

PA0 Report

1. 实验进度

我完成了所有的任务，包括：

1. 前置学习：

- vim
- makefile
- git

2. 按照手册要求编译项目

2. 遇见的问题

zsh错误

我使用的shell是zsh而非bash，而代码中是默认只会给 `~/.bashrc` 的末尾加上环境变量

```
export NEMU_HOME=/home/yama/ics2024/nemu
export AM_HOME=/home/yama/ics2024/abstract-machine
```

Fence 1

所以需要手动使用vim修改 `~/.zshrc`

make出错

```
~/ics2024/nemu pa0 ?1 > make 22:02:40
make -C tools/capstone
make[1]: Entering directory '/home/yama/ics2024/nemu/tools/capstone'
Cloning into 'repo'...
Connection closed by 198.18.0.63 port 22
fatal: could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
cd repo && CAPSTONE_ARCHS="x86|mips|riscv|loongarch" bash make.sh
/bin/sh: 1: cd: can't cd to repo
make[1]: *** [Makefile:23: repo/libcapstone.so.5] Error 2
make[1]: Leaving directory '/home/yama/ics2024/nemu/tools/capstone'
make: *** [src/utils/filelist.mk:23: tools/capstone/repo/libcapstone.so.5]
Error 2
~/ics2024/nemu pa0 ?1 > 22:03:35
```

Fence 2

出现了 `connection closed by 198.18.0.63 port 22` 的错误

发现198.18.0.63正是github的服务器ip，而port22是ssh连接所需要的端口，所以这波是使用ssh在连接clone code的时候出现了问题

```
~/ics2024/nemu pa0 ?1 > ssh -T git@github.com X HUP 22:09:06
Connection closed by 198.18.0.63 port 22
```

我记得我明明早就做好了这一步？

但是我重新将秘钥导入到之后，还是没用。

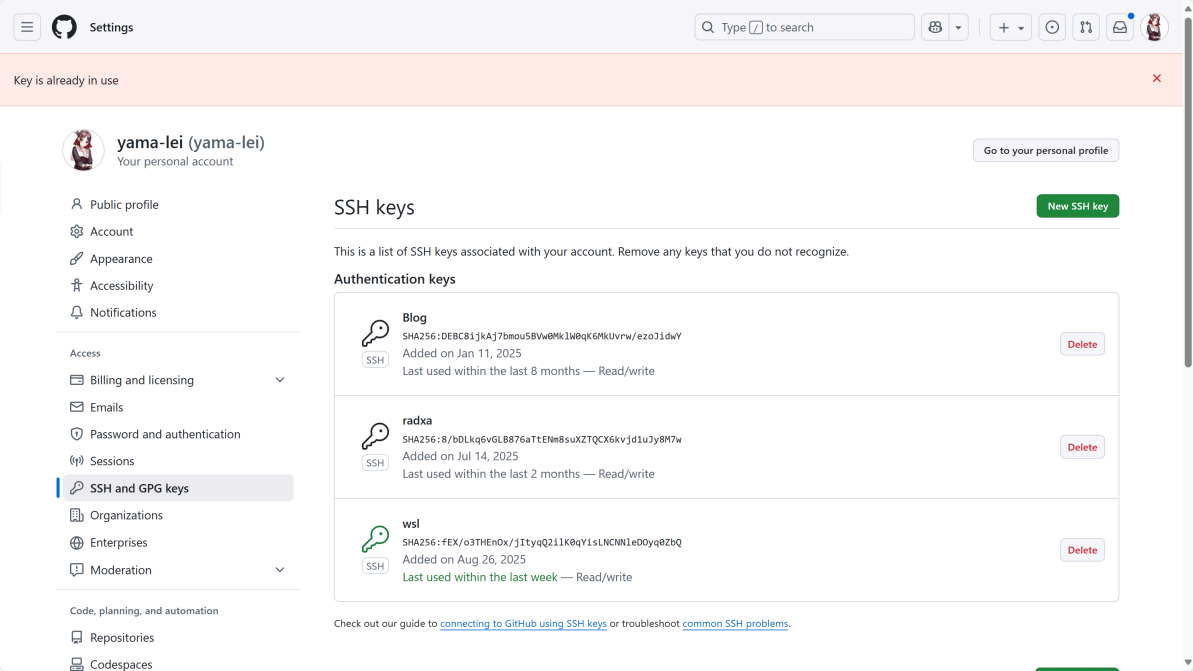


Figure 1

但是当我过了一段时间之后发现又可以了。猜测可能是网络环境的问题（比如代理等等）

git配置

项目要求使用自己的学号和名字作为git身份，但是我同时又想要使用 自己github的信息，所以只在 ics2024仓库下配置了学号和名字，global的git身份仍然是和github信息一致。

3. 必答题

学习gdb：

```
~/tmp > cat test.c

5s 22:09:03

int actual_calc(int a, int b){
    int c;
    c=a/b;
    return 0;
}

int calc(){
    int a;
    int b;
    a=13;
    b=0;
    actual_calc(a, b);
    return 0;
}

int main(){
```

```

    calc();
    return 0;
}
~/tmp > ls
22:09:07
test.c
~/tmp > gcc -ggdb test.c -o test.out
22:13:57
~/tmp > ./test.out
22:14:21
[1] 90785 floating point exception (core dumped) ./test.out
~/tmp >

```

Fence 4

- su认证失败是怎么回事?

开始我认为可能是密码输错了?

```

~ > su
11:18:27
Password:
su: Authentication failure

```

5s

Fence 5

然后搜索发现: "现在很多的Linux的桌面系统出于安全考虑, 在默认安装后, root用户默认是被锁定了的, 不允许登录, 也不允许 su 到 root, 对于桌面用户来说这个可能是为了增强安全性, 但是服务器版上确实有一大问题。"

然后按照教程, 成功解决问题:

```

~ > sudo passwd
11:18:41
[sudo] password for yama:
New password:
Retype new password:
passwd: password updated successfully
~ > su
11:20:00
Password:
root@LAPTOP-24A7CVC0: /home/yama#

```

7s

8s

Fence 6

- grep提示no such file or directory是什么意思?

当然是没有想要找的那个文件

- 请问怎么卸载Ubuntu?

如果是使用wsl的话, 在cmd中运行:

```

wsl --list
wsl --unregister Ubuntu-xxx # 换成要删除的系统

```

如果是使用虚拟机的话，可以删除虚拟机 Fence 7

如果是装了系统的话，重新刷一个别的系统就行了

- C语言的xxx语法是什么意思？

我怎么知道xxx语法是什么意思？

- ignoring return value of 'scanf'是什么意思？
- 出现curl: not found该怎么办？
- 为什么strtok返回NULL？
- 为什么会有Segmentation fault这个错误？
- 什么是busybox？

写到一半突然发现原来不是叫我回答问题.....这体现了RTFM的重要性！ 原来作业是写读后感

结合 [提问的智慧](#) 和Stop-Ask-Questions-The-Stupid-Ways之后，我对于提问的艺术有了更加深刻的把握。

在 [提问的智慧](#) 这一篇文章中作者讲到要在提问之前先写serach the fucking website看看是否可以从已有的互联网上搜索到自己问题的解决方案，这是很好解决问题的思路。但是现在ai时代，我认为还需要加上一句Ask the fucking model，问大模型往往也是一个很不错的解决方案，学会合理的使用ai也是每个人必不可少的技能。从这个角度看，手册和这两篇文章具有一定的滞后性，考虑到都是n年前写的，倒也是挺合理的。

好的提问，应该要能够将自己所做过的尝试、项目的环境等等背景信息都清楚地给到，让回答者能够充分地理解问题。一个好的问题，就像一个精心准备的舞台，所有必要的道具和背景都已就位，就等主角登场。你不能指望别人在你只给了一个模糊的舞台灯光下，就猜出你演的是哪出戏。你得告诉他，你用了什么语言，什么框架，遇到了什么报错信息，甚至最好能贴上你认为是核心问题的代码片段。这不是为了让别人帮你调试，而是为了让对方能快速定位到问题的本质。

下面这个就是群里一个同学发的**好的提问**，给出了清晰的说明和解释，帮助助教和老师定位问题

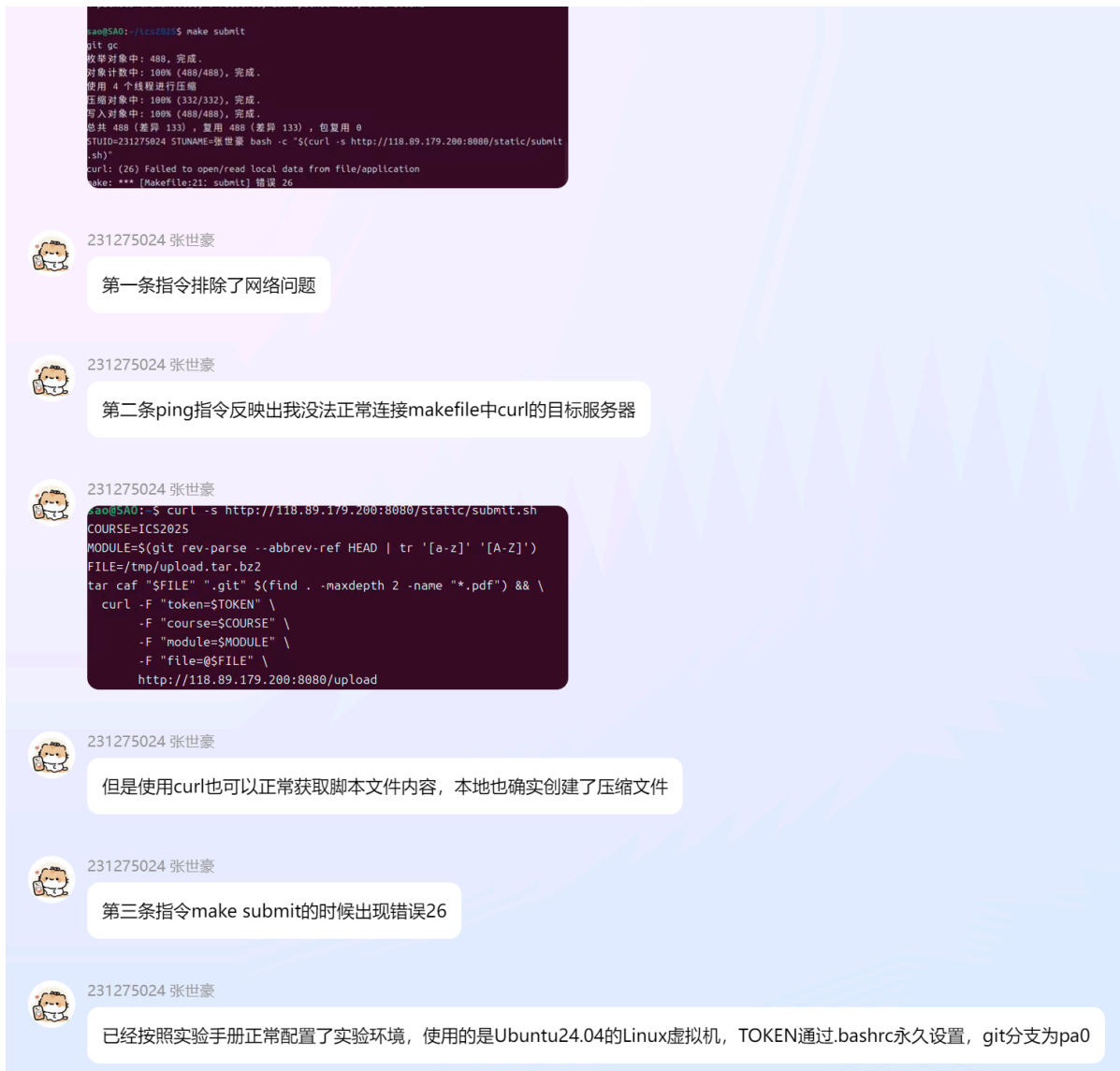


Figure 2

提问的艺术，归根结底是一种沟通的艺术。它要求你站在对方的角度去思考：我怎么才能让对方以最高效的方式理解我的困境？只给一个错误码，就像是我去食堂买放一样，我问阿姨有没有肉包，她说没有肉包，我指了指那个包子问她这个是什么，她说这个是鸡肉包。然后我说也许，接着阿姨就给我装了一个鸡肉包，但是我需要的打包好的肉包，于是还需要再次和阿姨讲清楚，实际上我们之间就存在着这样的不对称信息（没有对齐颗粒度），所有我们互相都误解了对方。。。

我很多时候在组织问题的过程中，就自己已经找到了答案。这种自问自答的过程，其实就是将一个混沌的思绪，通过逻辑的梳理，变得条理清晰。当你把自己的尝试、遇到的问题思考的过程都写下来时，你会发现，你对这个问题的理解已经上了一个台阶。就算最终还是要求助，你的问题也会变得更高质量，更值得被认真回答。因为你不仅是在寻求帮助，更是在展示你的思考能力和解决问题的态度。简单粗暴的问题，通常只能得到简单粗暴的答案s甚至得不到任何答案。

在做PA0的时候，我阅读了很多的文档（主要是PA的官方文档，和一些linux手册说明），来理解linux的一些基本指令、基本操作等等，在不断地阅读文档的过程中，，我渐渐地理解了阅读文档的重要性，通过阅读文档，按照文档的步骤和规范来进行操作，能够让我们更加深入地理解项目的细节，更加顺利地完成任务。

但是RTFM还远远不够解决所有的问题，很多东西都需要搜索引擎和ai大模型的辅助。

比如在按照文档的要求，执行某一个脚本的时候（好像是init.sh还是啥来着），默认的行为是在~/.bashrc中添加环境变量，但是我使用的shell不是bash，而是zsh,所以需要把这里环境变量添加到~/.zshrc中，才能够正常地使用。

