

厦门大学 RoboMaster 代码协作规范 V2.0

一、Git 服务器的选择

统一使用厦门大学源代码托管中心 <http://git.xmu.edu.cn>

全体成员完成注册，并加入对应组群(电控 RCS Electronic; 视觉 RCS Visual)



二、Git GUI 的选择

统一使用 TortoiseGit

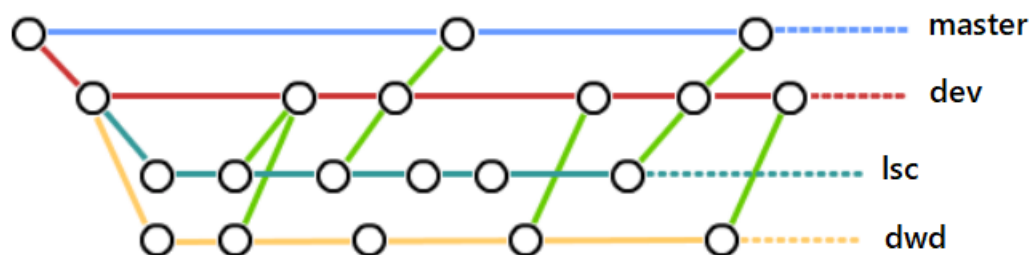
三、项目命名与分支管理

1、项目命名规则

统一以项目名的英文名命名且单词首字母大写，单词之间以下划线隔开，如步兵项目命名为 Infantry，能量机关项目命名为 Power_Rune。

2、分支管理

接轨通用规则，采用五大分支系统，即 master/dev/个人/feature/bug 分支，接下来对其分别进行介绍



(1) master 分支

主分支，要求保持其稳定性，只保存确认无误的重大修改（如：新功能开发完毕），不在此处进行工作

(2) dev(develop)分支

工作分支，用于保存和合并个人分支的修改

(3) 个人分支

个人分支，统一以开发者姓名拼音首字母命名(如罗上聪命名为 lsc)，

以便识别。个人修改在合并到 dev 分支前都应在此分支进行。

(4) feature 分支（可自选是否使用）

从属于个人分支，用于在不破坏个人分支中程序稳定性的情况下开发实验性代码。

(5) bug 分支

与个人分支独立，用于在不破坏个人分支的情况下快速修复程序已存在的错误。

关于 feature 和 bug 分支的具体操作参见以下链接，其余内容将在本文第五条介绍：

<https://www.liaoxuefeng.com/wiki/896043488029600>（Git 教程-廖雪峰）

章节：分支管理——分支管理策略、Bug 分支、Feature 分支、多人协作

四、代码与注释规范

1、代码编写规范

(1) 排版规范

①缩进

每个层次的缩进以一个 tab 键为标准，不要使用空格

②空格与空行

在代码段之间适当加一些空行，增强可读性。

二元运算符左右要加空格，如：s = a + b；而非 s=a+b.

③大括号

if 和 else 后跟的语句，即使只有一句，也应该加上大括号。

大括号单独占一行，如：

```
while(条件)
```

```
{
```

```
    //代码
```

```
}
```

若是空循环，一般写成 while(条件);

(2) 函数命名规范

1、函数名以大写字母开头，每个单词的首字母大写

如: SetSpeed, GetAngle 等.

2、to 一般缩写为 2, for 一般缩写为 4

如: Point2Point, Wait4StartSignal 等.

3、如果有多个模块,供外部使用的函数要在最前面加上模块名,并以下划线连接。

如: void Platform_SetLeftMotorSpeed(形参);(Platform 模块)

void Motion_Goto(形参);(Motion 模块)

(3) 变量命名规范

变量以小写字母开头,从第二个单词开始,单词首字母大写.

(如: speed, maxLength, motorSpeed 等.)

注意:

①除显而易见的缩写(如 num 等),尽量不要采用缩写.

②不要使用“魔数”

所谓的“魔数”指的是代码中突然出现的一个莫名的数字,

比如 $s = 3.14 * r * r$;这里的 3.14 就是一个魔数。虽然显然这里是 π ,但是一定要避免这种情况的出现。一般常量使用宏定义或者 const 修饰。并在旁边注释这个数的来源。

(4) 可读性与性能

①减少冗余计算

②删除无用代码

③不要牺牲程序的可读性

2、注释规范

(1) *.h 和 *.c 文件的头注释格式

//@filename: xxxx.c 或 xxxx.h (该文件的文件名)

//@date: 2020-10-4 (最后修改的日期)

//@author: XXX

//@brief: (说明文件内代码的功能, 注意事项等)

//@history: (每次修改文件时, 添加修改原因和修改时间)

(2) 函数注释格式

//@name: (函数名)

//@brief: (说明函数代码的功能, 注意事项等)

//@param: (函数的输入参数说明, 格式: 类型名 形参名 说明)

//@retval: (返回值, 格式: 类型名 返回变量名 说明)

//@note: (备注, 这项可以没有)

(3) 代码注释要求

1. 对于比较难懂的, 或关键算法应该注释。
2. 对于一些常用的全局变量应该注释。
3. 注释应有一个度, 适量即可。

(4) 注意事项

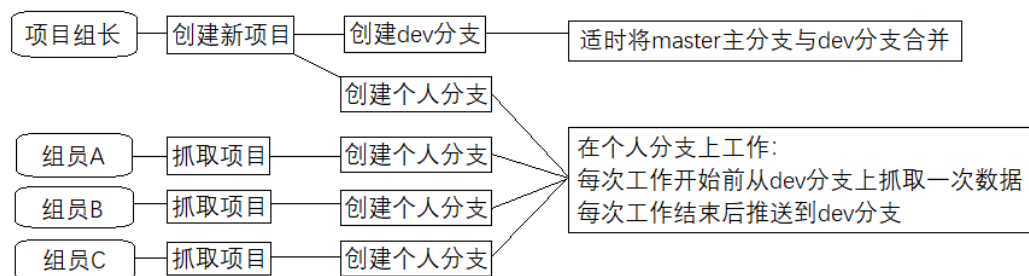
1. 文件注释和函数注释: 在编写完毕后添加
2. 代码注释: 在编写的过程中添加

3、多文件编程

* 参见附录【12 代码规范.pptx】第 13~15 页

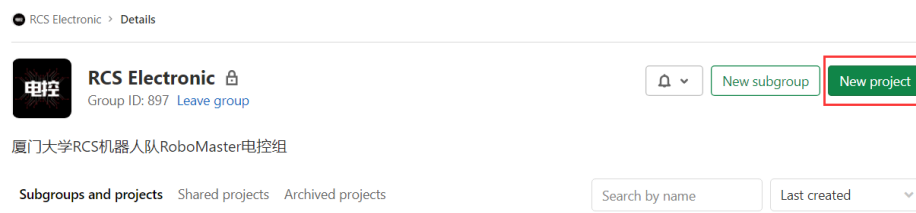
五、一个简单的多人协作流程举例

1、图示



2、项目组长创建新项目

第一步：在服务器端创建新工程



Blank project Create from template Import project

Project name
LiuChengJvLi 以项目名称的中文拼音命名

Project URL Project slug
https://git.xmu.edu.cn/ rcs-electronic liuchengjvli

Want to house several dependencies? [Create a group.](#)

Project description (optional)
Description format Groups
rcs-electronic 选择电控组
Users
shili

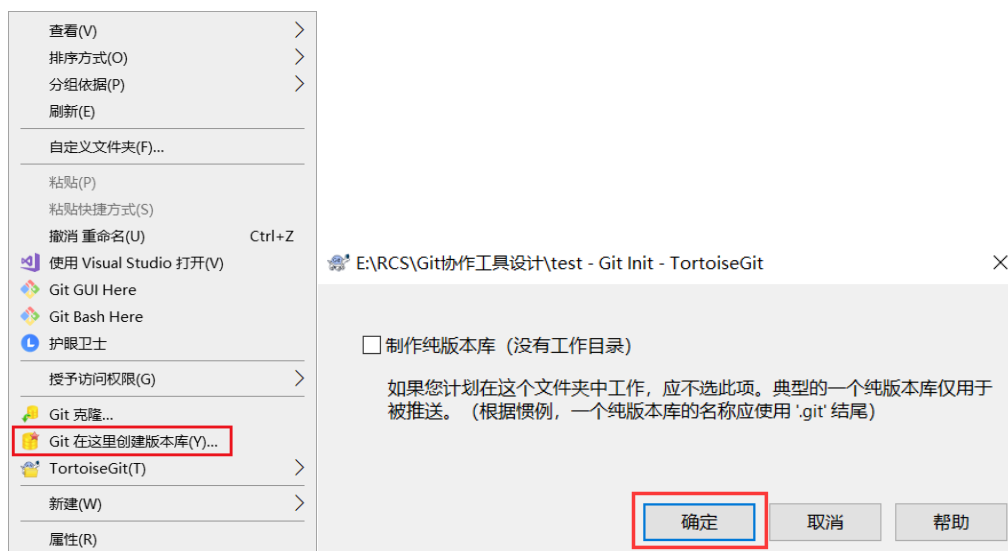
Visibility Level ⓘ
☒ Private
Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.
Other visibility settings have been disabled by the administrator.

☐ Initialize repository with a README
Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project Cancel

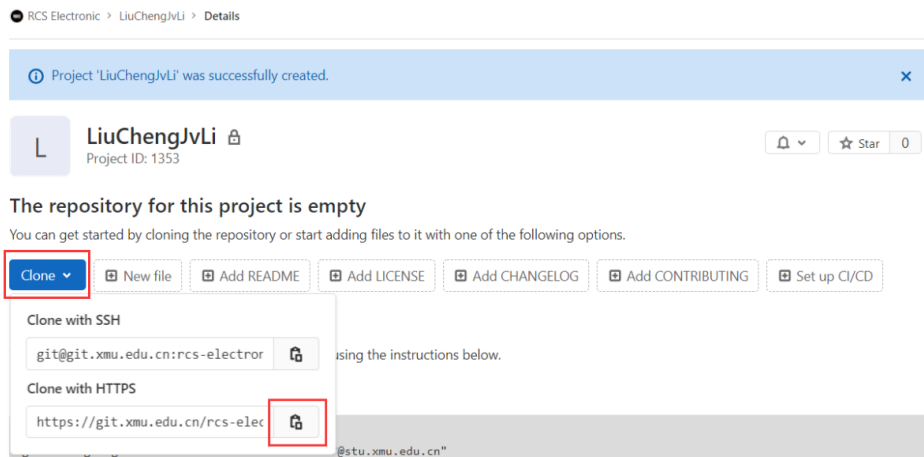
第二步：创建本地版本库

在本地的代码文件夹里点击右键，然后按图示操作



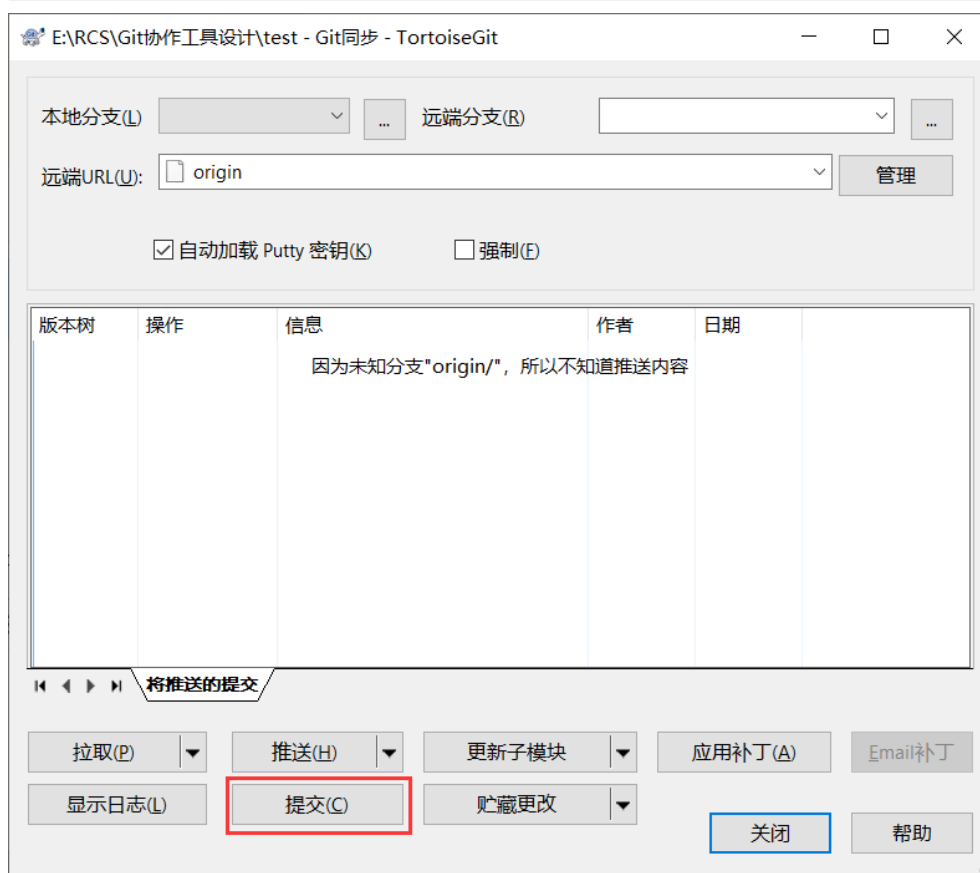
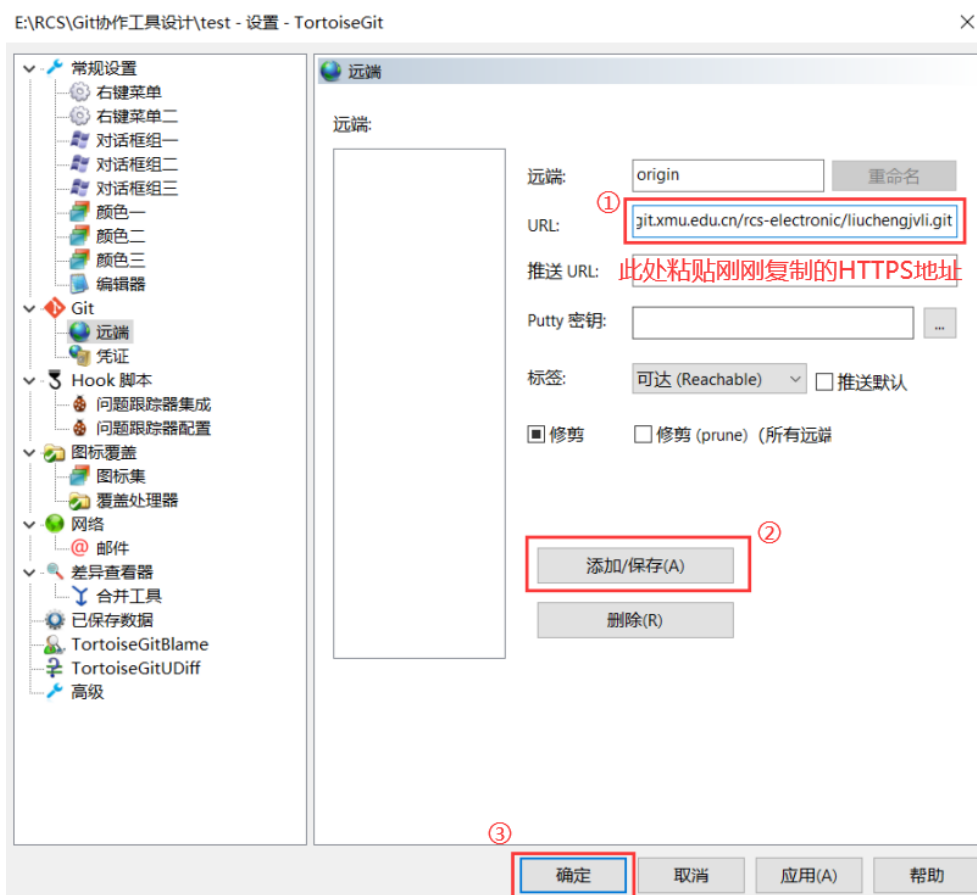
第三步：将服务器版本库与本地版本库关联

首先点击 Clone，然后点击图标复制 HTTPS 地址



在刚刚已经建好本地代码库的文件里单击右键，然后按图示操作





E:\RCS\Git协作工具设计\test - 提交 - TortoiseGit

提交至: master ☐ 新建分支

日志信息(M):

创建项目

☐ 修改上次提交(L) 养成良好的习惯 每次提交都要勾选 1/5

☒ 设置作者日期(D) 2020/12/12 16:28:10

☒ 设置作者(T) dengwendi <174202200196@stu 添加 "Signed-off-by" (S)

变更列表 (双击文件查看差异):

选中: **全部(A)** 无(N) 未版本控制 已版本控制 已添加 已删除 已修改 文件 子模块

路径	扩展名	状态	添加行数	删除行数
未受版本控制的文件				
<input checked="" type="checkbox"/> 123 (2).txt	.txt	未知		
<input checked="" type="checkbox"/> 123.txt	.txt	未知		

☒ 显示未受版本控制的文件(U) 已选中 2 个文件, 总计 2 个文件

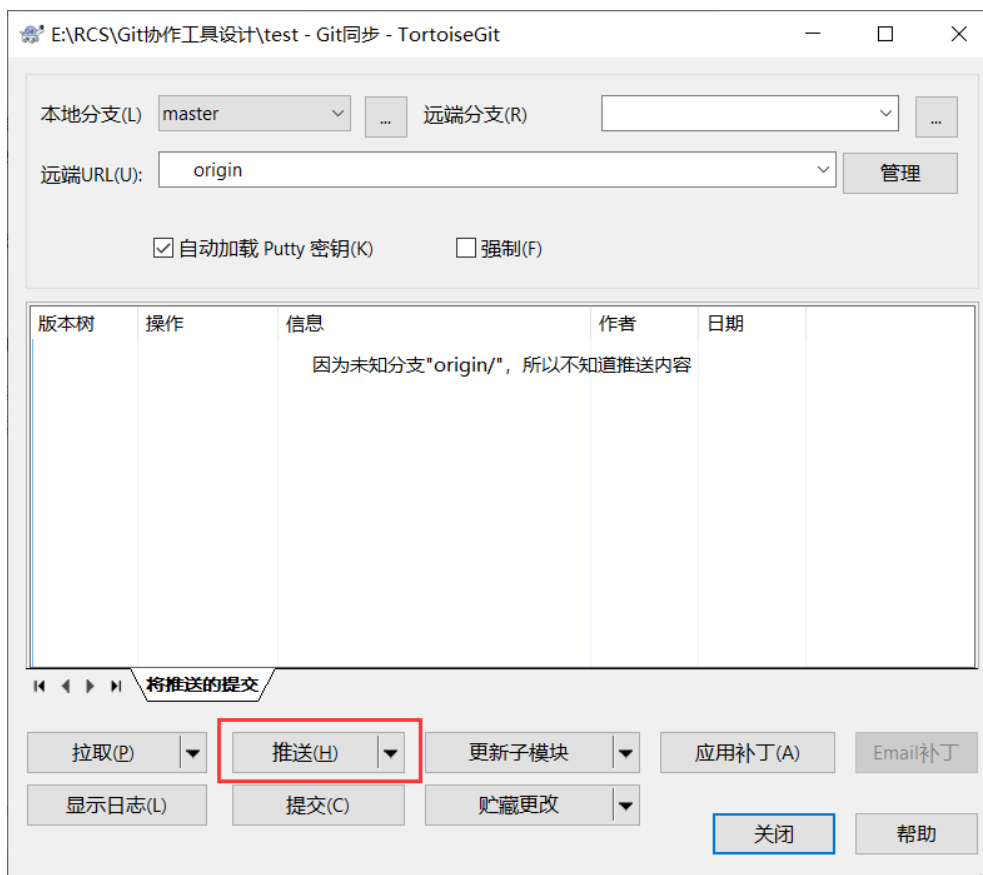
☐ 不自动选择子模块 [查看补丁>>](#)

☐ 显示整个工程(W)

☐ 仅仅消息(Y)

提交(O) 取消 帮助

提交完成后把刚才的窗口也关闭掉, 然后[右键->Git 同步...]重新打开一次



单击推送后会显示成功，这样就推送完成了，可以回到网站上看看推送的成果

L

LiuChengJvLi

Project ID: 1353

🔔

★ Star 0

🍴 Fork 0

🔗 1 Commit

🌿 1 Branch

🏷️ 0 Tags

📁 133 KB Files

💾 133 KB Storage

Auto DevOps
 It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.
[Learn more in the Auto DevOps documentation](#)

Enable in settings

master

liuchengjvli / +

History

Find file

Web IDE

📄

Clone

👤

创建项目
 邓文狄 authored 2 minutes ago

62434ed6

🔒

Add README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

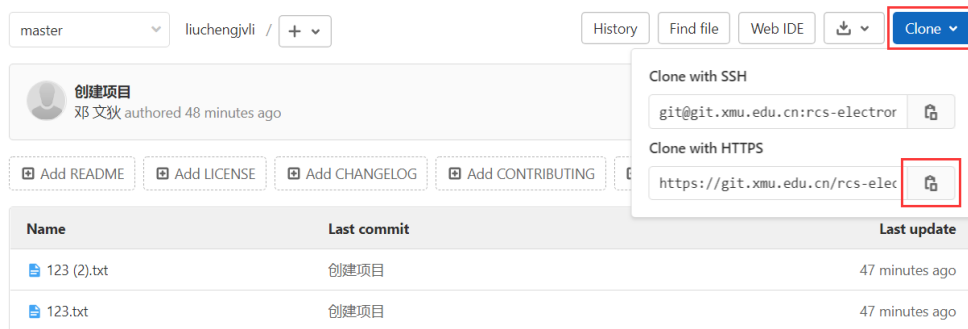
Add Kubernetes cluster

Set up CI/CD

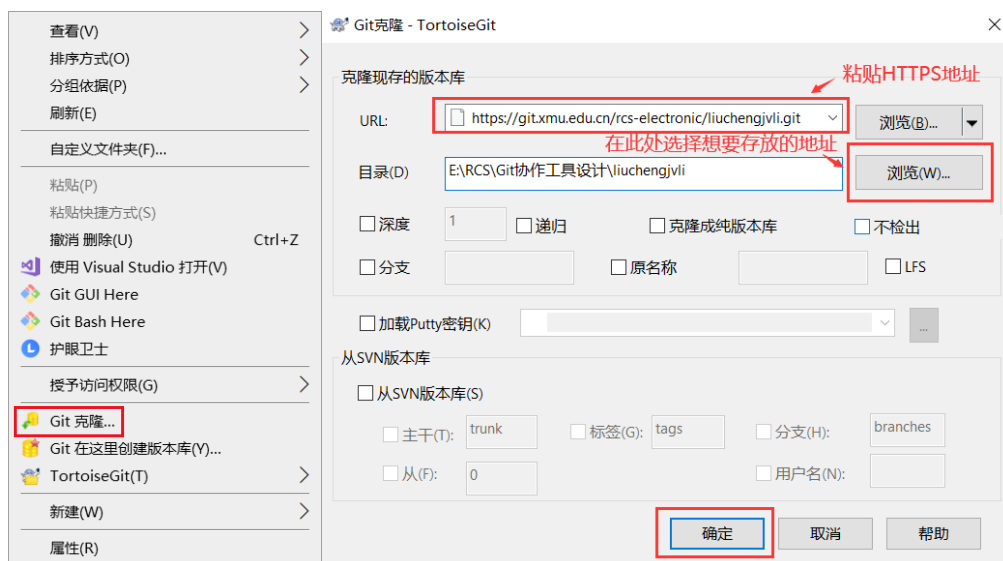
Name	Last commit	Last update
📄 123 (2).txt	创建项目	1 minute ago
📄 123.txt	创建项目	1 minute ago

3、组员抓取项目

打开项目组长已经建好的版本库，复制 HTTPS 地址



在电脑任意处单击右键，粘贴 HTTPS 地址，并选择想要存放本地版本库的目录

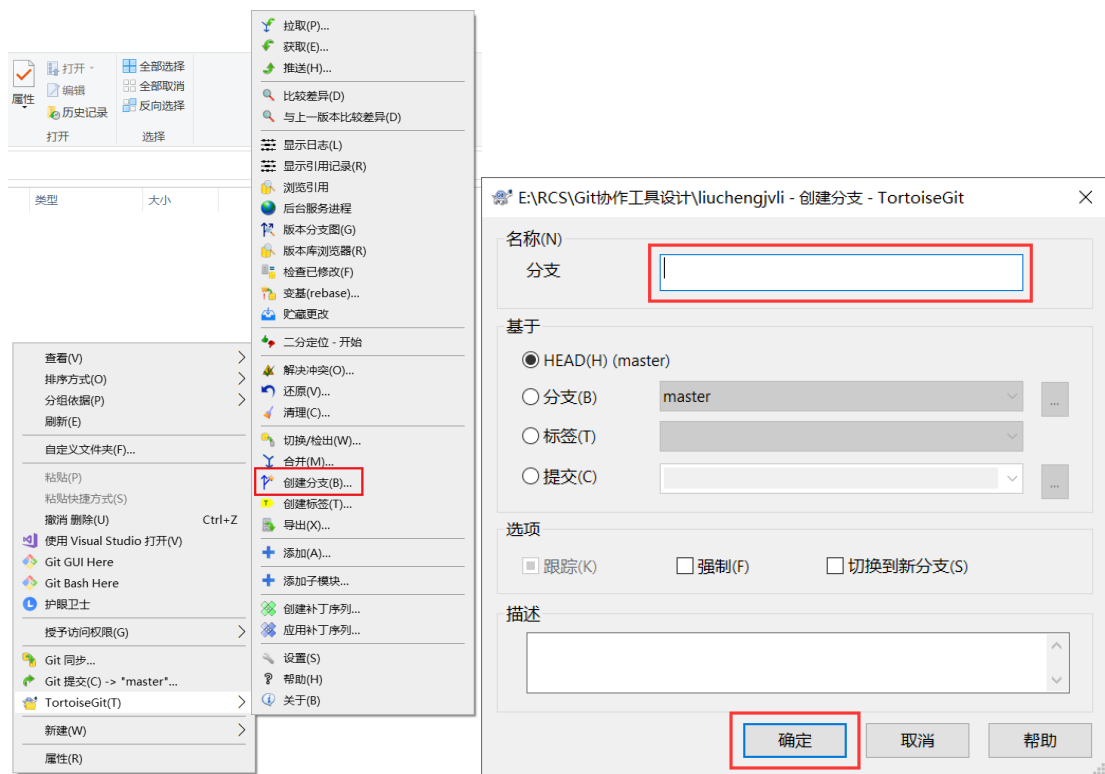


版本库即克隆完成。

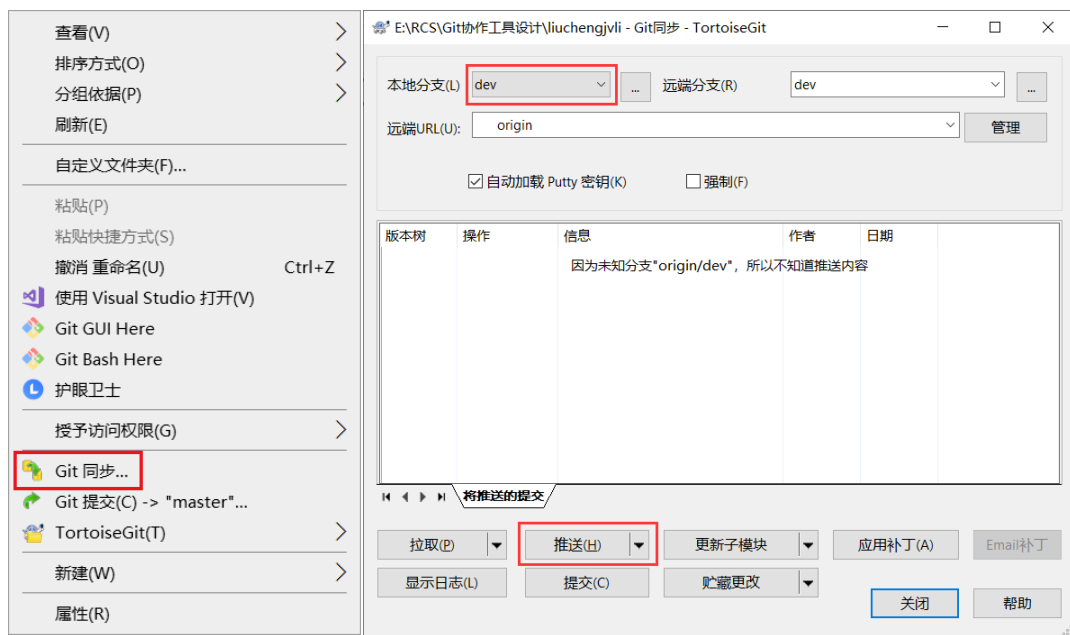
4、工作初始化

项目组长创建 dev 分支和个人分支，组员创建自己的个人分支

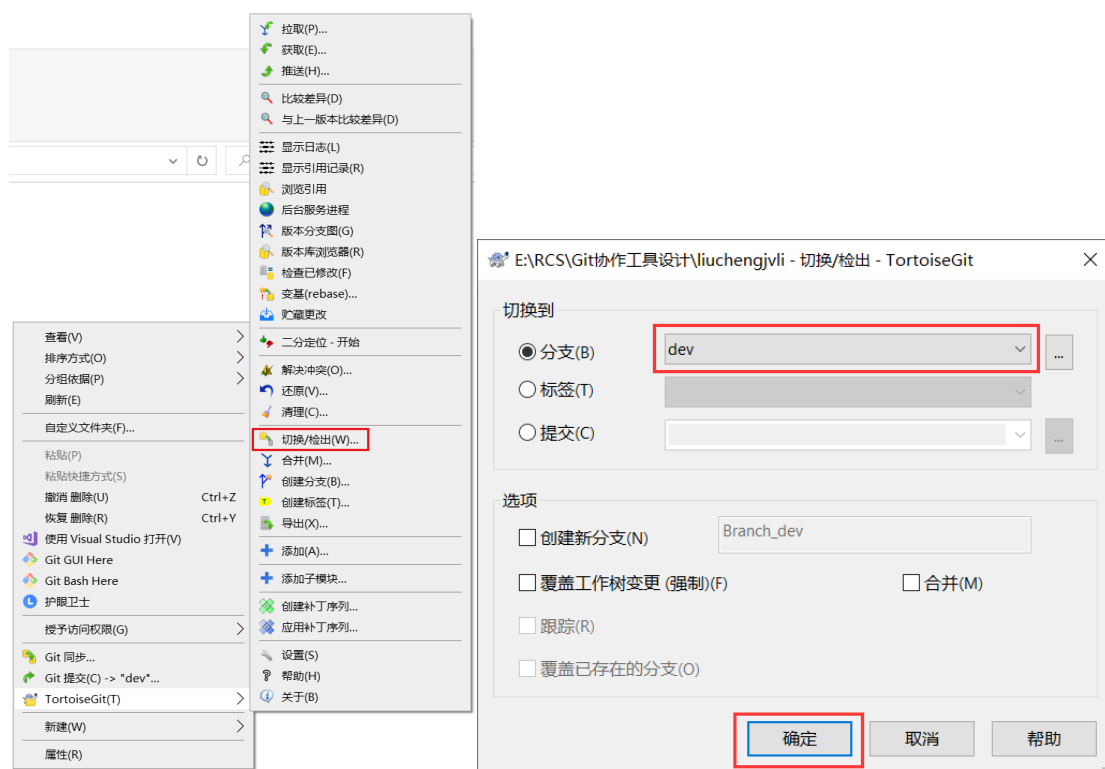
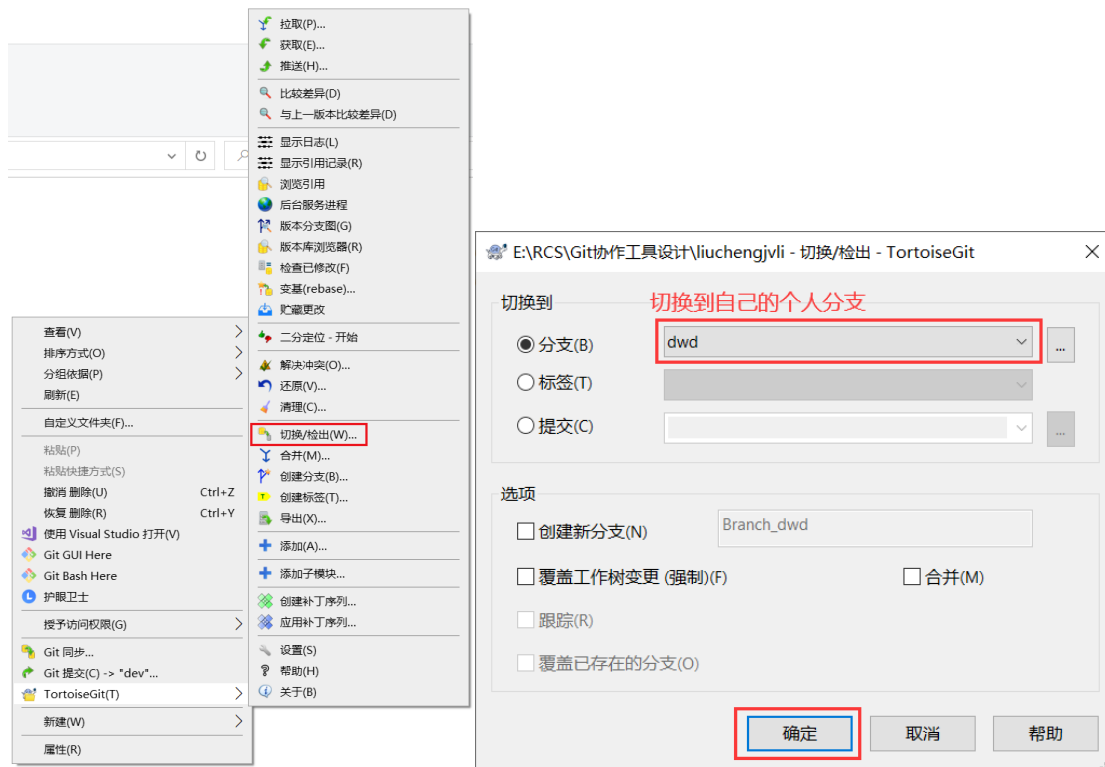
注意：个人分支统一以自己姓名的首字母拼音命名，如罗上聪命名为 lsc

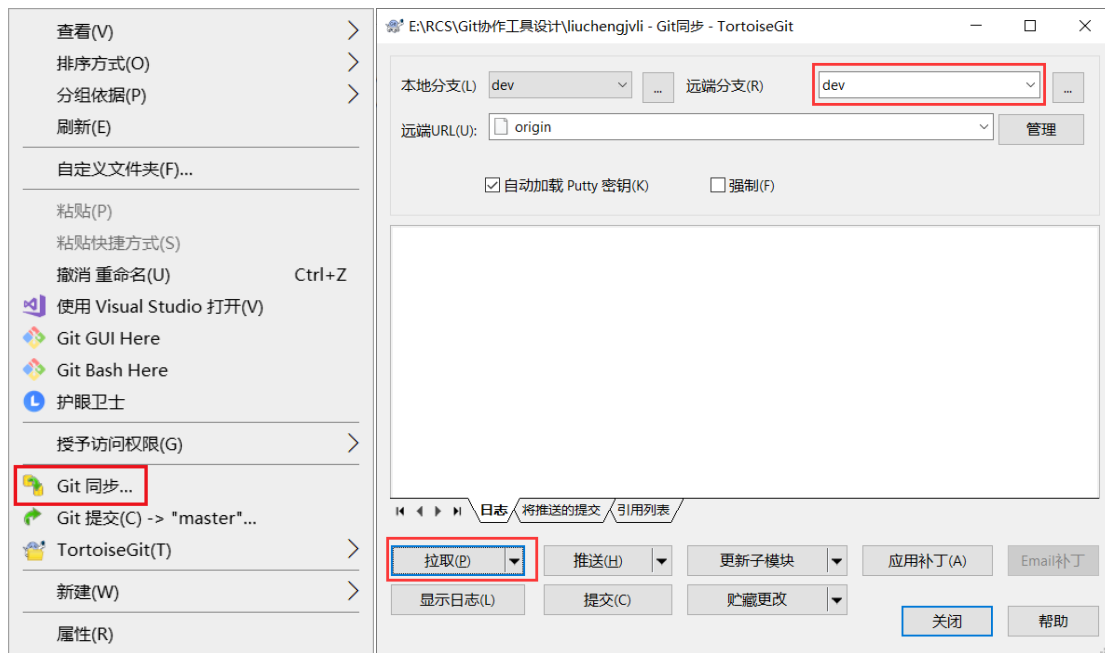


分支创建完成后，组长推送 dev 分支

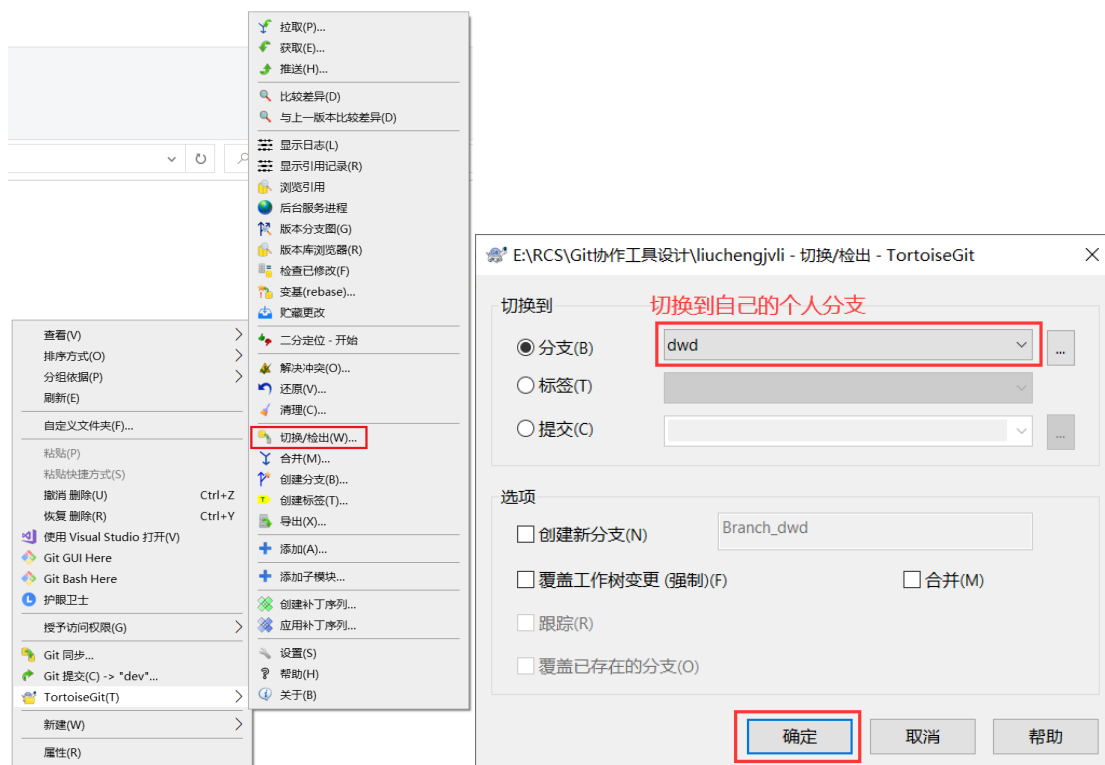


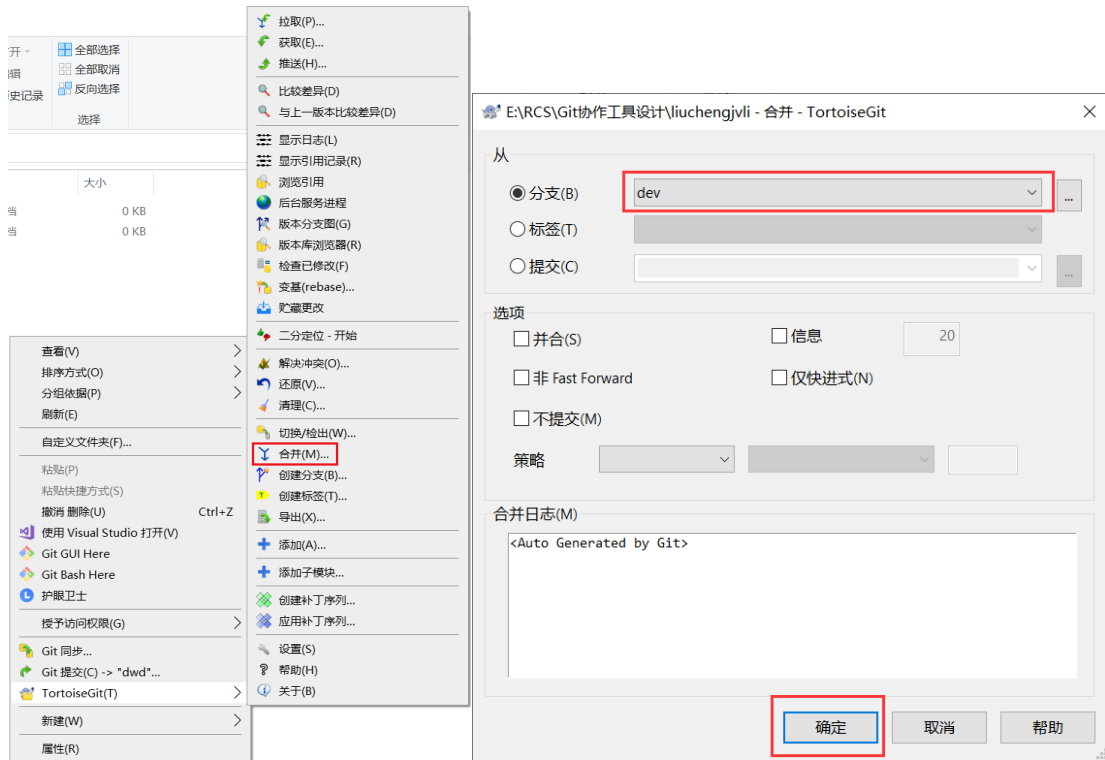
组长和组员切换到自己的个人分支





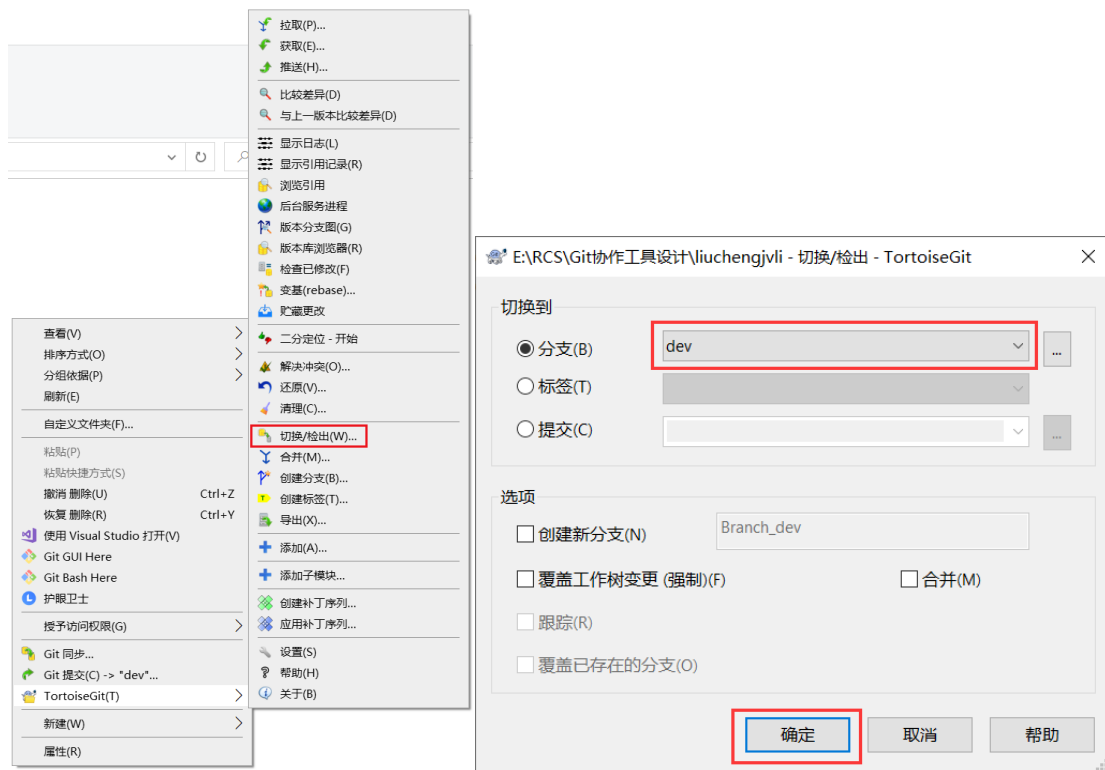
将 dev 分支内容合并到个人分支

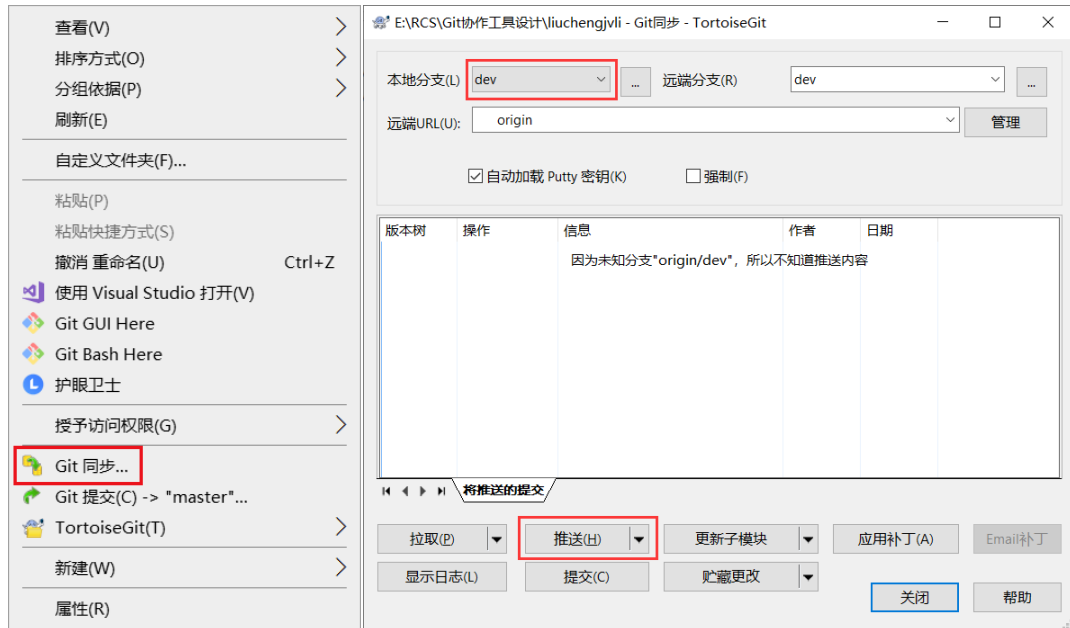




工作过程中分工明确，尽量避免不同成员同时修改同一部分内容

每次工作结束后，将个人分支内容合并到 dev 分支，并推送 dev 分支修改





(若出现分支合并冲突, 参见附录 [Git 教程.docx](#) 6.2 合并和解决冲突(P36))

七、附录

1、代码规范详细文档: 12 代码规范.pptx

<https://xmurcsrobot.feishu.cn/file/boxcncY0uDlIgNpTCfZjgsAkoLg>

2、TortoiseGit 简单教程(海韵机器人队).docx

<https://xmurcsrobot.feishu.cn/file/boxcn3WZnfZrlnYVLUvrZUqt8Fd>

3、Git 入门教程-廖雪峰: <https://www.liaoxuefeng.com/wiki/896043488029600>