

哈尔滨工业大学计算机科学与技术学院  
2023 年秋季学期 《开源软件开发实践》

Lab 1: Git 实战

姓名	学号	联系方式
唐葳蕤	2021112846	19828247939

## 目 录

1	实验要求.....	1
2	安装 Git .....	1
2.1	本地机器上安装 git.....	1
2.2	申请 github 帐号 .....	1
3	Git 操作过程 .....	2
3.1	实验场景(1): 仓库创建与提交 .....	2
3.2	实验场景(2): 分支管理 .....	5
3.3	实验场景(3): 在线 Git 练习 .....	10
4	小结.....	14

## 1 实验要求

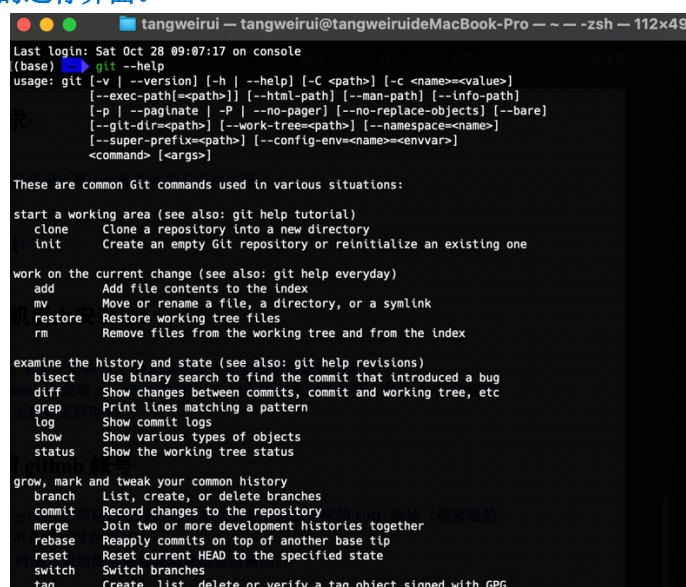
1. 熟练掌握 git 的基本指令和分支管理指令；
2. 掌握 git 支持软件配置管理的核心机理；
3. 在实践项目中使用 git/github 管理自己的项目源代码。

## 2 安装 Git

### 2.1 本地机器上安装 git

我在 MacOS 上安装的 Git，由于在之前我已经安装过 Git，因此只给出安装命令而没有截图：`brew install git`

下面是安装后的运行界面。



```
Last login: Sat Oct 28 09:07:17 on console
(base) ➜ git --help
usage: git [-v | --version] [-h | --help] [-C <path>] [-c <name>=<value>]
          [--exec-path<=path>] [--html-path] [--man-path] [--info-path]
          [-p | --paginate] [-P | --no-pager] [--no-replace-objects] [--bare]
          [--git-dir<path>] [--work-tree<path>] [--namespace<name>]
          [--super-prefix<path>] [--config-env=<name>=<envvar>]
          <command> [<args>]

These are common Git commands used in various situations:


start a working area (see also: git help tutorial)
 clone      Clone a repository into a new directory
 init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
 add        Add file contents to the index
 mv         Move or rename a file, a directory, or a symlink
 restore    Restore working tree files
 rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
 bisect     Use binary search to find the commit that introduced a bug
 diff       Show changes between commits, commit and working tree, etc
 grep       Print lines matching a pattern
 log        Show commit logs
 show       Show various types of objects
 status     Show the working tree status

grow, mark and tweak your common history
 branch     List, create, or delete branches
 commit     Record changes to the repository
 merge      Join two or more development histories together
 rebase     Reapply commits on top of another base tip
 reset      Reset current HEAD to the specified state
 switch     Switch branches
 tag        Create, list, delete or verify a tag object signed with GPG
```

### 2.2 申请 github 帐号

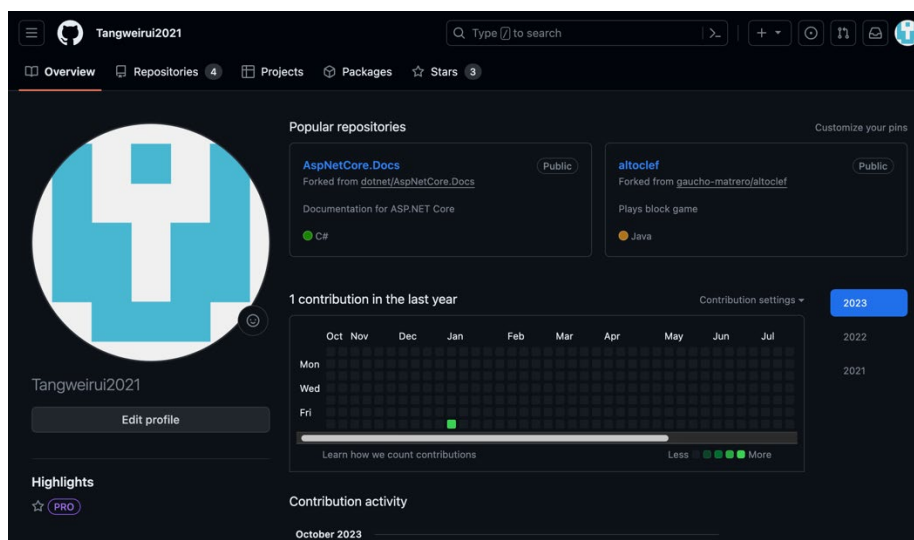
给出自己在 github 上申请的帐号名称和本次实验中创建仓库的 URL 地址(做实验的仓库地址，不是提交报告的地址)。

给出 github 网站上你的账号信息和项目信息的截图。

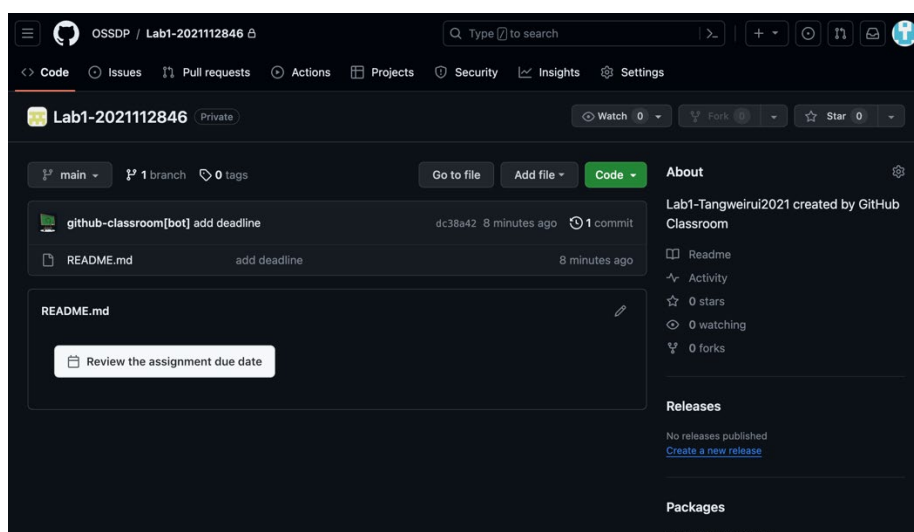
账号名: Tangweirui2021

仓库 URL: <https://github.com/OSSDP/Lab1-2021112846>

账号信息截图:



项目信息截图：



### 3 Git 操作过程

#### 3.1 实验场景(1)：仓库创建与提交

R0:在进行每次 git 操作之前，随时查看工作区、暂存区、git 仓库的状态，确认项目里的各文件当前处于什么状态；

```
(base) ~/Seafile/Work/开源软件开发实践 ➤ cd lab1
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ ls
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ ls -a
.
..
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤
```

R1:本地初始化一个 git 仓库，将自己开发的某个软件的源代码文件 加入进去，纳入 git 管理

```
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ git init
Initialized empty Git repository in /Users/tangweirui/Seafile/Work/开源软件开发实践/lab1/.git/
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤ ls -la
.
..
GitHubInfoUpdate
GitHubInfoUpdate.sln
LICENSE.txt
README.md
.git
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤ git add .
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   GitHubInfoUpdate.sln
    new file:   GitHubInfoUpdate/Functions.cs
    new file:   GitHubInfoUpdate/GitHubInfoUpdate.csproj
    new file:   GitHubInfoUpdate/MainForm.Designer.cs
    new file:   GitHubInfoUpdate/MainForm.cs
    new file:   GitHubInfoUpdate/MainForm.resx
    new file:   GitHubInfoUpdate/Objects/Assets.cs
    new file:   GitHubInfoUpdate/Objects/Config.cs
    new file:   GitHubInfoUpdate/Objects/Release.cs
    new file:   GitHubInfoUpdate/Objects/Repositories.cs
    new file:   GitHubInfoUpdate/Program.cs
    new file:   LICENSE.txt
    new file:   README.md
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤
```

## R2:提交

```
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤ git commit
[main (root-commit) 4a09ced] c1
13 files changed, 1108 insertions(+)
 create mode 100755 GitHubInfoUpdate.sln
 create mode 100755 GitHubInfoUpdate/Functions.cs
 create mode 100755 GitHubInfoUpdate/GitHubInfoUpdate.csproj
 create mode 100755 GitHubInfoUpdate/MainForm.Designer.cs
 create mode 100755 GitHubInfoUpdate/MainForm.cs
 create mode 100755 GitHubInfoUpdate/MainForm.resx
 create mode 100755 GitHubInfoUpdate/Objects/Assets.cs
 create mode 100755 GitHubInfoUpdate/Objects/Config.cs
 create mode 100755 GitHubInfoUpdate/Objects/Release.cs
 create mode 100755 GitHubInfoUpdate/Objects/Repositories.cs
 create mode 100755 GitHubInfoUpdate/Program.cs
 create mode 100755 LICENSE.txt
 create mode 100755 README.md
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤ git status
On branch main
nothing to commit, working tree clean
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤
```

## R3:手工对 3 个文件进行修改，然后查看有哪些文件被修改、具体的修改内容是什么

```
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ➤ git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   GitHubInfoUpdate/MainForm.cs
    modified:   GitHubInfoUpdate/Program.cs
    modified:   README.md

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

no changes added to commit (use "git add" and/or "git commit -a")
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ± ➤ git diff README.md
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ± ➤ git diff GitHubInfoUpdate/Program.cs
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ± ➤ git diff GitHubInfoUpdate/MainForm.cs
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ main ± ➤
```

```
lab1 — git diff README.md
diff --git a/README.md b/README.md
index 66280ac..45e50fe 100755
--- a/README.md
+++ b/README.md
@@ -1,3 @@
-# GitHubInfoUpdate
\ No newline at end of file
+# GitHubInfoUpdate
+
+已被修改
\ No newline at end of file
(END)
```

```
lab1 — git diff GitHubInfoUpdate/Program.cs — git — less < git diff Gi
diff --git a/GitHubInfoUpdate/Program.cs b/GitHubInfoUpdate/Program.cs
index 169966d..0dd1bbd 100755
--- a/GitHubInfoUpdate/Program.cs
+++ b/GitHubInfoUpdate/Program.cs
@@ -12,6 +12,7 @@ namespace GitHubInfoUpdate
    // see https://aka.ms/applicationconfiguration.
    ApplicationConfiguration.Initialize();
    Application.Run(new MainForm());
+    // 已被修改
+
    }
}
\ No newline at end of file
(END)
```

```
lab1 — git diff GitHubInfoUpdate/MainForm.cs — git — less < git diff GitHubInfoUpdate/MainF...
diff --git a/GitHubInfoUpdate/MainForm.cs b/GitHubInfoUpdate/MainForm.cs
index 92aefd2..1661401 100755
--- a/GitHubInfoUpdate/MainForm.cs
+++ b/GitHubInfoUpdate/MainForm.cs
@@ -98,7 +98,6 @@ namespace GitHubInfoUpdate
    (var repo, var owner, var date, var tag, var silence) = config.ReleaseInfo
[index].ToTuple();

    var tmp = await func.GetRelease($"https://api.github.com/repos/{owner}/{re
po}/releases");
-
    //monitors.Add(tmp);
    tmpMonitors[index] = tmp;
    if (tmp?.Count > 0)
    {
    }
}
\ No newline at end of file
(END)
```

#### R4:重新提交

```
((base) ~/Seafile/Work/开源软件开发实践/lab1 main git commit -a
[main d80f340] c2
3 files changed, 4 insertions(+), 2 deletions(-)
(base) ~/Seafile/Work/开源软件开发实践/lab1 main
```

#### R5:再对三个文件进行修改，重新提交

```
((base) ~/Seafile/Work/开源软件开发实践/lab1 main git status
On branch main
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   GitHubInfoUpdate/Functions.cs
    modified:   GitHubInfoUpdate/Objects/Release.cs
    modified:   GitHubInfoUpdate/Objects/Repositories.cs

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

no changes added to commit (use "git add" and/or "git commit -a")
((base) ~/Seafile/Work/开源软件开发实践/lab1 main git commit -a
[main 5b1ea0e] c3
3 files changed, 2 insertions(+), 2 deletions(-)
(base) ~/Seafile/Work/开源软件开发实践/lab1 main
```



## R6:把最后一次提交撤销

```

(base) ~/Seafile/Work/开源软件开发实践/lab1 main git revert HEAD
[main adb7a28] Revert "c3"
3 files changed, 2 insertions(+), 2 deletions(-)
(base) ~/Seafile/Work/开源软件开发实践/lab1 main git log
(base) ~/Seafile/Work/开源软件开发实践/lab1 main

```

## R7:查询项目的全部提交记录

```

lab1 - git log - git - less < git log - 95x45
commit adb7a28cd201d5cb94d9d06bbde07938845ac676 (HEAD -> main)
Author: Daniel <2116528714@qq.com>
Date: Sat Oct 28 10:25:36 2023 +0800

    Revert "c3"

    This reverts commit 5b1ea0ef6434abf16d264bd7744576a1018977d1.

commit 5b1ea0ef6434abf16d264bd7744576a1018977d1
Author: Daniel <2116528714@qq.com>
Date: Sat Oct 28 10:22:22 2023 +0800

    c3

commit d80f34005aaf099a4f284bbb343601ea76032211
Author: Daniel <2116528714@qq.com>
Date: Sat Oct 28 10:20:23 2023 +0800

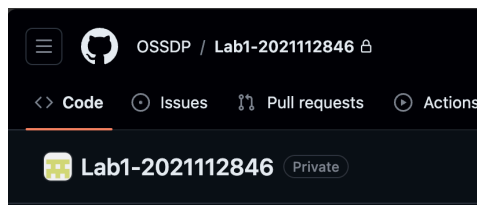
    c2

commit 4a09ced61ec0c6711ab1e091f7e397b283f87f97
Author: Daniel <2116528714@qq.com>
Date: Sat Oct 28 09:36:40 2023 +0800

    c1
(END)

```

## R8:在 GitHub 上创建名为 Lab1 的仓库，并在本地仓库建立相应的远程仓库



## R9:将之前各步骤得到的本地仓库全部内容推送到 GitHub 的 Lab1 仓库

```

(base) x ~/Seafile/Work/开源软件开发实践/lab1 main git push -f -u origin main
Enumerating objects: 31, done.
Counting objects: 100% (31/31), done.
Delta compression using up to 8 threads
Compressing objects: 100% (29/29), done.
Writing objects: 100% (31/31), 10.31 KiB | 5.15 MiB/s, done.
Total 31 (delta 13), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (13/13), done.
To https://github.com/OSSDP/Lab1-2021112846.git
+ dc38a42...adb7a28 main -> main (forced update)
branch 'main' set up to track 'origin/main'.
(base) ~/Seafile/Work/开源软件开发实践/lab1 main

```

## 3.2 实验场景(2): 分支管理

## R1:获得本地 Lab1 仓库的全部分支，切换至分支 master;

```

(base) ~/Seafile/Work/开源软件开发实践/lab1 main git branch
(base) ~/Seafile/Work/开源软件开发实践/lab1 main
* main
(END)

(base) ~/Seafile/Work/开源软件开发实践/lab1 main git checkout -b master
Switched to a new branch 'master'
(base) ~/Seafile/Work/开源软件开发实践/lab1 master

```

**R2:在 master 基础上建立两个分支 B1、B2;**

```

(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ master ➤ git branch B1
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ master ➤ git branch B2
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ master ➤ git checkout B2
Switched to branch 'B2'
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ B2 ➤ git branch C4
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ B2 ➤

```

**R3:在 B2 分支基础上创建一个新分支 C4;**

```

(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ B2 ➤ git checkout C4
Switched to branch 'C4'
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ C4 ➤ git status
On branch C4
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   GitHubInfoUpdate/MainForm.cs
    modified:   GitHubInfoUpdate/Program.cs
    modified:   README.md

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

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

```

**R4:在 C4 上, 对 3 个文件进行修改并提交;**

```

(base) x ➤ ~/Seafile/Work/开源软件开发实践/lab1 ➤ C4 ± ➤ git commit -a
[C4 1a89b42] modify c4
3 files changed, 3 insertions(+), 2 deletions(-)

```

**R5:在 B1 分支上对同样的 3 个文件做不同修改并提交;**

```

(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ B1 ± ➤ git status
On branch B1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   GitHubInfoUpdate/MainForm.cs
    modified:   GitHubInfoUpdate/Program.cs
    modified:   README.md

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

no changes added to commit (use "git add" and/or "git commit -a")
(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ B1 ± ➤ git commit -a
[B1 11ec6cd] modify B1
3 files changed, 3 insertions(+), 2 deletions(-)

```

**R6:将 C4 合并到 B1 分支, 若有冲突, 手工消解;**

```

(base) ~/Seafile/Work/开源软件开发实践/lab1 ➤ B1 ➤ git merge C4
Auto-merging GitHubInfoUpdate/MainForm.cs
Auto-merging GitHubInfoUpdate/Program.cs
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

```



## 发生冲突

```

(base) x ~/SeafFile/Work/开源软件开发实践/lab1 B1 ±+ >M< git status
On branch B1
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

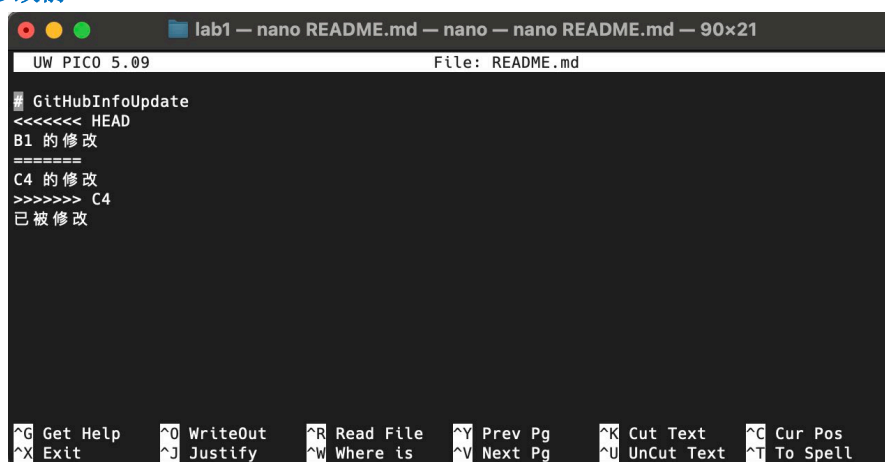
Changes to be committed:
  modified:   GitHubInfoUpdate/MainForm.cs
  modified:   GitHubInfoUpdate/Program.cs

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

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

```

## 修改前



```

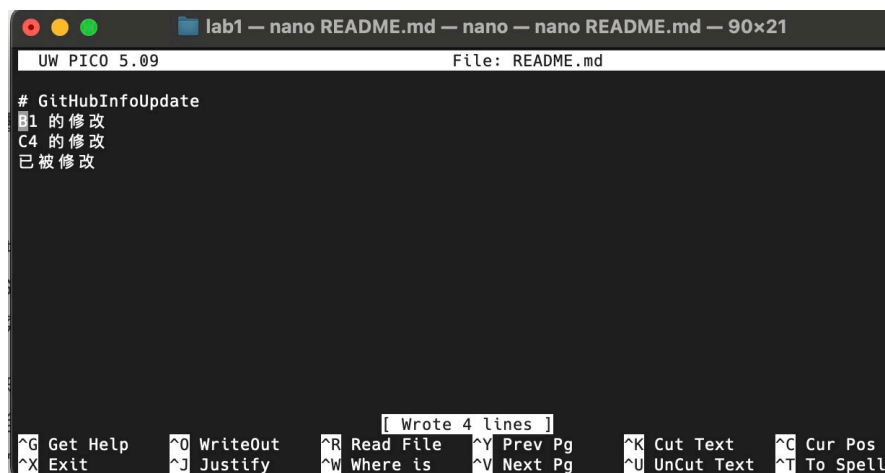
lab1 — nano README.md — nano — nano README.md — 90x21
UW PICO 5.09 File: README.md

# GitHubInfoUpdate
<<<<<< HEAD
B1 的修改
=====
C4 的修改
>>>>>> C4
已修改

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg   ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where is   ^V Next Pg   ^U UnCut Text ^T To Spell

```

## 修改后



```

lab1 — nano README.md — nano — nano README.md — 90x21
UW PICO 5.09 File: README.md

# GitHubInfoUpdate
B1 的修改
C4 的修改
已修改

[Wrote 4 lines]

^G Get Help  ^O WriteOut  ^R Read File  ^Y Prev Pg   ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where is   ^V Next Pg   ^U UnCut Text ^T To Spell

```

## 重新提交

```

(base) ~/SeafFile/Work/开源软件开发实践/lab1 B1 ±+ >M< git add README.md
(base) ~/SeafFile/Work/开源软件开发实践/lab1 B1 ±+ >M< git commit
[B1 6651676] Merge branch 'C4' into B1
(base) ~/SeafFile/Work/开源软件开发实践/lab1 B1

```

## R8:查看目前哪些分支已经合并、哪些分支尚未合并;

```

lab1 — tangweirui@tangweiruideMacBook-Pro — ..开发实践/lab1 —
((base) ~/Seafile/Work/开源软件开发实践/lab1) B1 git checkout B2
Switched to branch 'B2'
((base) ~/Seafile/Work/开源软件开发实践/lab1) B2 git status
On branch B2
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   GitHubInfoUpdate/MainForm.cs
    modified:   GitHubInfoUpdate/Program.cs
    modified:   README.md

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

no changes added to commit (use "git add" and/or "git commit -a")
((base) ~/Seafile/Work/开源软件开发实践/lab1) B2 git commit -a
[B2 b031ce5] modify B2
3 files changed, 3 insertions(+), 1 deletion(-)
((base) ~/Seafile/Work/开源软件开发实践/lab1) B2

```

## 当前分支情况

```

B1
* B2
C4
main
master
(END)

```

## 已合并的分支

```

* B2
main
master
(END)

```

## 未合并的分支

```

B1
C4
(END)

```

## R9:将已经合并的分支删除，将尚未合并的分支合并到一个新分支上，分支名字为学号;

```

((base) ~/Seafile/Work/开源软件开发实践/lab1) B2 git branch -d master
Deleted branch master (was adb7a28).
((base) ~/Seafile/Work/开源软件开发实践/lab1) B2 git branch -d main
Deleted branch main (was adb7a28).

((base) ~/Seafile/Work/开源软件开发实践/lab1) 2021112846 git branch -d B2
Deleted branch B2 (was b031ce5).

```

```
((base) ~/Seafile/Work/开源软件开发实践/lab1 2021112846 git merge B1
Auto-merging GitHubInfoUpdate/MainForm.cs
CONFLICT (content): Merge conflict in GitHubInfoUpdate/MainForm.cs
Auto-merging GitHubInfoUpdate/Program.cs
CONFLICT (content): Merge conflict in GitHubInfoUpdate/Program.cs
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
((base) x ~/Seafile/Work/开源软件开发实践/lab1 2021112846 ±+ >M< git commit -a
[2021112846 823f8d4] Merge branch 'B1' into 2021112846
((base) ~/Seafile/Work/开源软件开发实践/lab1 2021112846 git merge C4
Already up to date.
((base) ~/Seafile/Work/开源软件开发实践/lab1 2021112846
```

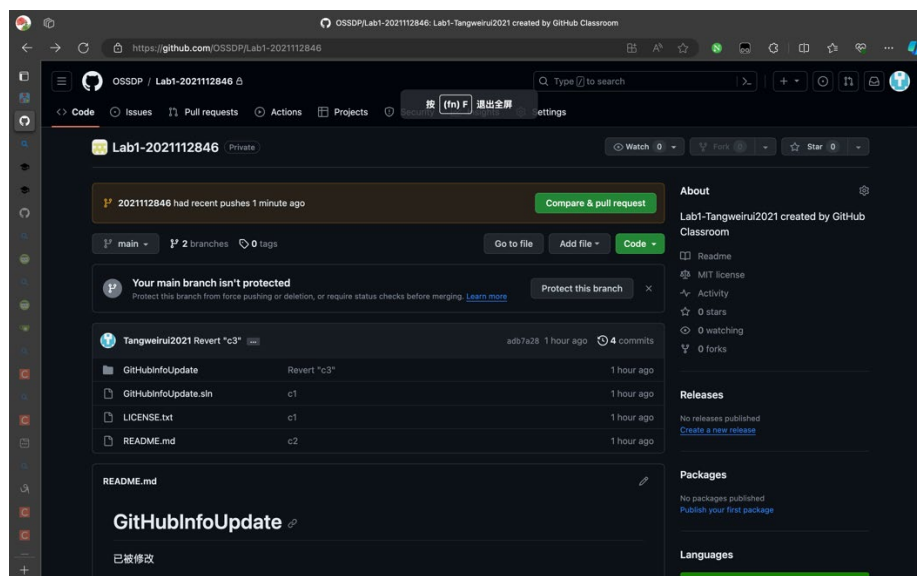
R10:将本地以学号命名的分支推送到 GitHub 上自己的仓库内;

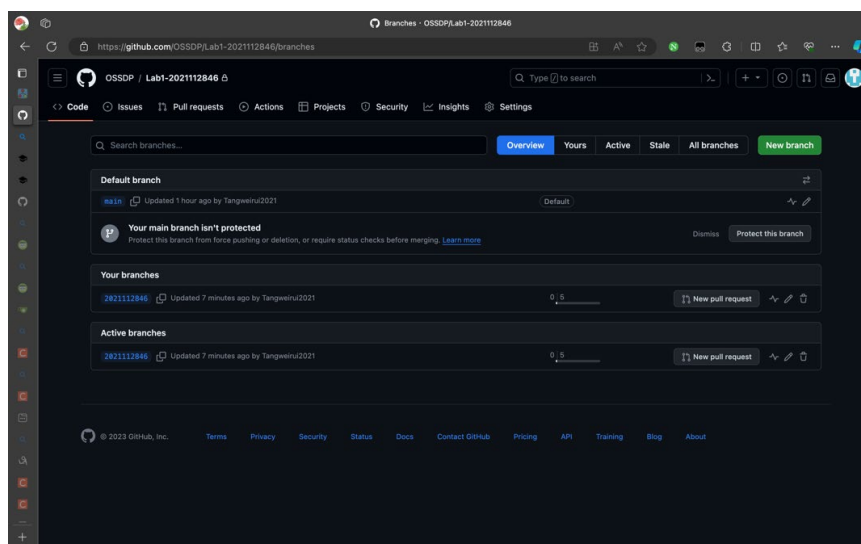
```
Already up to date.
((base) ~/Seafile/Work/开源软件开发实践/lab1 2021112846 git push origin 2021112846:2
021112846
Enumerating objects: 35, done.
Counting objects: 100% (35/35), done.
Delta compression using up to 8 threads
Compressing objects: 100% (27/27), done.
Writing objects: 100% (30/30), 2.17 KiB | 2.17 MiB/s, done.
Total 30 (delta 22), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (22/22), completed with 4 local objects.
remote:
remote: Create a pull request for '2021112846' on GitHub by visiting:
remote:   https://github.com/OSSDP/Lab1-2021112846/pull/new/2021112846
remote:
To https://github.com/OSSDP/Lab1-2021112846.git
 * [new branch]      2021112846 -> 2021112846
((base) ~/Seafile/Work/开源软件开发实践/lab1 2021112846
```

R11:查看完整的版本变迁树;

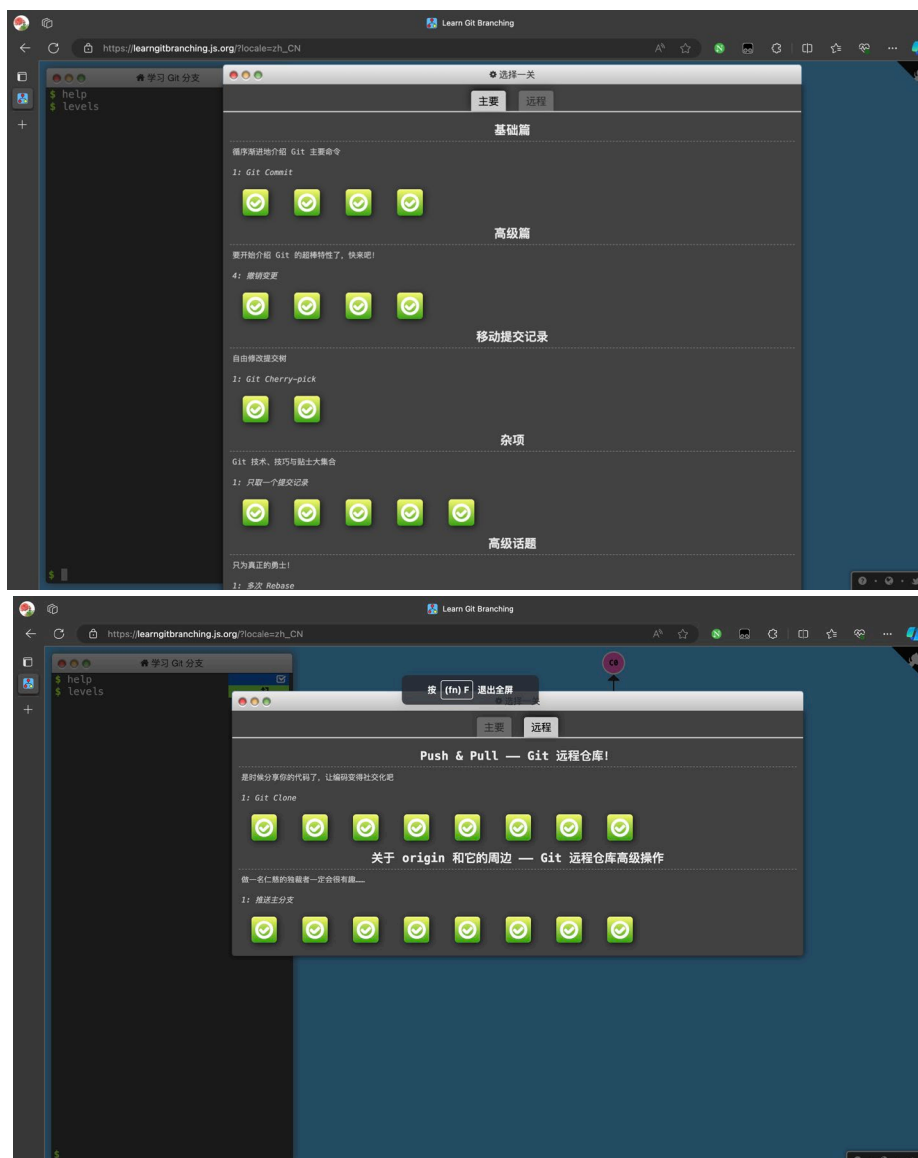
```
lab1 — git log --oneline --graph --all — git — less • git log --oneline --graph --all — 99x44
* 823f8d4 (HEAD -> 2021112846, origin/2021112846) Merge branch 'B1' into 2021112846
|
| * 6651676 (B1) Merge branch 'C4' into B1
| |
| | * 1a89b42 (C4) modify c4
| | * 11ec6cd modify B1
| |
| | * b031ce5 modify B2
| |
| | * adb7a28 (origin/main) Revert "c3"
| | * 5b1ea0e c3
| | * d80f340 c2
| | * 4a09ced c1
| |
| (END)
```

R12:在 Github 上以 web 页面的方式查看你的 Lab1 仓库的当前状态。





### 3.3 实验场景(3): 在线 Git 练习



**(一) 主要页面-基础篇****任务 1: Git Commit**`git commit``git commit`**任务 2: Git Branch**`git checkout -b bugFix`**任务 3: Git Merge**`git checkout -b bugFix``git commit``git checkout main``git commit``git merge bugFix`**任务 4: Git Rebase**`git checkout -b bugFix``git commit``git checkout main``git commit``git checkout bugFix``git rebase main`**(二) 主要页面-高级篇****任务 1: 分离 HEAD**`git checkout C4`**任务 2: 相对引用 (^)**`git checkout bugfix^`**任务 3: 相对引用 2 (~)**`git branch -f main C6``git checkout HEAD^``git branch -f bugFix HEAD^`**任务 4: 撤销变更**`git branch -f local C1``git checkout pushed``git revert HEAD`**(三) 主要页面-移动提交记录****任务 1: Git Cherry-pick**`git cherry-pick C3 C4 C7`**任务 2: 交互式 rebase**`git rebase -i HEAD~4`**(四) 主要页面-杂项****任务 1: 只取一个提交记录**`git rebase -i HEAD~3``git branch -f main HEAD`**任务 2: 提交的技巧 #1**`git rebase -i HEAD~2`



```
git commit --amend
git rebase -i HEAD~2
git branch -f main HEAD
```

**任务 3: 提交的技巧 #2**

```
git checkout main
git cherry-pick C2
git commit --amend
git cherry-pick C3
```

**任务 4: Git Tags**

```
git tag v0 C1
git tag v1 C2
git checkout C2
```

**任务 5: Git Describe**

```
git describe main
git commit
```

**(五) 远程页面-Git 远程仓库****任务 1: Git Clone**

```
git clone
```

**任务 2: 远程分支**

```
git commit
git checkout o/main
git commit
```

**任务 3: Git Fetch**

```
git fetch
```

**任务 4: Git Pull**

```
git pull
```

**任务 5: 模拟团队合作**

```
git clone
git fakeTeamwork 2
git commit
git pull
```

**任务 6: Git Push**

```
git commit
git commit
git push
```

**任务 7: 偏离的提交历史**

```
git clone
git fakeTeamwork
git commit
git pull --rebase
git push
```

**任务 8: 锁定的 Main(Locked Main)**

```
git checkout -b feature
git push
```

```
git branch -f main C1
```

#### (六) 远程页面-Git 远程仓库高级操作

##### 任务 1: 推送主分支

```
git checkout main
```

```
git pull
```

```
git rebase main side1
```

```
git rebase side1 side2
```

```
git rebase side2 side3
```

```
git branch -f main HEAD
```

```
git checkout main
```

```
git push
```

##### 任务 2: 合并远程仓库

```
git checkout main
```

```
git pull
```

```
git merge side1
```

```
git merge side2
```

```
git merge side3
```

```
git push
```

##### 任务 3: 远程追踪

```
git checkout -b side o/main
```

```
git commit
```

```
git pull --rebase
```

```
git push
```

##### 任务 4: Git push 的参数

```
git push origin main
```

```
git push origin foo
```

##### 任务 5: Git push 的参数 2

```
git push origin main^:foo
```

```
git push origin foo:main
```

##### 任务 6: Git fetch 的参数

```
git pull origin foo:main
```

```
git pull origin main^:foo
```

```
git branch -f foo HEAD
```

```
git checkout foo
```

##### 任务 7: 没有 source 的 source

```
git push origin :foo
```

```
git fetch origin :bar
```

##### 任务 8: Git pull 的参数

```
git pull origin bar:foo
```

```
git pull origin main:side
```

## 4 小结

经过本次实验，我更加系统地了解了 Git 的使用方法以及其强大的功能；同时还进行了实际实验，了解了在实际项目情况下如何使用 Git 进行版本管理，如何通过 Git 管理多人开发的程序。相信本次实验将为将来的学习和实践打下基础，在未来的实验中我将继续努力。但是，我认为本次实验存在一定的不足之处，例如没有提供网站习题的答案，不利于我们更好地掌握最优解法。