

# 作业提交配置手册

本学期通过GitLab提交代码作业，提交配置过程如下：

在配置过程中如果提示某个指令需要安装，则执行 `sudo apt install 软件名` 安装对应软件即可。

## 1. 在Ubuntu中添加SSH认证

在terminal中执行 `ssh-keygen -t ed25519`，执行后按照提示输入内容，默认输入三次回车即可：

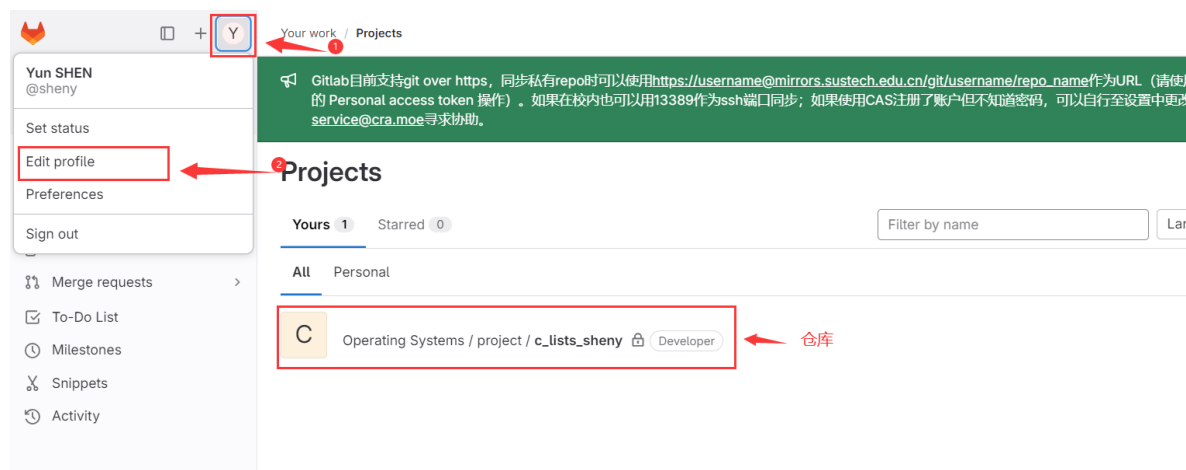
```
sy@sy-virtual-machine:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/sy/.ssh/id_ed25519):
Created directory '/home/sy/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sy/.ssh/id_ed25519
Your public key has been saved in /home/sy/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:v8Ga... sy@sy-virtual-machine
The key's randomart image is:
+--[ED25519 256]--+
|
|      o .
|      =
|      +
|      .
|ob   .
|      +
|      o
|      .
|B*   .
|      +
+-----[SHA256]-----+
```

成功后，在 `~` 路径下会出现一个 `.ssh` 文件夹，其中有个 `id_ed25519.pub` 文件，里面就是你的ssh密钥。

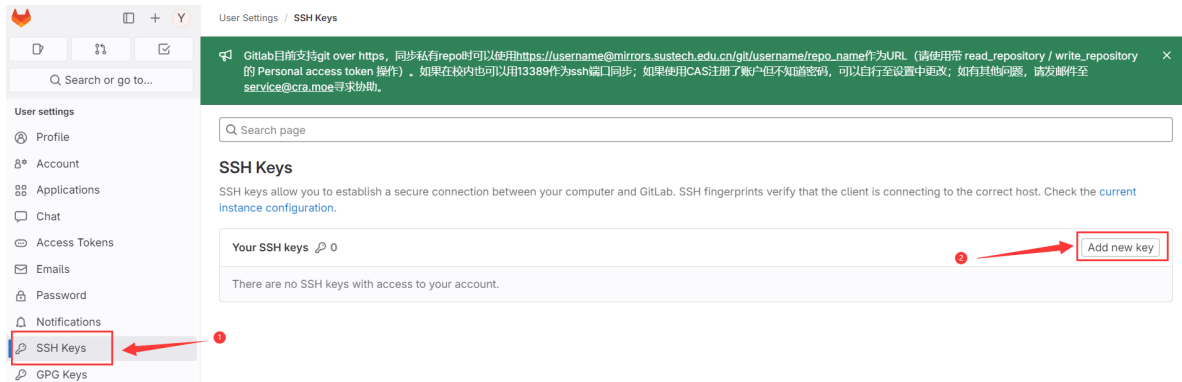
## 2. 在GitLab添加密钥

通过CAS账号登陆<https://mirrors.sustech.edu.cn/git/>，已经为选课的同学添加了私人仓库。

点击Edit profile



找到SSH Keys, 点击Add new key



将 id\_ed25519.pub 文件中的内容粘贴到Key, Usage type需要包含Authentication, 过期时间至少设置到学期结束。

### SSH Keys

Add an SSH key for secure access to GitLab. [Learn more.](#)

**Key**

ssh-ed25519 AAA...tF sy@sy-virtual-machine

Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

**Title**

sy@sy-virtual-machine

Key titles are publicly visible.

**Usage type**

Authentication & Signing

**Expiration date**

2025-02-26

Optional but recommended. If set, key becomes invalid on the specified date.

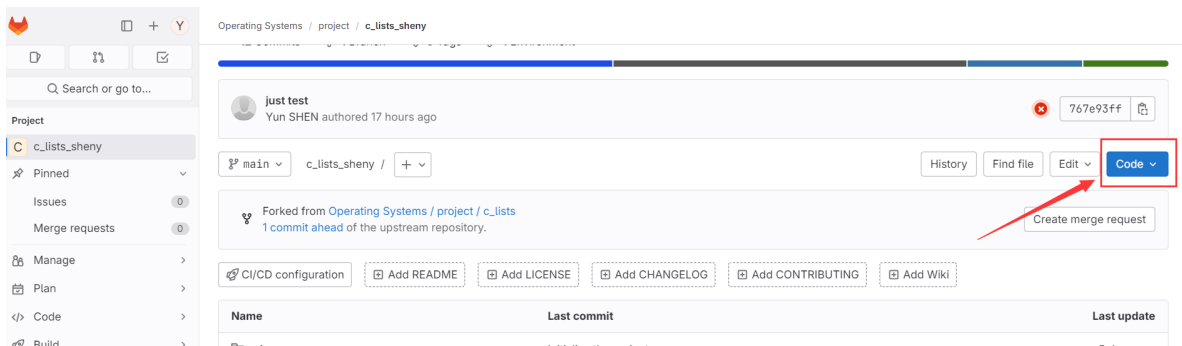
[Add key](#) [Cancel](#)

## 3. 将仓库clone至本地

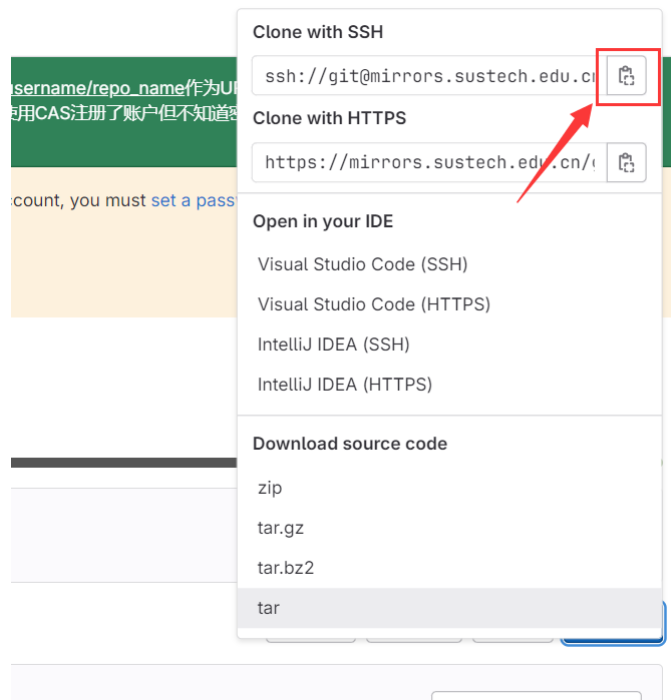
在terminal中通过cd指令跳转至你想要存放仓库内容的工作目录, 例如我在桌面创建了一个git文件夹用于存放仓库的内容:

```
sy@sy-virtual-machine:~/Desktop$ cd git
sy@sy-virtual-machine:~/Desktop/git$
```

通过<https://mirrors.sustech.edu.cn/git/>, 进入自己的仓库, 点击右上方蓝色Code按钮:



点击Clone with SSH右侧复制按钮:



将复制到的文本中的13098更改为13389，在terminal中执行 `git clone` 复制的文本 指令，输入yes，仓库内容会被clone至执行指令的本地工作目录：

```
sy@sy-virtual-machine:~/Desktop/git$ git clone ssh://git@mirrors.sustech.edu.cn:13389/operating-systems/project/c_lists_sheny.git
Cloning into 'c_lists_sheny'...
The authenticity of host '[mirrors.sustech.edu.cn]:13389 ([172.18.6.198]:13389)' can't be established.
ED25519 key fingerprint is SHA256:q6Vq5/Gl0PMQb1HT1nBBrtHZ8+XkcjLHjZ3tp5UulHY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? Y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[mirrors.sustech.edu.cn]:13389' (ED25519) to the list of known hosts.
remote: Enumerating objects: 71, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 71 (delta 13), reused 7 (delta 7), pack-reused 52
Receiving objects: 100% (71/71), 18.37 KiB | 3.67 MiB/s, done.
Resolving deltas: 100% (21/21), done.
sy@sy-virtual-machine:~/Desktop/git$ ls
c_lists_sheny
```

## 4. 提交代码至仓库

第一次使用时可以对git进行基本设置：

```
#设置用户名
git config --global user.name "sy"

#设置邮箱
git config --global user.email "sheny@mail.sustech.edu.cn"
```

提交代码时需要切换至仓库所在的本地工作目录，执行下列指令：

#将更改过的文件或者新文件提交到暂存区

```
git add src/main.c
```

#为本次更改创建一个提交对象，并保存至本地仓库，-m可以为本次提交添加一些信息

```
git commit -m "message"
```

#首次将提交推送至远程仓库

```
git push -u origin main
```

#之后推送只需要执行

```
git push
```

之后我们会对同学们提交到仓库的代码进行测评。

## \*5. 测试用作业

为大家熟悉作业提交流程及C语言指针，我们提供了一次测试用的作业供大家提交，注意本次作业不强制作提交，也不占任何课程分数，仅供大家熟悉环境使用。

下面是完成作业的具体流程：

- 1) 将仓库代码clone至本地仓库，作业为完成src文件夹中main.c文件中的TODO内容
- 2) 将作业测试环境镜像pull至本地（此镜像仅用于本次作业）

```
podman pull glcr.cra.ac.cn/operating-systems/project/kernel_legacy/ucore-tools:v0.0.10
```

或（此镜像包含更多开发工具）

```
podman pull glcr.cra.ac.cn/operating-systems/project/kernel_legacy/ucore-dev-tools:v0.0.10
```

- 3) 创建并启动容器，将代码挂载至该容器，注意terminal的工作路径

```
sy@sy-virtual-machine:~/Desktop/git/c_lists_sheny$ podman pull glcr.cra.ac.cn/operating-systems/project/kernel_legacy/ucore-tools:v0.0.10
Trying to pull glcr.cra.ac.cn/operating-systems/project/kernel_legacy/uore-tools:v0.0.10...
Getting image source signatures
Copying blob dc6f8a9ad3cb skipped: already exists
Copying blob 6a63d5478647 done
Copying config f75e6c5f26 done
Writing manifest to image destination
Storing signatures
f75e6c5f2666f86e435c143999e19d6abd3bf10a74b00486f5d486032b970a9c
sy@sy-virtual-machine:~/Desktop/git/c_lists_sheny$ podman run -v ./src
-it --name c_lab ucore-tools:v0.0.10
root@c38343630826:/# cd src
root@c38343630826:/src# ls
Makefile compile_flags.txt guix src tests
root@c38343630826:/src# make
```

- 4) 容器中执行指令 `make check` 可以对完成的代码进行测试

啥都没写的测试结果：

```
root@c38343630826:/src# make check
avocado run --job-results-dir test_results tests
JOB ID      : 40a9af69c3e5e0ef90a624750d870854ca905c2e
JOB LOG     : /src/test_results/job-2024-02-27T03.40-40a9af6/job.log
(2/2) tests/iotest.py:IOTest.test_simple_lists: STARTED
(1/2) tests/iotest.py:IOTest.test_rm_empty_lists: STARTED
(2/2) tests/iotest.py:IOTest.test_simple_lists: FAIL: 1 != 0 : Expectin
g a zero for the exit status (1.15 s)
(1/2) tests/iotest.py:IOTest.test_rm_empty_lists: PASS (3.13 s)
RESULTS     : PASS 1 | ERROR 0 | FAIL 1 | SKIP 0 | WARN 0 | INTERRUPT 0 |
CANCEL 0
JOB HTML    : /src/test_results/job-2024-02-27T03.40-40a9af6/results.html
JOB TIME    : 4.41 s

Test summary:
2-tests/iotest.py:IOTest.test_simple_lists: FAIL
make: *** [Makefile:77: check] Error 1
```

测试全部通过的结果:

```
root@cef714b5a50d:/src# make check
avocado run --job-results-dir test_results tests
JOB ID      : 37504fe7ac2d16c6fa5b5d357f140d4e6962d713
JOB LOG     : /src/test_results/job-2024-02-26T10.56-37504fe/job.log
(1/2) tests/iotest.py:IOTest.test_rm_empty_lists: STARTED
(2/2) tests/iotest.py:IOTest.test_simple_lists: STARTED
(1/2) tests/iotest.py:IOTest.test_rm_empty_lists: PASS (1.68 s)
(2/2) tests/iotest.py:IOTest.test_simple_lists: PASS (1.73 s)
RESULTS     : PASS 2 | ERROR 0 | FAIL 0 | SKIP 0 | WARN 0 | INTERRUPT 0 |
CANCEL 0
JOB HTML    : /src/test_results/job-2024-02-26T10.56-37504fe/results.html
JOB TIME    : 2.55 s
root@cef714b5a50d:/src#
```

5) 你可以选择在任何地方完成你的代码，但需要通过容器进行测试。建议测试通过后再提交到git，之后的作业我们会设置部分分，即通过部分样例可以得到部分分数。