

1 计算环境

小组通过内网穿透软件 frp，向每个成员呈现了以下计算设备，成员可以通过 ssh 远程访问这些设备。这种 frp+ssh 的方式和传统的 ssh 的使用方式有一些差别，细节参 sharedKnowledge/tools 文档的“frp/访问者的 ssh 访问命令”部分。

为了上述目的，以及部署 git 服务，我们租用了华为的云服务器，其公网 IP 是 124.70.32.23，以下文档多处会出现这个 IP 地址。

1.1 树莓派

这个设备是一个低功耗 Linux 系统，主要被用来唤醒以下的 sw 机器，或者可以被用来传输大文件。

```
ssh -oPort=8000 pi@124.70.32.23
```

密码为：resourceGate

唤醒以下的 sw 机器的方法：执行/home/pi/swon 脚本即可。

由于这只是一个低配置的机器，所以用户不要在这个机器上执行任何计算量大的程序。

文件传输。用户可以使用该机器的/home/pi/tem 目录作为中转，向其他成员传输一些大文件。例如，用户可以使用下面的命令将自己机器的/home/user/test.txt 传输到树莓派机器的上述目录：

```
scp -P 8000 /home/user/test.txt pi@124.70.32.23/home/pi/tem
```

如果想要从上述目录复制一个文件到自己本地机器，可以使用以下命令：

```
scp -P 8000 pi@124.70.32.23/home/pi/tem/test.txt /home/user
```

1.2 sw 机器

设备简介。这个设备的主要作用：

1. 存放小组成员都可以访问的数据集，部署项目开发中的可执行程序，多个成员协作完成的实验（例如复现无人车避障算法中的实验）。这样，成员能够在同一个机器看到其他成员的工作、实验数据，便于成员协作。该设备有 16T 的存储容量，计算速度较快。

2. 配备有一块 GPU（NVIDIA GeForce RTX 3080），可以被用来做较长时间深度学习训练。

不要存放重要数据。由于这个机器主要的目的是做实验用，可能由于环境配置等原因导致宕机，所以不要在这个机器上存放重要的个人数据。

节能考虑。由于该设备运行时峰值功率较高，为了节省能耗，用户应该以下面的方式使用该设备。该设备平时处于 suspended 状态，用户登录树莓派机器后，可以唤醒该机器。然而通过树莓派登录该机器，或者直接通过 frp/ssh 方式登录该机器。用户使用完毕该机器后，运行/usr/bin/off 脚本可以令该机器重新进入到 suspended 状态。sw 机器在内网中的 IP 地址：192.168.1.102。

登录命令。为每个成员创建了账户，账户的名字是各位姓名的简写，也就是：cjw cyr gx wyf zry zb。密码和账户名字一致。以用户 zb 为例，登录命令为：

```
ssh -oPort=8010 zb@124.70.32.23
```

每个成员应该将自己的数据存放在自己账户对应的目录下（例如/home/zb），除非大家约定将实验数据放在一个工作的系统目录（例如/sw/cv/exp）。

1.3 git 管理的项目

将设备的 ssh key 传输给管理员。小组将需要共享的知识以及协同研发的项目组成若干 git 项目。为了访问这些项目内容，一个成员首先需要将自己所用的计算设备的 ssh key 传输给管理员（张波），在管理员做了一些管理方面的工作后，会通知该成员可以访问某个 git 项目了。一个成员如果需要在多个计算设备上访问这样的 git 项目，需要将每个设备的 ssh key 传输给管理员。

以下是一个成员在一个设备上生成 ssh key 的步骤：在客户端执行以下命令，

```
ssh-keygen -t rsa
```

当该程序提供用户输入一些信息时，都按回车以采用该程序提供的默认值，包括提示“Enter passphrase”。程序运行结束时会在 ~/.ssh 中创建文件 id_rsa 以及 id_rsa.pub。将那个 id_rsa.pub 复制到一个临时目录，并将名字更改为一个能够标识该设备的名字，比如 zb_linuxNotebook.pub，然后传输给管理员。

小组 git 项目介绍。目前本小组所有成员都可以访问的 git 项目如下：

1. sharedKnowledge。拉取的命令为：

```
git clone git@124.70.32.23:sharedKnowledge
```

这个项目具有以下子目录：

cmake。存放 cmake 用法知识。

cpplibs。存放常用的 C++ 库的用法。

git。存放 git 用法方面的信息。

groupCooperation。存放小组协作时的注意事项，比如代码规范等。

python。存放 python 语言以及开发的知识。

tools。存放常用工具（比如文档写作、绘图、frp 用法）等信息。

web。存放 web 开发的相关知识。

2. obstacleAvoidance。拉取的命令为：

```
git clone git@124.70.32.23:obstacleAvoidance
```

这个项目存放避障算法研发相关的文档、源代码。

3. testing。拉取的命令为：

```
git clone git@124.70.32.23:testing
```

这是一个测试项目，用于联系 git 的用法。