

转移部署教程

1. 安装Anaconda

安装包

由于是aarch64架构，所以官方没有提供编译好的conda，要从其他渠道安装。

移动U盘中的Archiconda3-0.2.3-Linux-aarch64.sh安装包到当前目录（如果不清楚当前目录在哪，可输入pwd查看）

安装

输入

```
./Archiconda3-0.2.3-Linux-aarch64.sh
```

即可执行安装命令

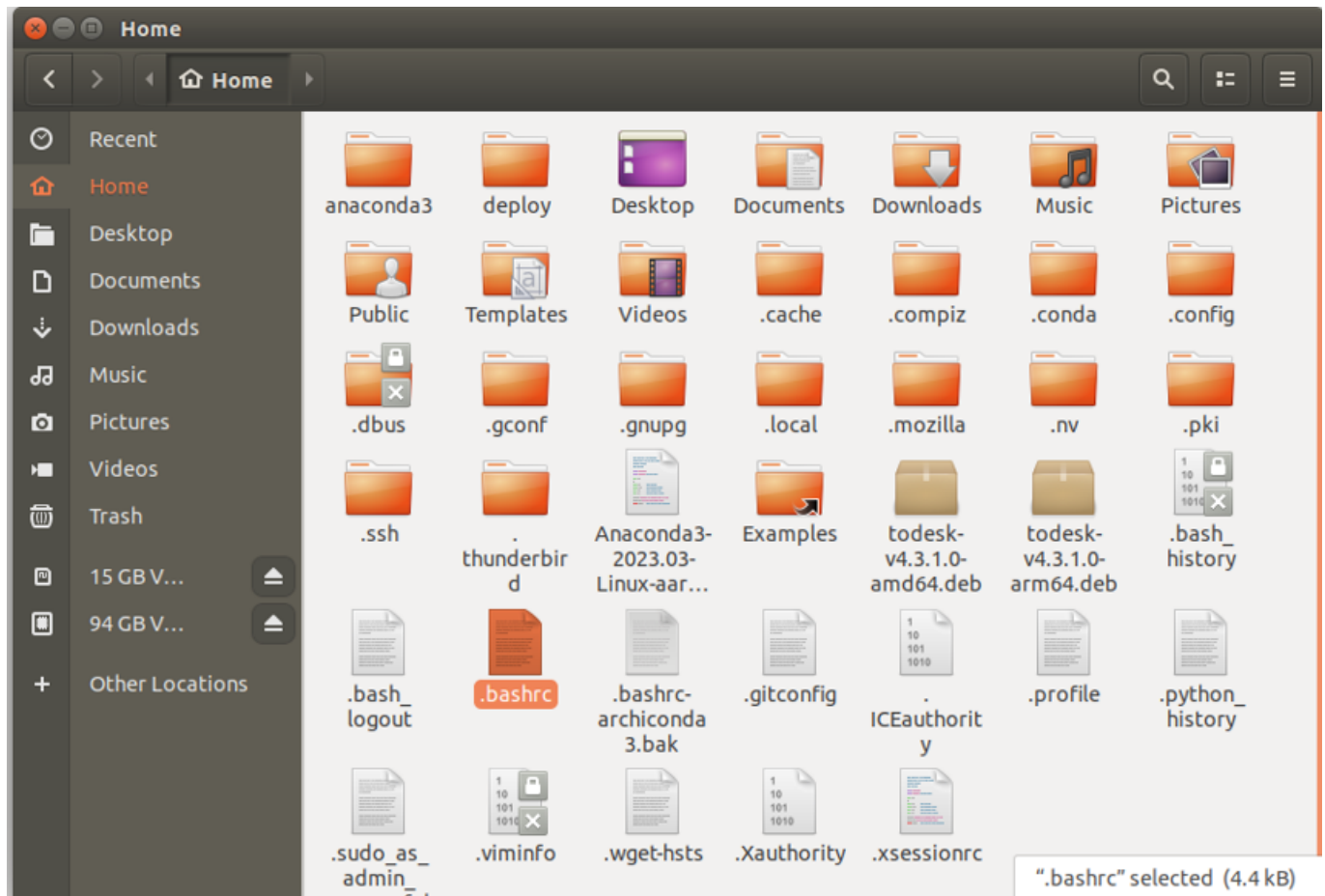
路径配置

conda应该会被安装至/opt/conda下面，将其全部文件迁移到/home/nvidia/.conda下

若当前目录在home，使用下列命令来新建文件夹并修改权限：

```
sudo mkdir nvidia  
cd nvidia  
sudo mkdir .conda  
cd ..  
chmod -R 777 nvidia
```

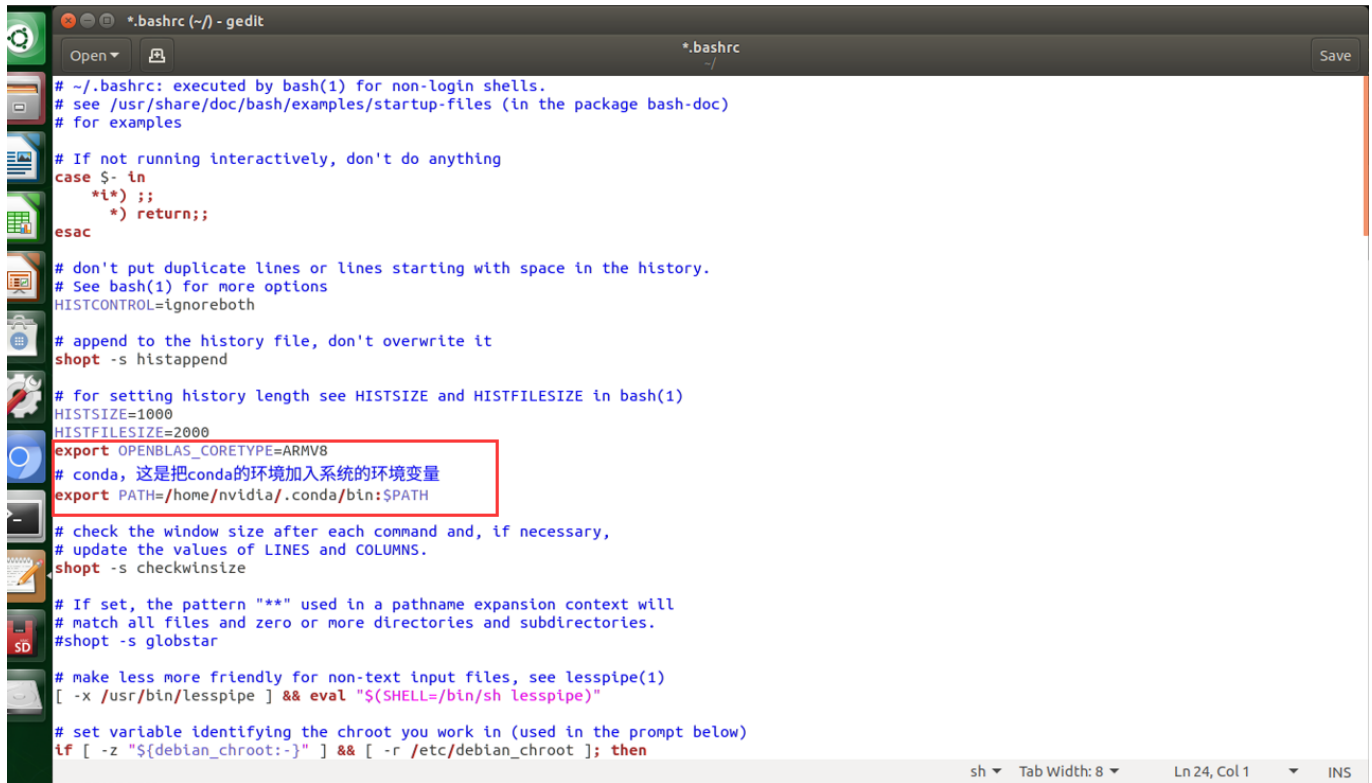
然后分别打开/home/jetson(这里是用户名，自行修改)/.bashrc和/etc/profile，如果没有权限就使用vim进行编辑



```
sudo vim xxx(文件路径)
```

打开编辑，按a进入输入状态，按:wq保存并退出

在文件里加入如下几行



```
#!/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
export OPENBLAS_CORETYPE=ARMv8
# conda, 这是把conda的环境加入系统的环境变量
export PATH=/home/nvidia/.conda/bin:$PATH

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "*" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
```

或选择gedit也可

```
sudo gedit xxx(文件路径)
```

注意检查路径是否和当前的**conda**位置符合

注意不要加在文件末尾，可以找文件中有相似格式的位置添加。

然后执行

```
source /etc/profile
```

来应用更改

测试

输入**conda -V**查看是否配置成功

2.配置环境

移动文件

将U盘里的文件先转移到任意位置

将U盘里的**envs**文件夹下的**py38**文件夹移动到**/home/nvidia/.conda/envs**下

显式声明环境变量

打开前文提到的**.bashrc**

在刚刚添加的语句后，新添一行

```
export LD_PRELOAD=$LD_PRELOAD:/home/nvidia/conda/envs/py38/lib/python3.8/site-packages/scikit_learn.libs/libgomp-d22c30c5.so.1.0.0
```

3.运行程序

在控制台里面使用**cd**命令转移到U盘里**deploy**所转移到的的文件夹并**cd deploy** (即当前控制台的最右边显示是'deploy')

或直接打开**deploy**文件夹再右键空白处点击**open terminal**

在执行过

```
conda activate py38
```

的情况下，直接输入

```
python new0612.py
```

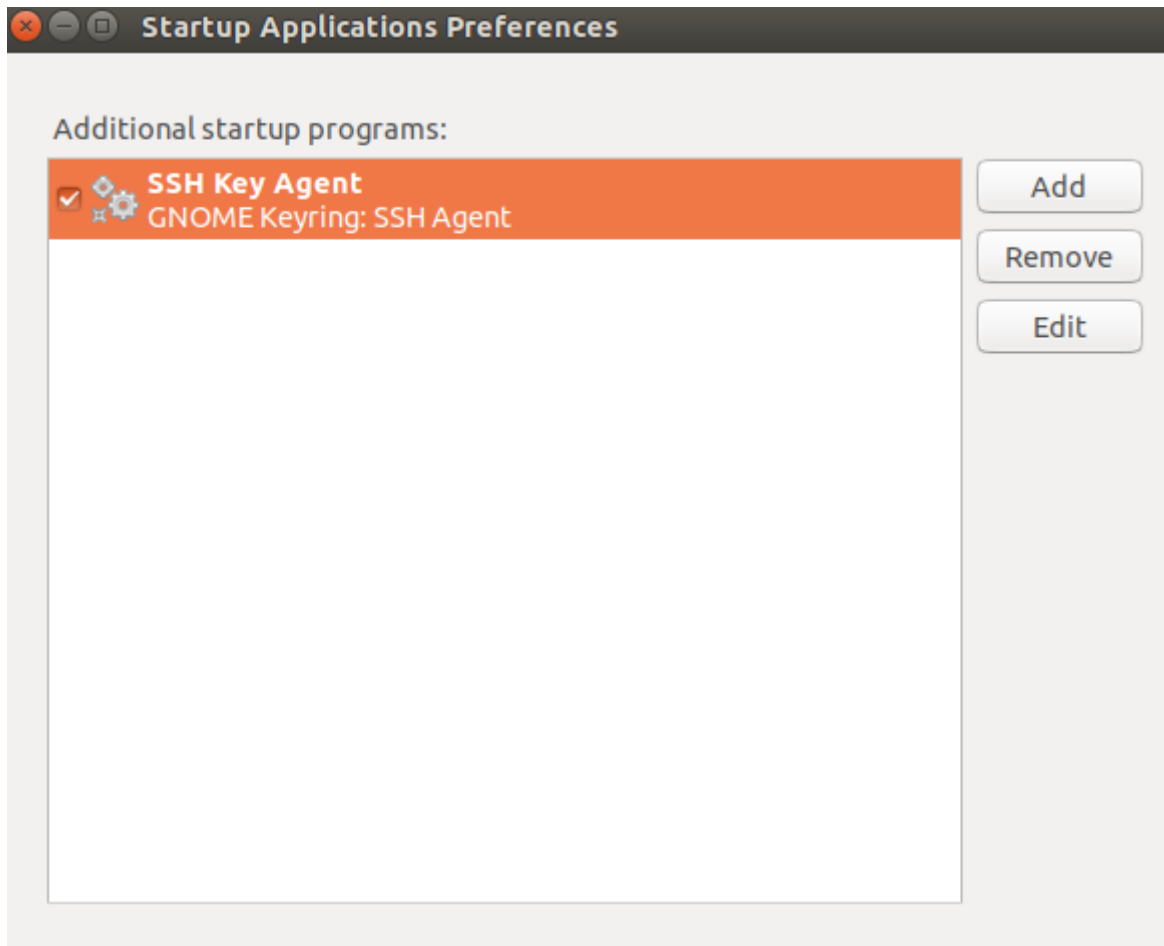
即可运行程序

4. 设置开机自启动

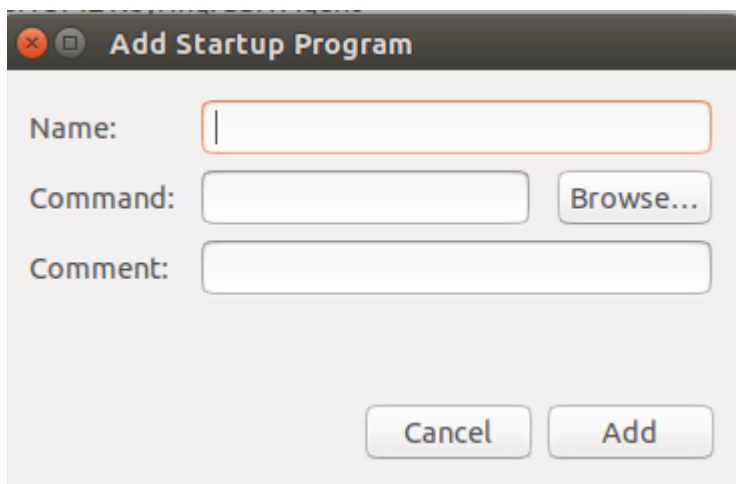
在终端输入

```
gnome-session-properties
```

或直接在搜索栏输入**start**即可打开**Startup Applications**



然后点击Add即可添加开机时自动运行的指令了



Name随便填，Comment可不填，在Command栏填写`gnome-terminal`

然后打开前文提到的.bashrc文件，在文件的末尾输入以下语句：

```
conda activate py38
sleep 2
cd deploy
python new0612.py
```

5.备份与还原

tar命令

1. 备份系统

我该如何备份我的Ubuntu系统呢？很简单，就像你备份或压缩其它东西一样，使用TAR。和Windows不同，Linux不会限制root访问任何东西，你可以把分区上的所有东西都扔到一个TAR文件里去！

首先成为root用户：

```
sudo su
```

然后，进入根目录后，使用下面的命令备份你的系统：

```
tar cvpzf backup.tgz --exclude=/proc --exclude=/lost+found --exclude=/backup.tgz -  
-exclude=/mnt --exclude=/sys /
```

让我们来简单看一下这个命令：

tar当然就是我们备份系统所使用的程序了。

cvpzf是**tar**的选项，意思是“创建档案文件”、“保持权限”(保留所有东西原来的权限)、“使用**gzip**来减小文件尺寸”。

backup.gz是我们将要得到的档案文件的文件名。

/是我们要备份的目录，在这里是整个文件系统。

在 档案文件名“**backup.gz**”和要备份的目录名“**/**”之间给出了备份时必须排除在外的目录。有些目录是无用的，例如“**/proc**”、“**/lost+ found**”、“**/sys**”。当然，“**backup.gz**”这个档案文件本身必须排除在外，否则你可能会得到一些超出常理的结果。如果不把“**/mnt**”排 除在外，那么挂载在“**/mnt**”上的其它分区也会被备份。另外需要确认一下“**/media**”上没有挂载任何东西(例如光盘、移动硬盘)，如果有挂载东西， 必须把“**/media**”也排除在外。

你还可以用**Bzip2**来压缩文件，**Bzip2**比**gzip**的压缩率高，但是速度慢一些。如果压缩率对你来说很重要，那么你应该使用**Bzip2**，用“**j**”代替命令中的“**z**”，并且给档案文件一个正确的扩展名“**bz2**”。完整的命令如下：

```
tar cvpjf backup.tar.bz2 --exclude=/proc --exclude=/lost+found --  
exclude=/backup.tar.bz2 --exclude=/mnt --exclude=/sys /
```

2. 恢复系统

在进行恢复系统的操作时一定要小心！如果你不清楚自己在做什么，那么你可能把重要的数据弄丢，请务必小心！

接着上面的例子。切换到root用户，并把文件“**backup.tgz**”拷贝到分区的根目录下。

在 Linux中有一件很美妙的事情，就是你可以在一个运行的系统中恢复系统，而不需要用**boot-cd**来专门引导。当然，如果你的系统已经挂掉不能启动了， 你可以用**Live CD**来启动，效果是一样的。

使用下面的命令来恢复系统：

```
tar xvpfz backup.tgz -C /
```

如果你的档案文件是使用Bzip2压缩的，应该用：

```
tar xvpfj backup.tar.bz2 -C /
```

注意：上面的命令会用档案文件中的文件覆盖分区上的所有文件。

执行恢复命令之前请再确认一下你所键入的命令是不是你想要的，执行恢复命令可能需要一段不短的时间。

恢复命令结束时，你的工作还没完成，别忘了重新创建那些在备份时被排除在外的目录：

```
mkdir proc
mkdir lost+found
mkdir mnt
mkdir sys
```

等等

当你重启电脑，你会发现一切东西恢复到你创建备份时的样子了

dd命令

1. 磁盘克隆 也就是把整个硬盘复制一份。当然你首先需要在计算机上在接上一块新硬盘，并让系统识别。例如这块硬盘可能被识别为/dev/sdb,原有硬盘叫/dev/sda. 然后你可以在linux命令行上简单地执行：
`dd if=/dev/sda of=/dev/sdb` 对就这么简单，此命令完成后，你的第二块硬盘上将有一个和第一块硬盘一模一样的副本，也是可以启动的。因为dd操作就是简单的按字节复制，什么分区表啊，MBR啊统统照搬。当然你也可以做一些微调，来定制一下克隆操作。例如你可能希望把硬盘上的内容全部备份到一个磁盘文件中，而不是另一块新硬盘，你可以 `dd if=/dev/sda of=~/.disk1.img` 之后，disk1.img就是备份好的磁盘映像文件，你可以复制或转移到其他介质也可以压缩一下：`gzip disk1.img`
`#generates disk1.img.gz` 或者在创建磁盘映像文件的同时就执行压缩：`dd if=/dev/sda | gzip > disk1.img.gz` 当然，你也可以使用bzip2来压缩，它的压缩率更高，但是速度慢一些：`dd if=/dev/sda | bzip2 > disk1.img.bz2`
2. 分区克隆 如果我们不想克隆整块磁盘，只想把某些存放数据的分区备份一下，你会需要克隆一个分区。正如预备知识所述，分区在linux下也是一个文件。例如 `dd if=/dev/sda1 of=~/.part1.img` 将把第一块磁盘的第一个分区备份到名为part1.img的文件中。也可以进行分区复制：`dd if=/dev/hda1 of=/dev/hda3` 将把分区一原样复制到分区3.
3. 从镜像文件恢复磁盘或分区 很简单，把上面命令中，of和if指向的文件名互换一下即可实现恢复。例如 `dd if=disk1.img of=/dev/sda` 把保存在disk1.img中的映像恢复到第一块磁盘 `dd if=part1.img of=/dev/hda2` 把保存在part1.img中的映像恢复到第一块硬盘的第二个分区

最后，需要注意一点是，无论备份还是恢复，都需要保证 `of` 指向的文件或磁盘空间要大于 `if` 指向的内容，因为是按字节复制 如果没有足够的空间去存储内容，备份自然会失败。

6.配置文件参数说明

配置文件说明：
本程序部分参数使用配置文件进行配置，配置文件`para.json`位于`deploy`文件夹内，具体的参数定义如下：

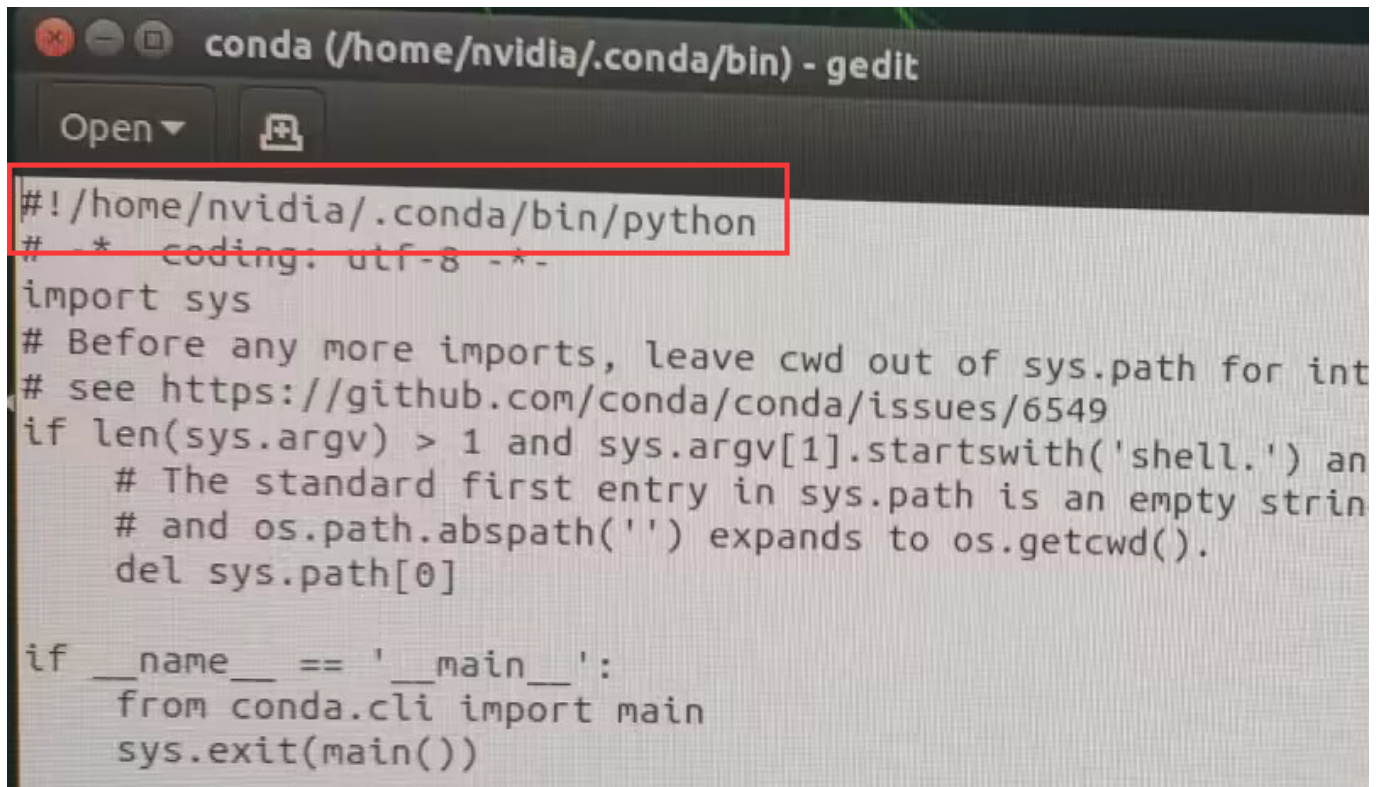
序号	参数名	含义说明
1	OPCAdd	PLC的opc地址
2	CamAdd	摄像头连接地址
3	lifebeat_to_etr	PLC端心跳包node地址
4	lifebeat_from_etr	ETRS端心跳包node地址
5	tray_in_pos	空框到位node地址
6	tray_empty	框体是否空node地址
7	high_top	检测画幅高度上
8	high_bottom	检测画幅高度下
9	width_left	检测画幅宽度左
10	width_right	检测画幅宽度右

可能遇到的问题

- Q：

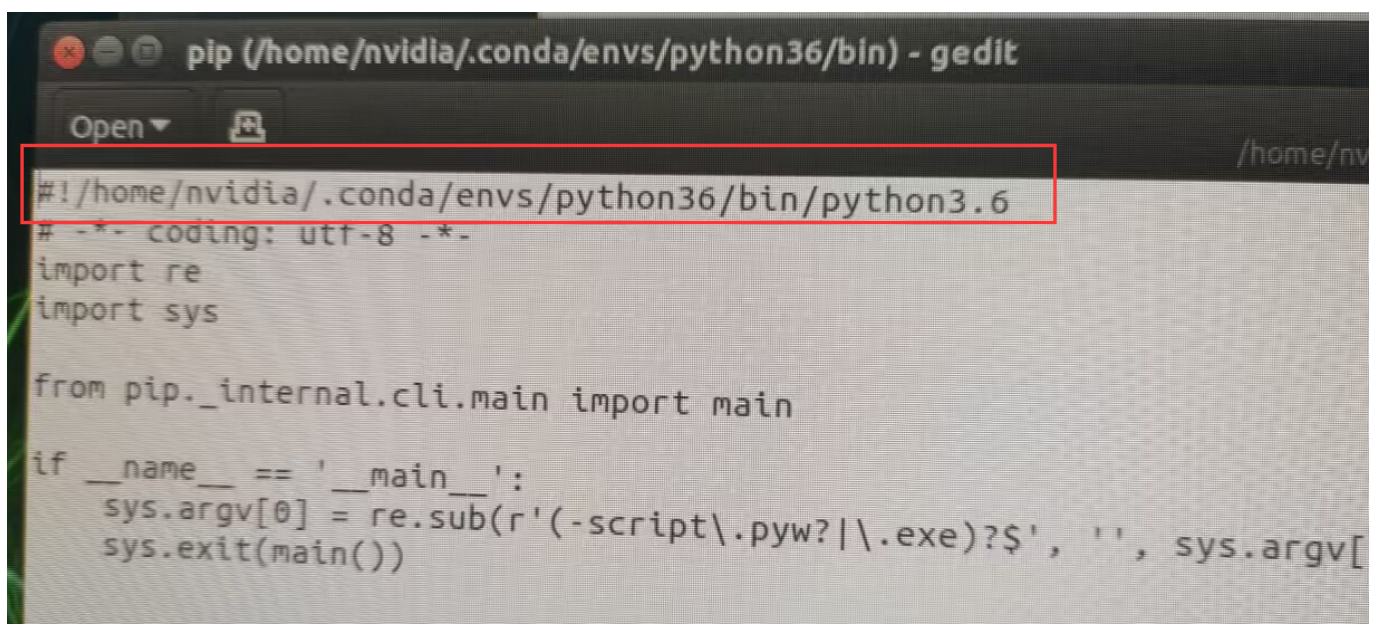
使用`pip`或者`conda`命令时出现`bad interpreter: No such file or directory:`的错误
- A：

打开`python36`和`conda`目录下的`bin`文件夹，查看`pip`文件和`conda`文件的第一行`#!`后面的解释器路径配置是否正确



```
#!/home/nvidia/.conda/bin/python
# -*- coding: utf-8 -*-
import sys
# Before any more imports, leave cwd out of sys.path for int
# see https://github.com/conda/conda/issues/6549
if len(sys.argv) > 1 and sys.argv[1].startswith('shell.') and
    # The standard first entry in sys.path is an empty string
    # and os.path.abspath('.') expands to os.getcwd().
    del sys.path[0]

if __name__ == '__main__':
    from conda.cli import main
    sys.exit(main())
```



```
#!/home/nvidia/.conda/envs/python36/bin/python3.6
# -*- coding: utf-8 -*-
import re
import sys

from pip._internal.cli.main import main

if __name__ == '__main__':
    sys.argv[0] = re.sub(r'(-script|.pyw?|\.exe)?$', '', sys.argv[0])
    sys.exit(main())
```

(此处以python3.6示例，根据实际需求自行修改)

若与当前目录不符，修改即可。若没权限(文件只读)则可使用`sudo vim`命令编辑文件