git基本概念与使用

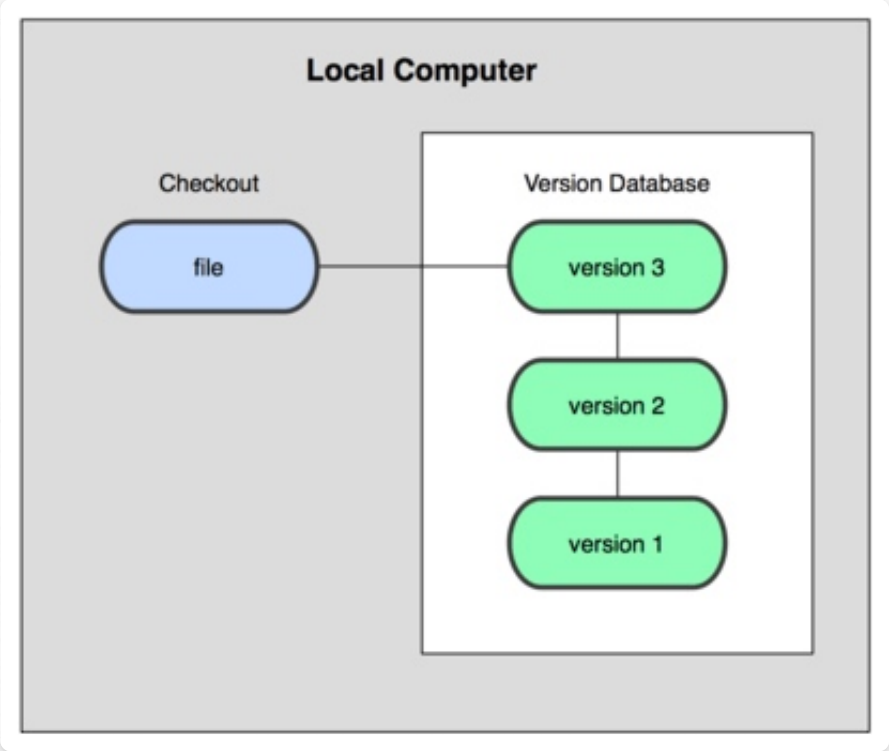
一、版本控制系统

版本控制系统： 一种记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的系统。

1.1 版本控制系统分类：

1.1.1本地版本控制系统

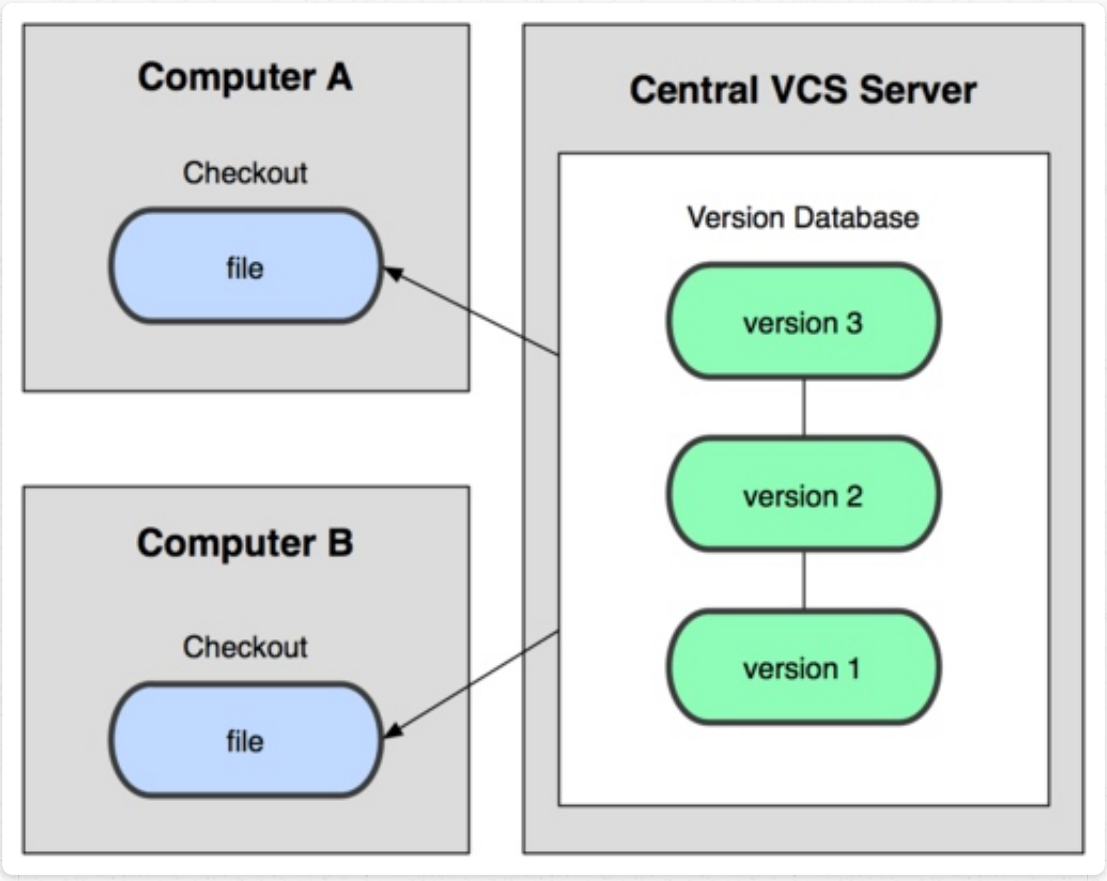
一般采用简单的数据库来记录文件的历次更新差异，如Mac OSX系统上的rcs。



1.1.2 集中化的版本控制系统

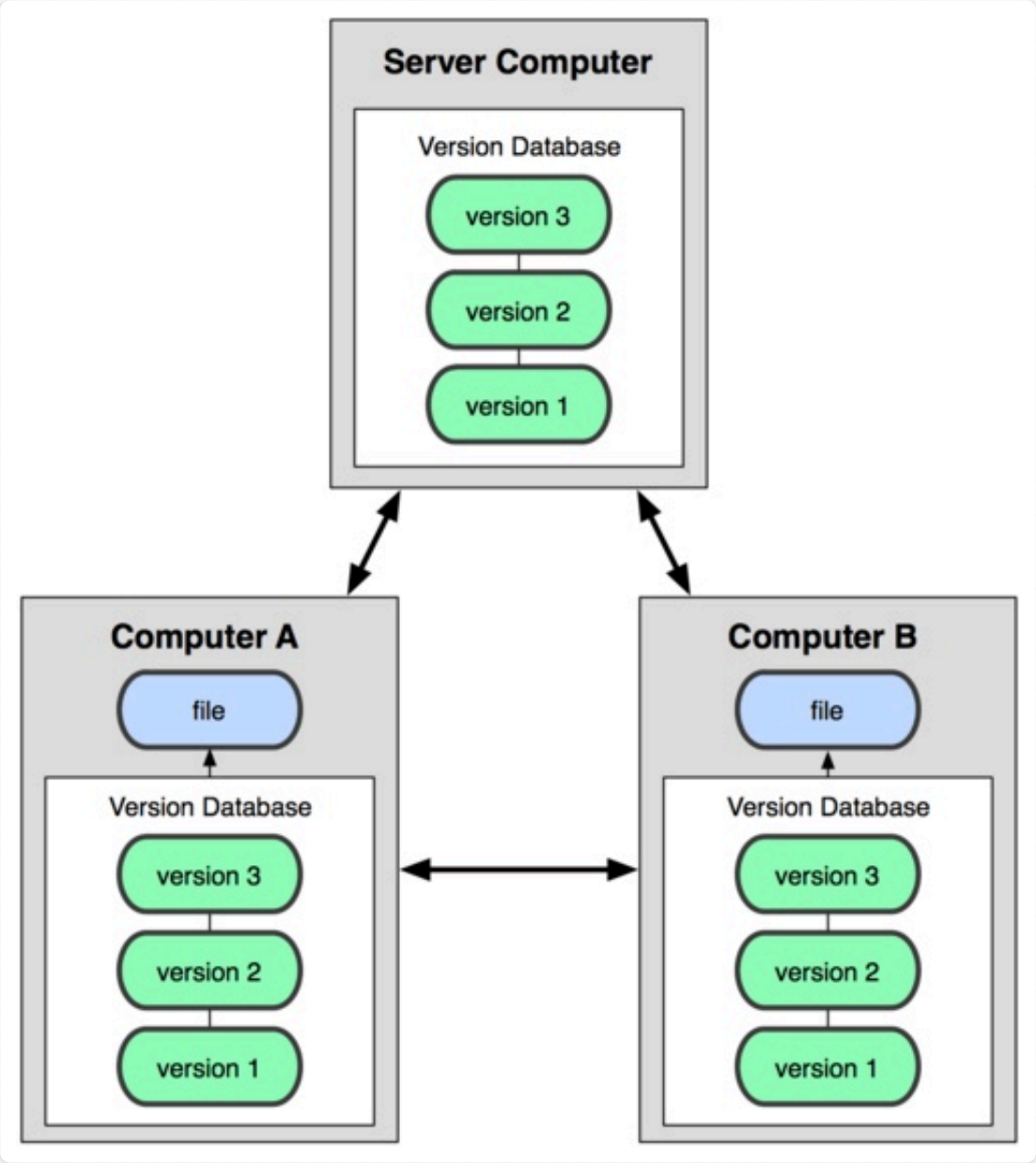
常用一个单一的集中管理的服务器，保存所有文件的修订版本，而协同工作的人们都

通过客户端连到这台服务器，取出最新的文件或者提交更新，如SVN系统。



1.1.3 分布式版本控制系统

这种系统中，客户端并不只是提取最新版本的文件快照，而是吧代码仓库完整的备份下来。任何一处协同工作用的服务器发生故障，事后都可以用任何一个镜像出来的本地仓库恢复。



二、git环境配置

gitlab是一个自托管的Git项目仓库服务软件，可通过Web界面进行访问公开的或者私人项目。 它拥有与GitHub类似的功能，能够浏览源代码，管理缺陷和注释。

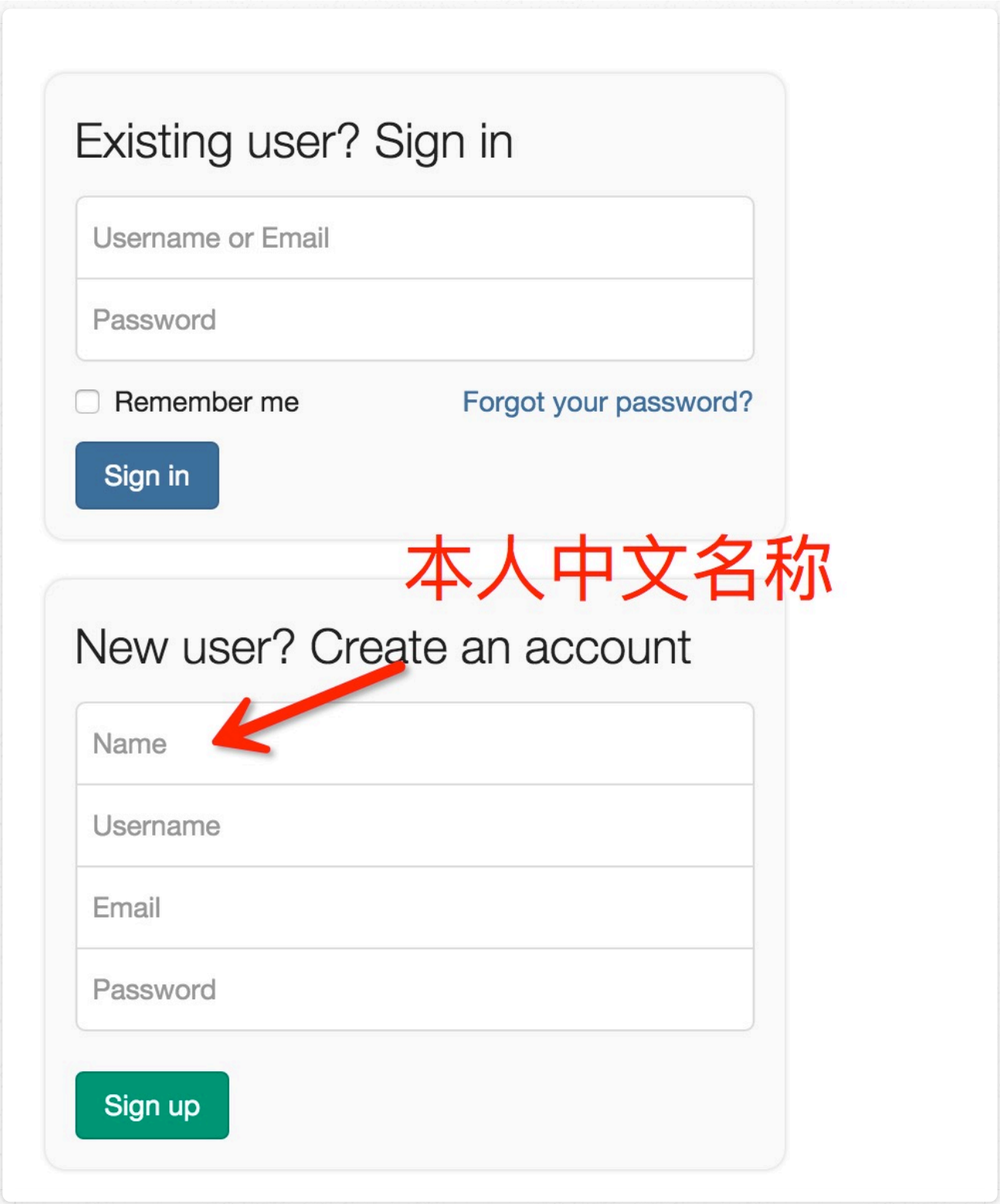
2.1 gitlab帐号申请

2.1.1 部门gitlab地址:

192.168.48.24

2.1.2 注册及要求:

根据注册页面提示信息填写名称、用户名、邮件地址、密码即可。为了方便管理与识别用户,名称请填写 **本人中文姓名**



2.2 软件安装

如果系统没有安装相应的git软件，请先安装。

- **Git客户端:** Git-2.6.3-32-bit.exe（必备）
- **Git gui客户端:** SourceTreeSetup_1.6.22.exe （可选,推荐）

2.3 ssh密钥生成与配置

2.3.1 生成密钥

执行如下命令，并一路回车：

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
```

注意: 如果一路回车，生成的密钥默认存放在用户目录下的.ssh文件夹下。

密钥对默认名称为：id_rsa 和 id_rsa.pub。

请确保同级目录下没有同名文件，以免被覆盖。

- 建议重命名 id_rsa 和 id_rsa.pub 这两个文件，以免被误覆盖
- winows用户目录一般为: C:\Users\用户名

2.3.2 配置密钥

将生成的密钥对私钥拷贝到 **用户目录下的.ssh目录**并加入该目录。

1.配置文件config

使用config配置文件，可以同时管理多个密钥，如果没有config文件则创建之。

文件名为 “***config***”，没有后缀！

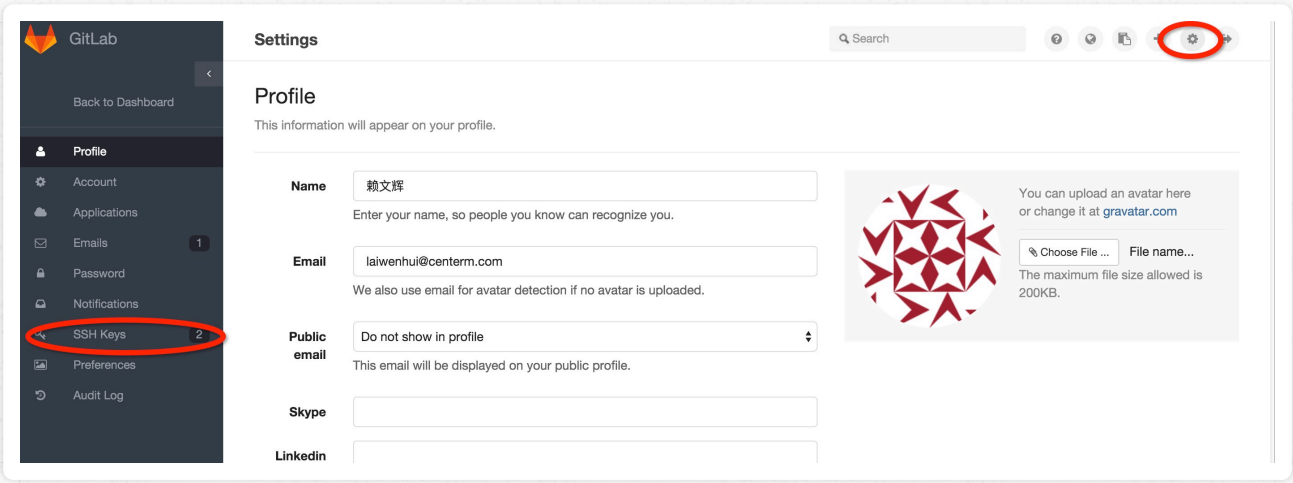
```
Host 192.168.48.24
    IdentityFile ~/.ssh/id_rsa
```

目录结构

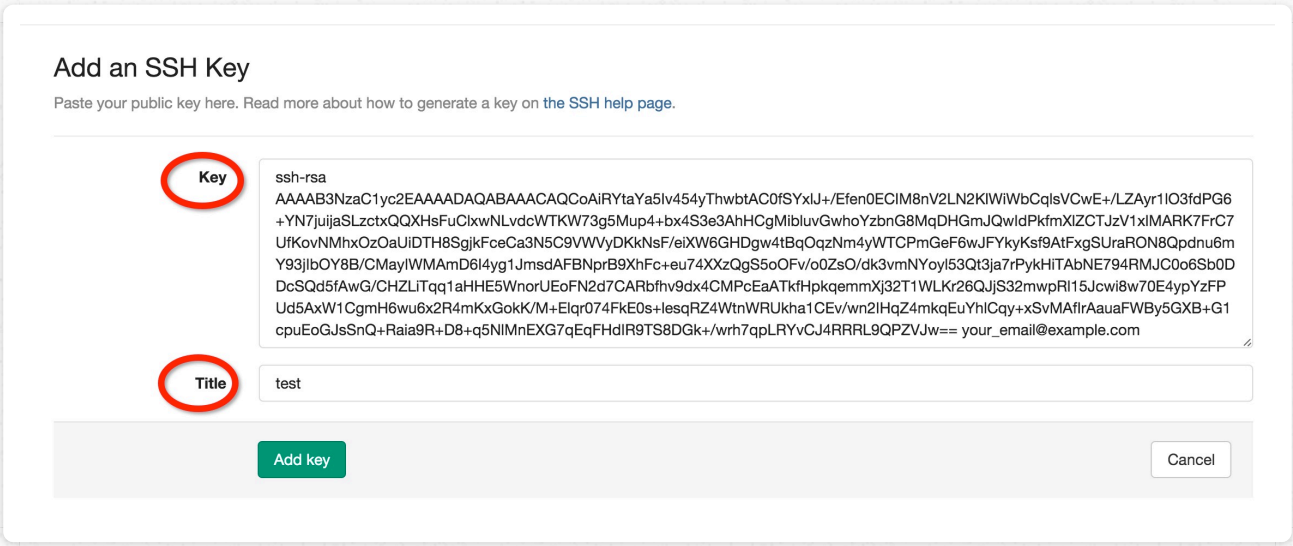
```
.ssh
├── config
├── id_rsa
└── id_rsa.pub
```

2.gitlab添加公钥id_rsa.pub

如图，登入gitlab后，点击setting



之后将公钥id_rsa.pub的 **全部内容**复制到key栏，点击 **Add key**按钮。



3.测试密钥

在终端程序上输入如下命令后，回车！

```
ssh -T git@192.168.48.24
```

如果返回 **Welcome to GitLab, XXXXXX!** 则表明密钥添加成功！

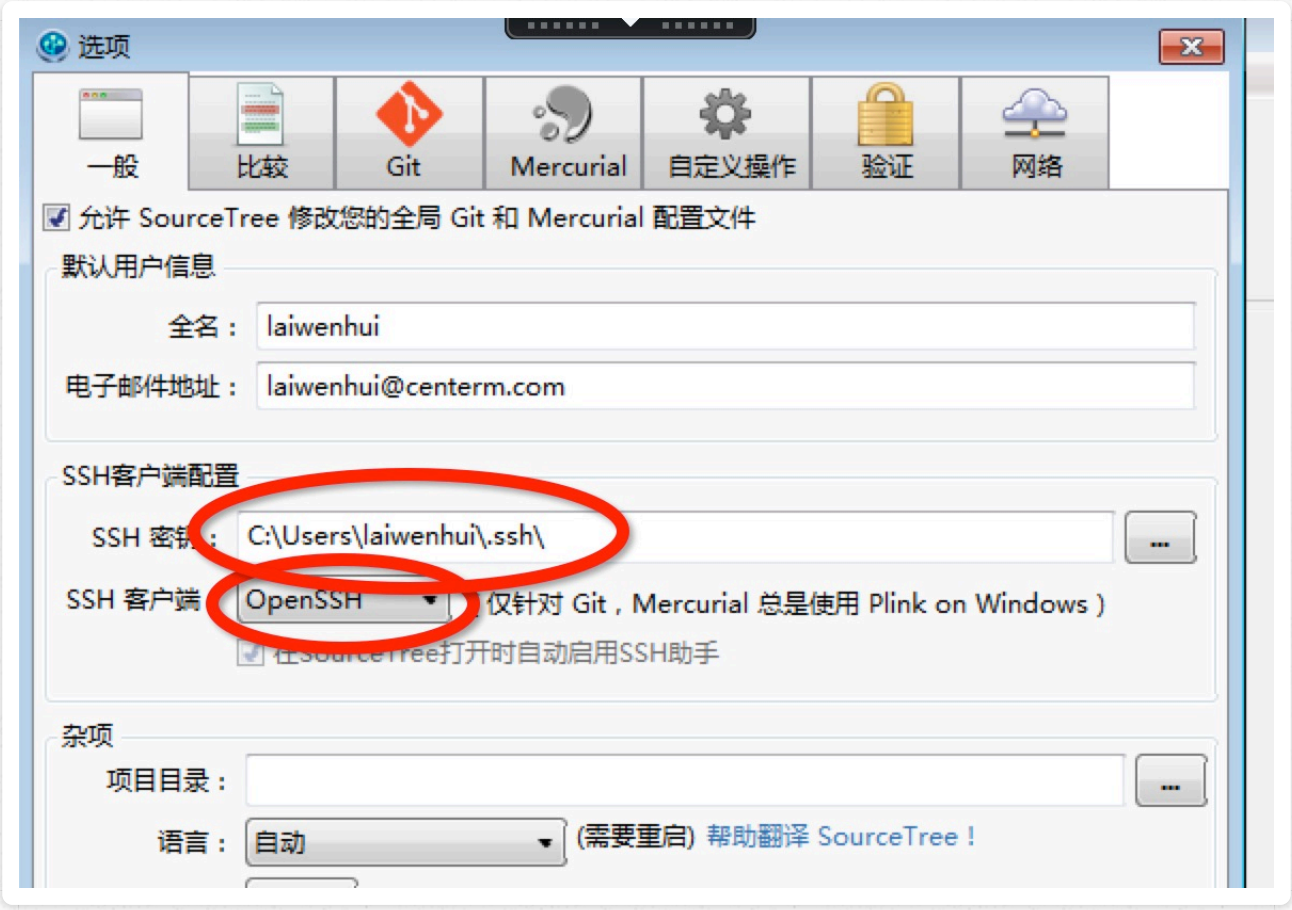
2.4、SourceTree配置

2.4.1 配置

启动source tree后，如果是第一次使用，则需要配置SSH客户端的密钥如下图:

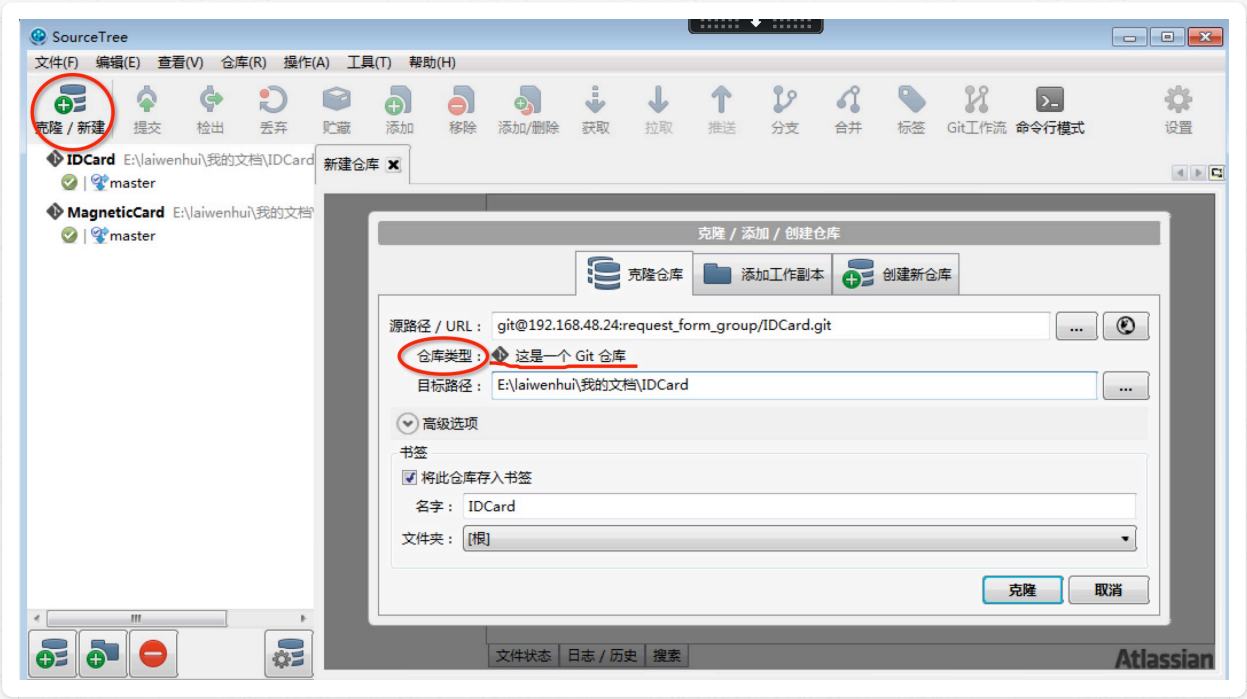
- 启动source tree后，按如下步骤操作: **工具栏-->工具-->选项-->一般**
- SSH客户端配置
    - **SSH客户端** 选择 **OpenSSH**

- **SSH密钥** 填入对应 **私钥存放的路径**(如: C:\Users\laiwenhui.ssh)，如果有多个存放路径，路径之间用分号隔开。



2.4.2 使用

- 1.将git工程路径输入到 *源路径*，如果路径没有问题或者权限没有问题，则会在 *仓库类型*:显示： **这是一个Git仓库**
- 2.在目标路径填入工程存放的地址
- 3.点击 **克隆** 即可 >软件的具体使用等待大家自己去学习



三、git常用操作

3.1 git克隆项目

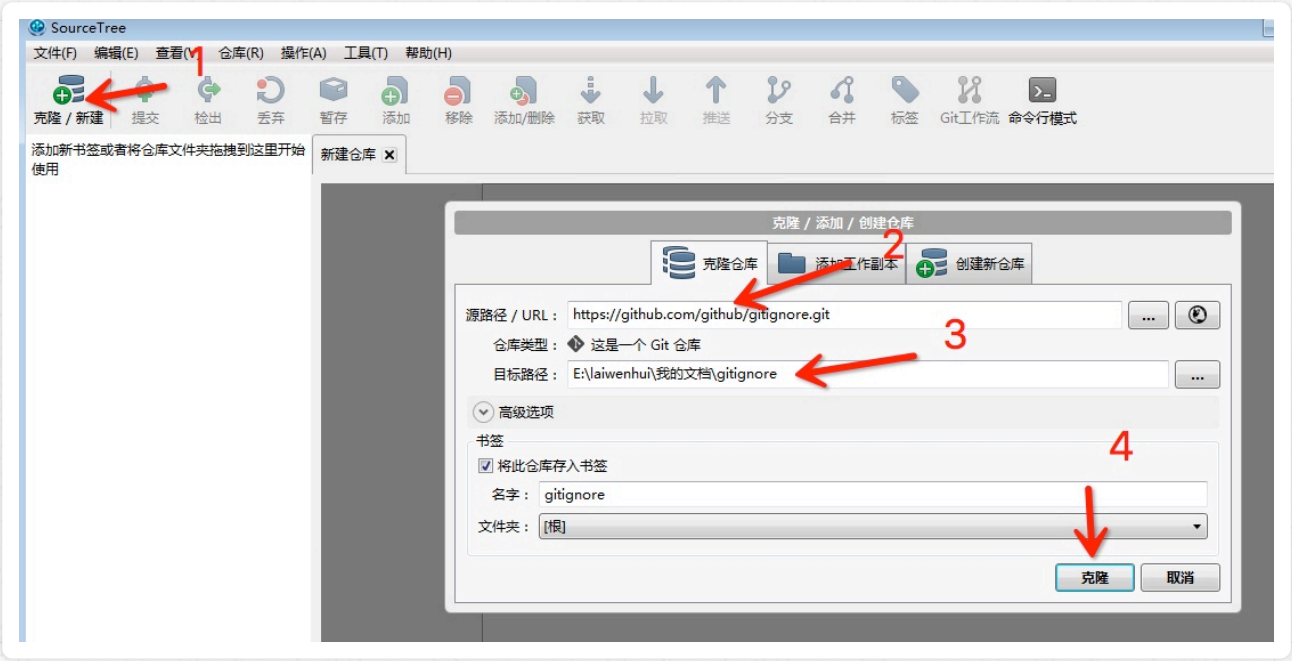
3.1.1 命令行操作:

```
git clone https://github.com/github/gitignore.git
```

3.1.2 SourceTree操作

如下图操作

1. 点击**克隆/新建**按钮
2. **源路径中**填入项目地址
3. 填写项目保存路径
4. 点击**克隆**按钮



3.2 git 提交、撤销

场景：你修改了main.c问题，并需要提交修改，之后发现提交有误需要撤销

3.2.1 命令行操作:

1. 查看项目修改情况（可选）

```
git status
```

1. 将修改文件提交到暂存区

```
git add main.c
```

1. 查看文件提交到暂存区是否成功（可选）

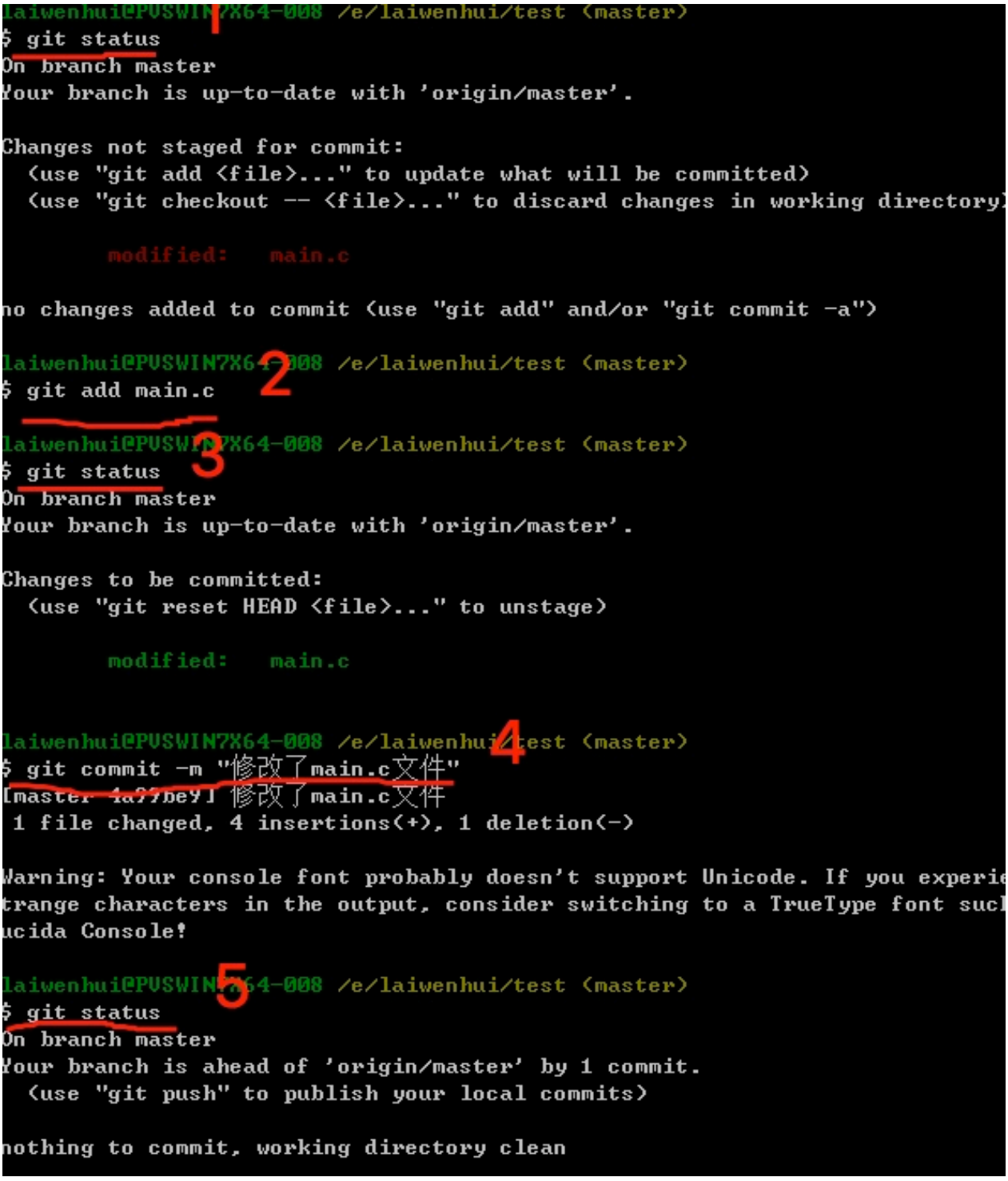
```
git status
```

4.提交

```
git commit -m "修改了main.c文件"
```

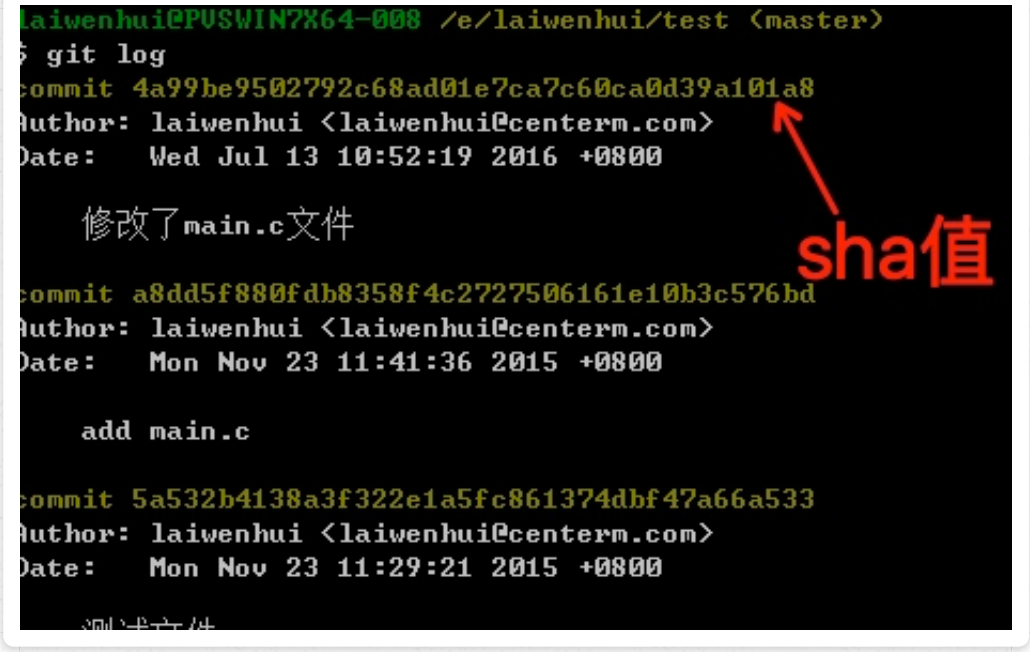
1. 查看提交是否成功（可选）

```
git status
```



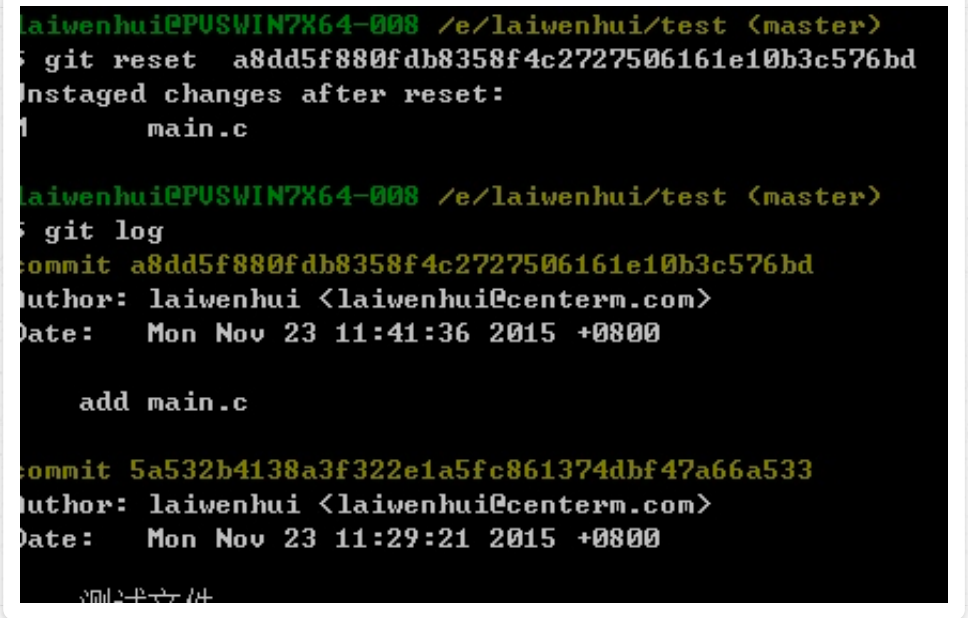
6. 查看提交记录

```
git log
```



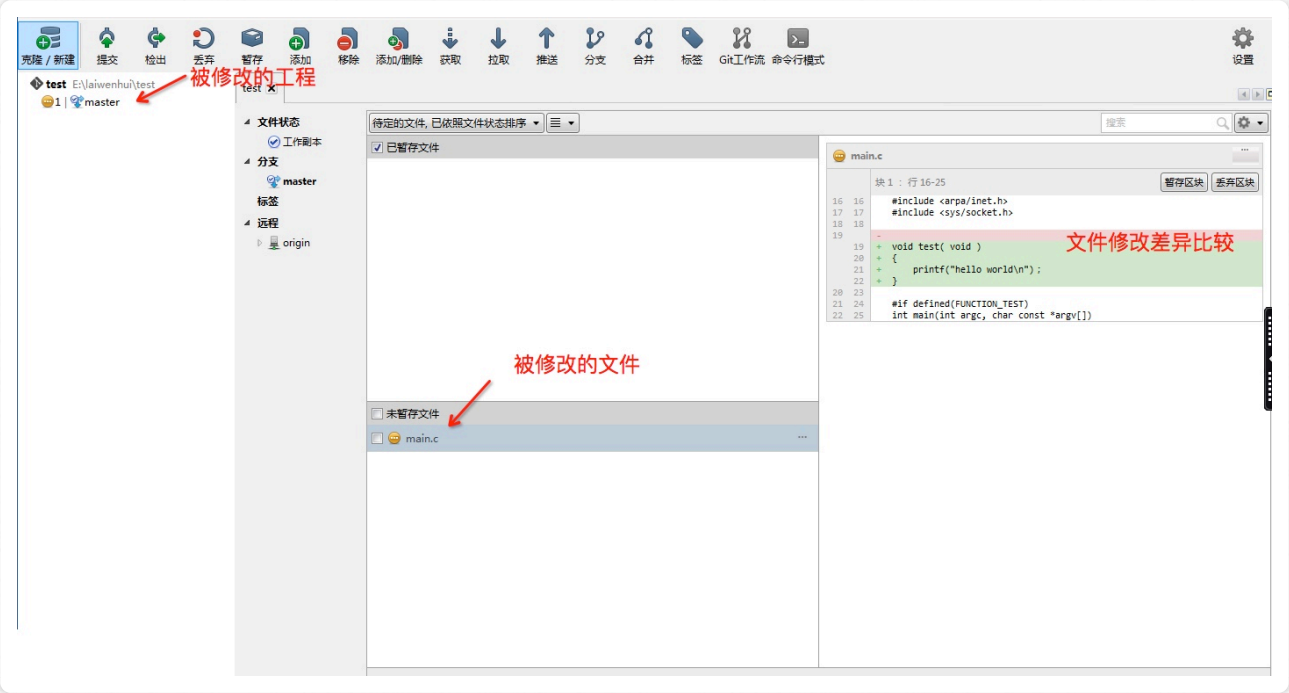
7. 撤销提交

命令格式：git reset <提交的sha值>，执行后，文件状态恢复成提交前的状态
git reset a8dd5f880fdb8358f4c2727506161e10b3c576bd

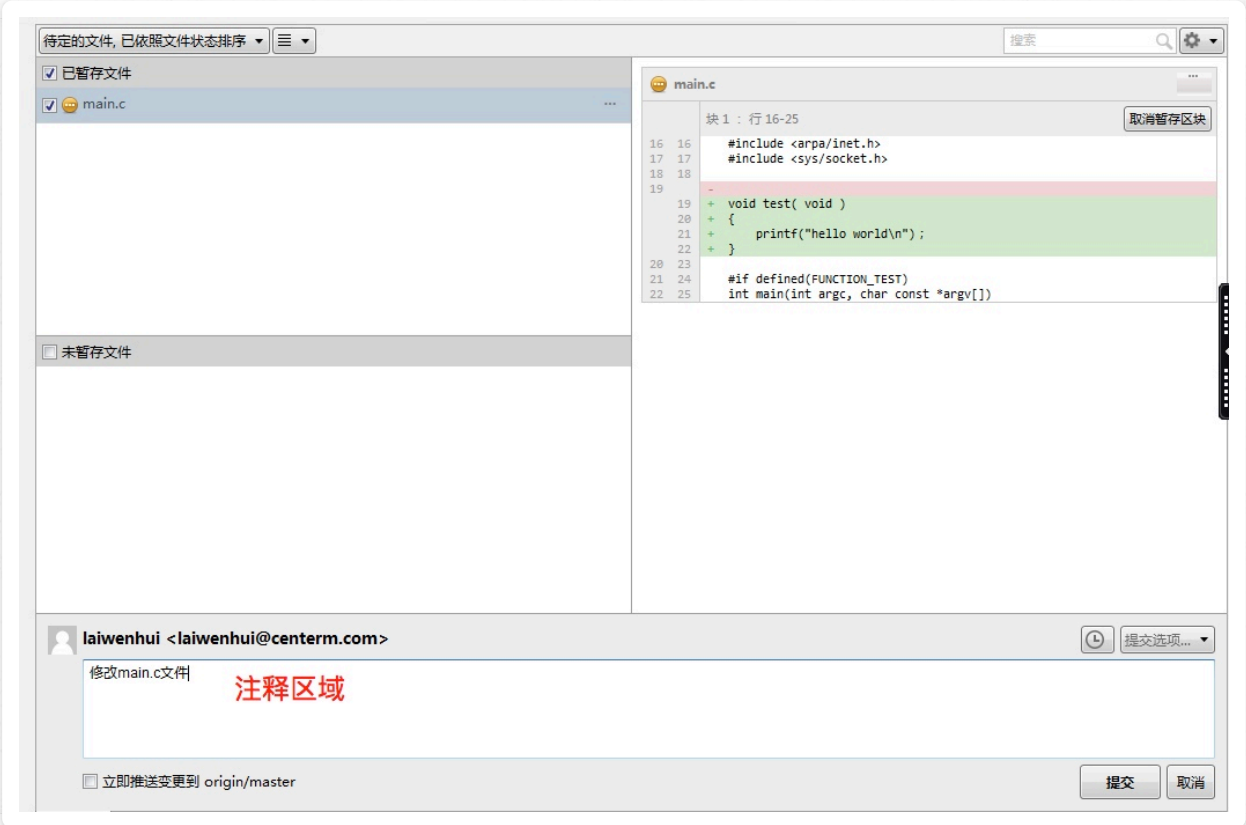


3.2.1 SourceTree操作

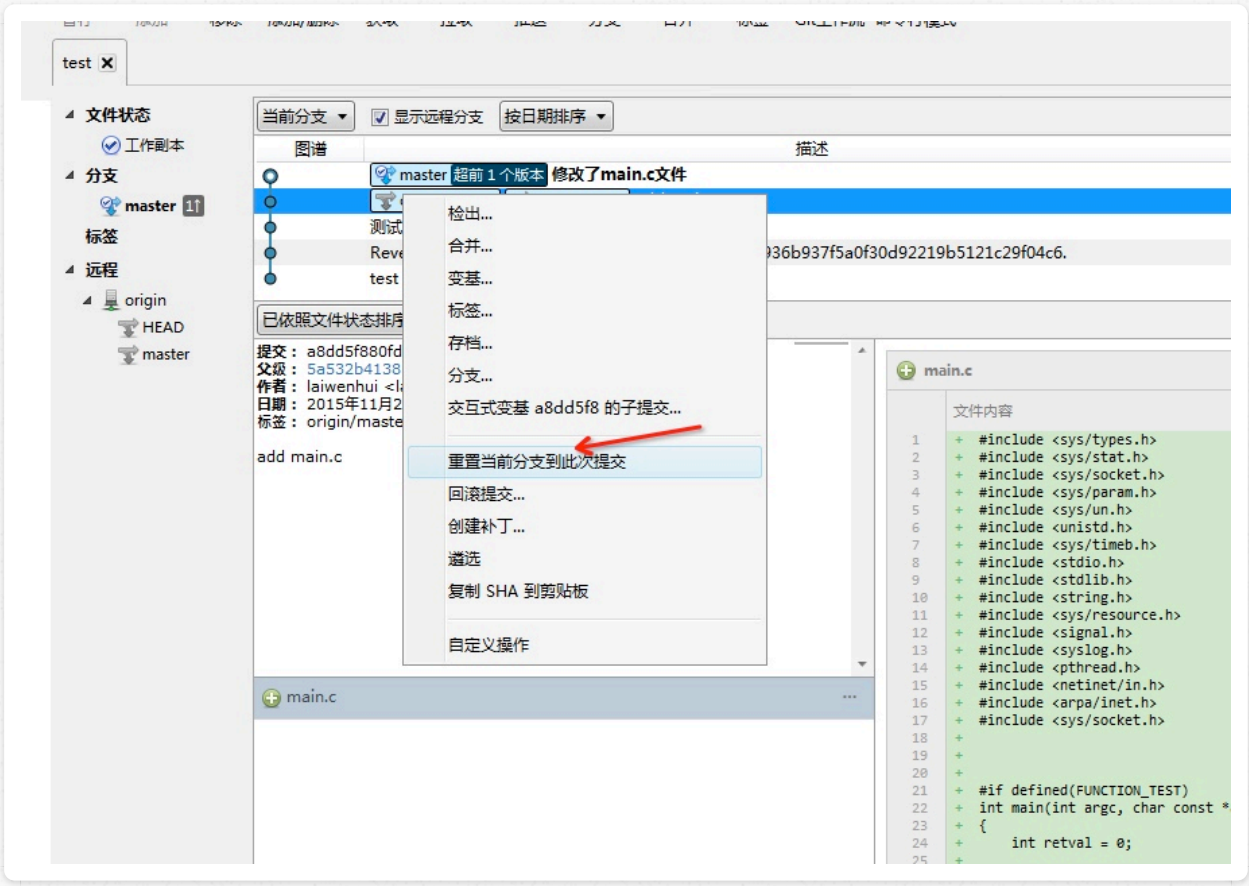
1. 要提交修改，选择被修改的文件main.c，sourcetree自动将其添加到暂存区。



1. 在注释区，写入提交信息，点击**提交**按钮即可。



2. 撤销提交，点击**分支master**，右侧会显示提交的历史记录。选择要回退到的提交记录上，右击选择
重置当前分支到次提交即可。



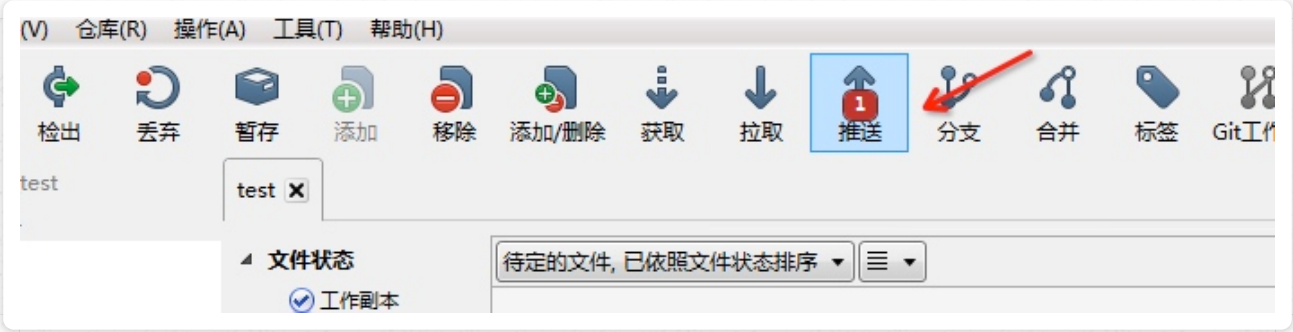
3.3 git推送提交到远程代码仓库

3.3.1 git命令行操作

```
git push –u origin master
```

3.3.2 SourceTree操作

如下图点击**推送**按钮即可



四、git重要概念

4.1 基本特点

4.1.1 Nearly Every Operation is local

1. 本地就是一个完整的仓库，包含所有的历史记录
2. 无需联网，就可以直接查看历史或提交

4.1.2 Git has a id

1. 每个版本基于文件或者文件夹计算得到的SHA1哈希值
2. 以哈希值作为版本id，用于版本的检出、撤销回退等

4.2工作区和暂存区

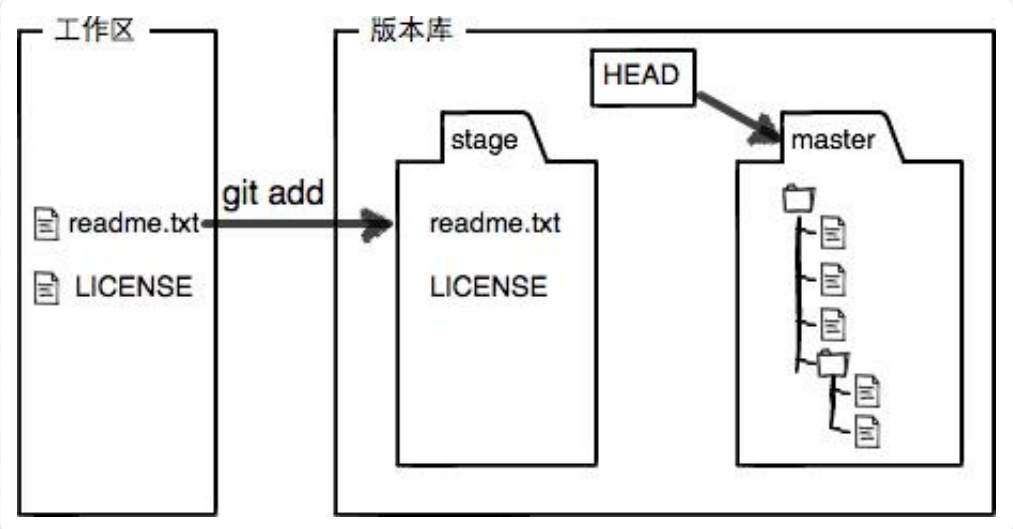
4.2.1 工作区

就是项目的当前目录，如test目录.

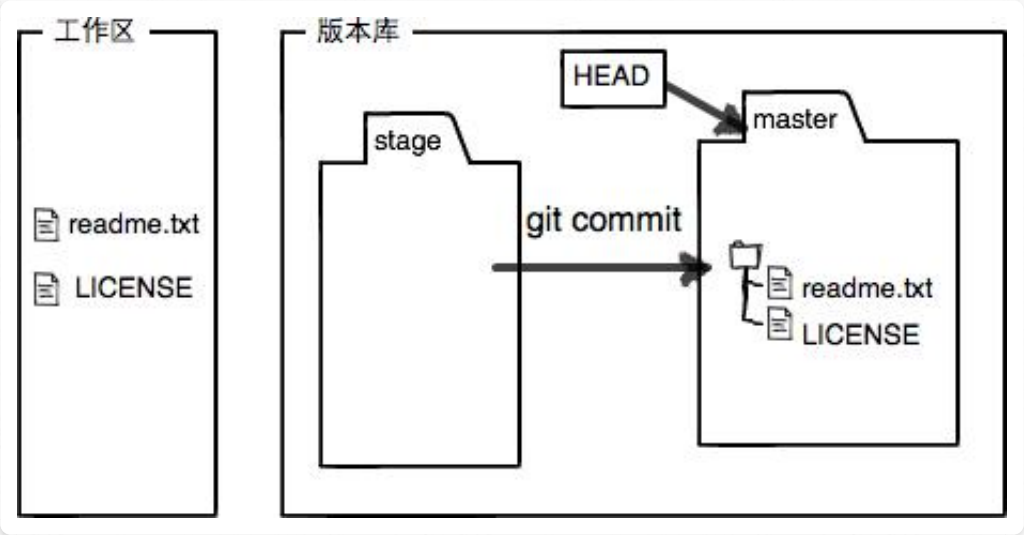
4.2.2 暂存区

在工作区中有个隐藏目录.git(注：Windows系统是可见的）,这个文件就是Git的版本库（Repository），里面保存了很多文件信息，其中有个称为stage的暂存区，通过git add命令添加的文件就保存在这个区里。

```
git add readme.txt
git add LICENSE
```




```
git commit -m "add readme.txt and LICENSE"
```



五、深入学习

以上都是git方面最基础的知识点，能能够满足日常工作的基本需求。深入学习可以参考如下链接。

1. http://blog.jobbole.com/87700/
2. http://www.liaoxuefeng.com/wiki/001373951630592960 6dd18361248578c67b8067c8c017b000