增行，和dev分支的最新提交完全相同



c) 查看历史记录

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git log
commit 5e25de1106954684ac4f139ec0b5a0a2bf1ba352 (HEAD -> master, dev)
Author: bxf <crazybai@163.com>
Date:   Fri May 18 00:42:07 2018 +0800

    add a new line in readme.txt file

commit 9555a09461f9f0b9a7520e2387c77712e0b9ecaa (origin/master)
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:19:56 2018 +0800

    delete test.txt from repository

commit c2a87277d78ec15f9fed29e0496ba096d8e853d1
Author: bxf <crazybai@163.com>
Date:   Thu May 17 23:06:16 2018 +0800

    add test.txt

commit 202c47e97219accd2458f84fb44403fd84f085ef
Author: bxf <crazybai@163.com>
Date:   Wed May 16 23:53:12 2018 +0800

    commit the repository's all modified files

commit 38ccf7e436cabedf14afc3b12ea9ee1cf10689f8
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:54:01 2018 +0800

    modify code from the result is... to the sum is ...

commit 3f87468ca46e0d1f91b565e87e967bad6cf84e38
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:26:01 2018 +0800

    modify the add operation from z=5+y to z=x+y

commit 2ae02ec5d018c2425785561e88934d8bfc24dc15
Author: bxf <crazybai@163.com>
Date:   Wed May 16 22:05:03 2018 +0800

    this program is a C program

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

9) **删除分支**

a) 将 dev 分支合并到 master 分支后，就可以删除 dev 分支了

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git branch -d dev
Deleted branch dev (was 5e25de1).
```

b) 查看分支

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git branch
* master    dev分支已被删除，只剩下master分支了
```

16. 解决冲突

1) 创建并切换到新分支 feature1

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git checkout -b feature1
Switched to a new branch 'feature1'
```

2) 修改 readme.txt 文件内容

sum.c | readme.txt

```
1  execute a C program's step is below:
2  1 write code;
3  2 compile source code;
4  3 link library;
5  4 execute binary file.
6  add a new line,just test,yes.   新增 yes
```

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (feature1)
$ git status
On branch feature1
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:   readme.txt

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

3) 添加并提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (feature1)
$ git add readme.txt
```

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (feature1)
$ git status
On branch feature1
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   readme.txt
```

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (feature1)
$ git commit -m 'in the last line add yes'
[feature1 bddb813] in the last line add yes
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (feature1)
$ git status
On branch feature1
nothing to commit, working tree clean
```

4) **切换到** master **分支**

切换

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (feature1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$
```

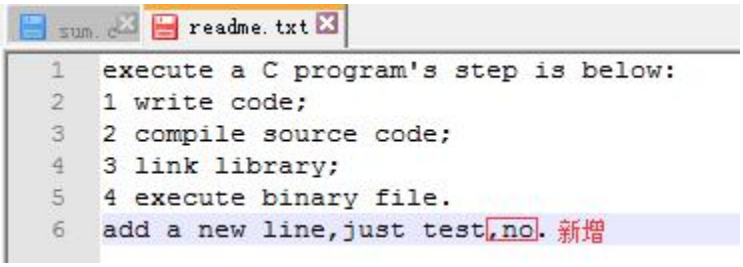
查看文件内容

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
add a new line,just test.
```

sum.c readme.txt

```
1  execute a C program's step is below:
2  1 write code;
3  2 compile source code;
4  3 link library;
5  4 execute binary file.
6  add a new line,just test.
```

5) 修改 readme.txt 文件内容



6) 添加并提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git add readme.txt
```

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git commit -m 'this is master branch,in the last line add a no'
[master af70220] this is master branch,in the last line add a no
 1 file changed, 1 insertion(+), 1 deletion(-)
```

7) 合并 feature1 分支到 master 分支

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git merge feature1
Auto-merging readme.txt
CONFLICT (content): Merge conflict in readme.txt
Automatic merge failed; fix conflicts and then commit the result.
```

提示合并分支时发生冲突，需要解决冲突后再提交

8) 查看状态

提示有冲突存在，需要处理

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master|MERGING)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   readme.txt

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

9) **查看文件内容**

Git 使用“=======”符号标记出不同分支的内容

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master|MERGING)
$ cat readme.txt
execute a C program's step is below:
1 write code;
2 compile source code;
3 link library;
4 execute binary file.
<<<<<<< HEAD
add a new line,just test,no.
=======
add a new line,just test,yes.
>>>>>>> feature1
```

sum.c readme.txt

```
1  execute a C program's step is below:
2  1 write code;
3  2 compile source code;
4  3 link library;
5  4 execute binary file.
6  <<<<<<< HEAD
7  add a new line,just test,no.
8  =======
9  add a new line,just test,yes.
10 >>>>>>> feature1
```

10) **解决冲突**

a) 手动修改 readme.txt 文件

sum.c readme.txt

```
1  execute a C program's step is below:
2  1 write code;
3  2 compile source code;
4  3 link library;
5  4 execute binary file.
6  add a new line,just test,yes.
7
```

修改文件

b) 添加文件

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master|MERGING)
$ git add readme.txt
```

c) 查看状态

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master|MERGING)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)
                                        提示 所有冲突已处理，但还未提交
Changes to be committed:

        modified:   readme.txt
```

d) 提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master|MERGING)
$ git commit -m 'conflict fixed'
[master 8b53393] conflict fixed
```

17. 多人协作

1) 查看远程仓库信息

简要信息

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git remote
origin
```

详细信息

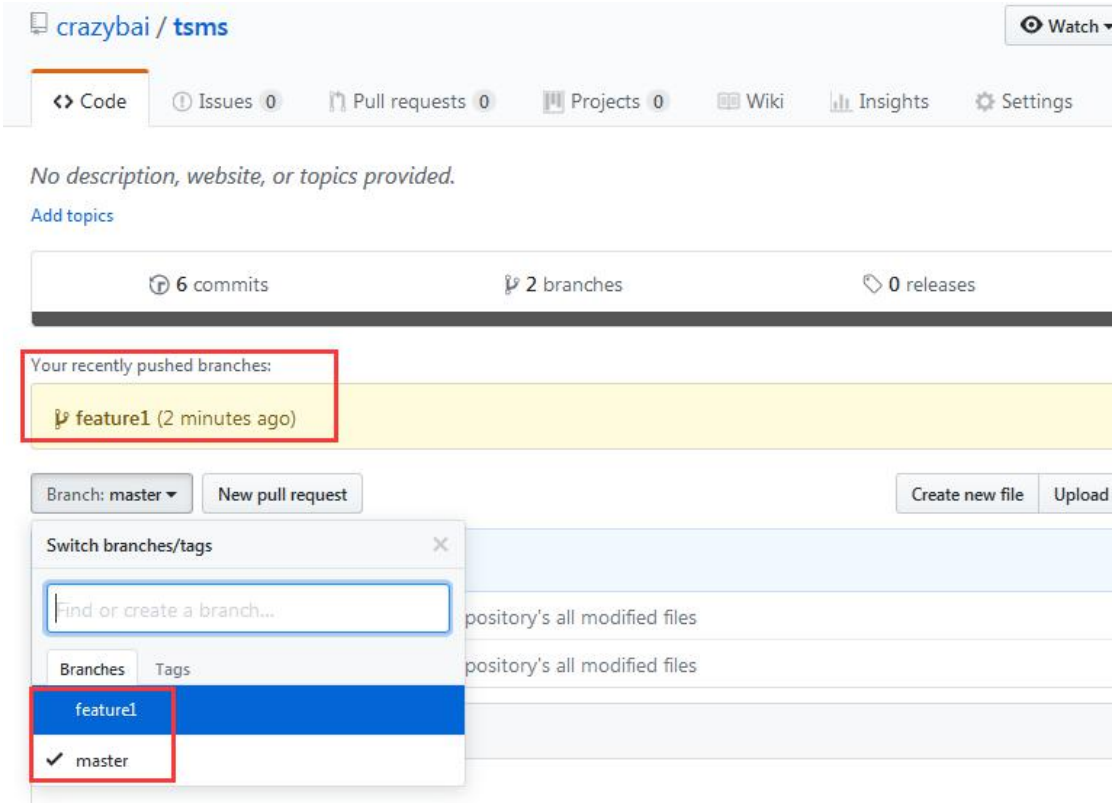
```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git remote -v
origin  https://github.com/crazybai/tsms.git (fetch)
origin  https://github.com/crazybai/tsms.git (push)
```

2) 向远程仓库推送分支

推送

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/tsms (master)
$ git push origin feature1
Username for 'https://github.com': crazybai@163.com
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 751 bytes | 187.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/crazybai/tsms.git
 * [new branch]      feature1 -> feature1
```

在 GitHub 上查看分支



3) **抓取分支**

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/one
$ git clone git@github.com:crazybai/tsms.git
Cloning into 'tsms'...
remote: Counting objects: 24, done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 24 (delta 4), reused 24 (delta 4), pack-reused 0
Receiving objects: 100% (24/24), done.
Resolving deltas: 100% (4/4), done.
```

4) **创建选择 orign 的 feature1 分支到本地**

Clone 远程库后，默认情况下，只能看到本地的 master 分支

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/one/tsms (master)
$ git branch
* master
```

创建远程 origin 的 feature1 分支到本地

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/one/tsms (master)
$ git checkout -b origin/feature1
Switched to a new branch 'origin/feature1'
```

本地 feature1 分支创建成功

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/one/tsms (origin/feature1)
$ git branch
  master
* origin/feature1
```

5) 修改 feature1 分支上的内容

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/one/tsms (origin/feature1)
$ vim sum.c
```

MINGW32:/d/project/one/tsms

```
#include 'stdio.h'
#include 'math.h'
main(){
int x=5;
int y=7;
int z=x+y;          将sum修改为result
printf('the result is:%d',z);
}
```

sum.c | readme.txt

```
1  #include 'stdio.h'
2  #include 'math.h'
3  main(){
4  int x=5;
5  int y=7;
6  int z=x+y;          将原来的sum修改为result
7  printf('the result is:%d',z);
8  }
```

6) 将 dev 分支提交、推送到远程库

提交

```
Administrator@XXOHB9NOZXXIATV MINGW32 /d/project/one/tsms (origin/feature1)
$ git commit -m 'modify the sum.c file'
On branch origin/feature1
Changes not staged for commit:
        modified:   sum.c
```

推送

Git push

拉下来

Git pull

项目 git 操作流程:

在自己的分支上

git add . // 添加文件到暂存区

git commit -m'备注' //暂存区文件提交到自己的分支

git checkout master //切换到 master 分支

git merge 自己的分支 //合并分支（有冲突解决冲突）

git pull //拉取

git push //推送

git checkout 自己的分支 //切换到自己的分支

git merge master //合并 master 分支内容