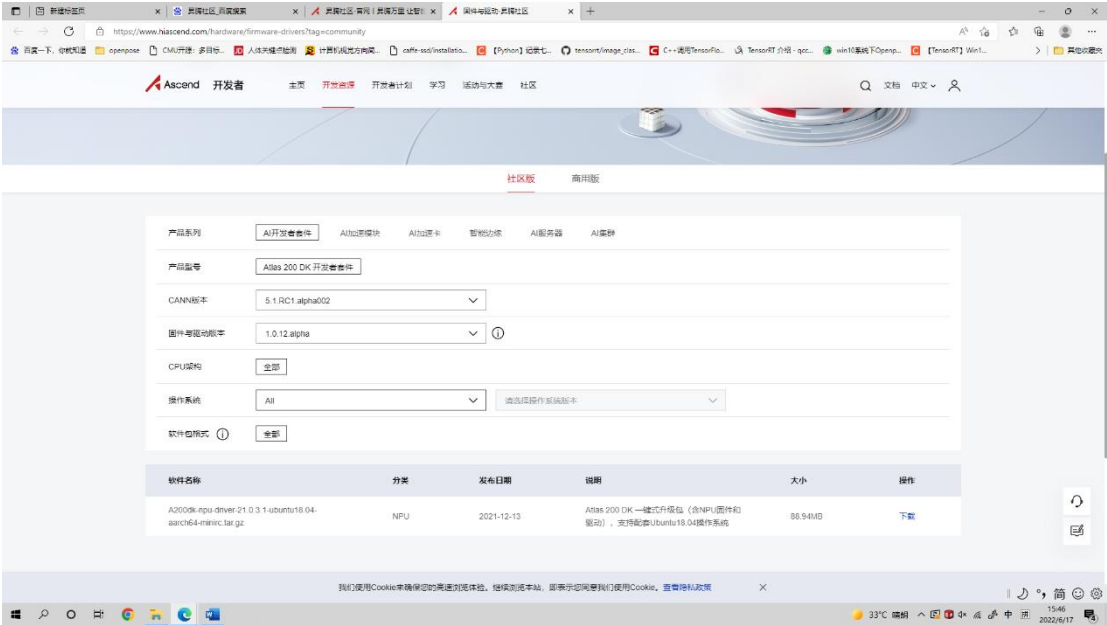


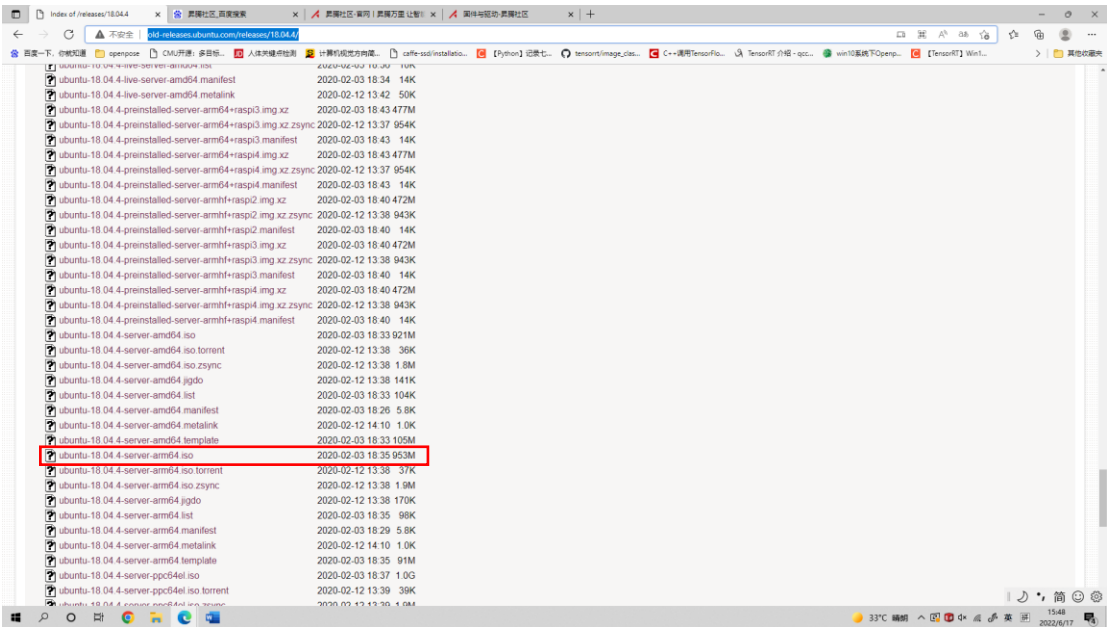
# 制卡环节

软件包准备 获取开发者板驱动与运行包、Ubuntu 操作系统镜像  
获取对应 CANN 版本驱动



Ubuntu 操作系统镜像选择对应版本为 18.04.4

链接：[ubuntu-18.04.4-server-arm64.iso](https://releases.ubuntu.com/18.04.4/)



将 SD 卡放入读卡器，并将读卡器与 Ubuntu 服务器的 USB 接口连接

Ubuntu 服务器中安装 qemu-user-static、binfmt-support、yaml、squashfs-tools 与交叉编译器

# 进入 root 用户

sudo su

# 更新源

apt-get update

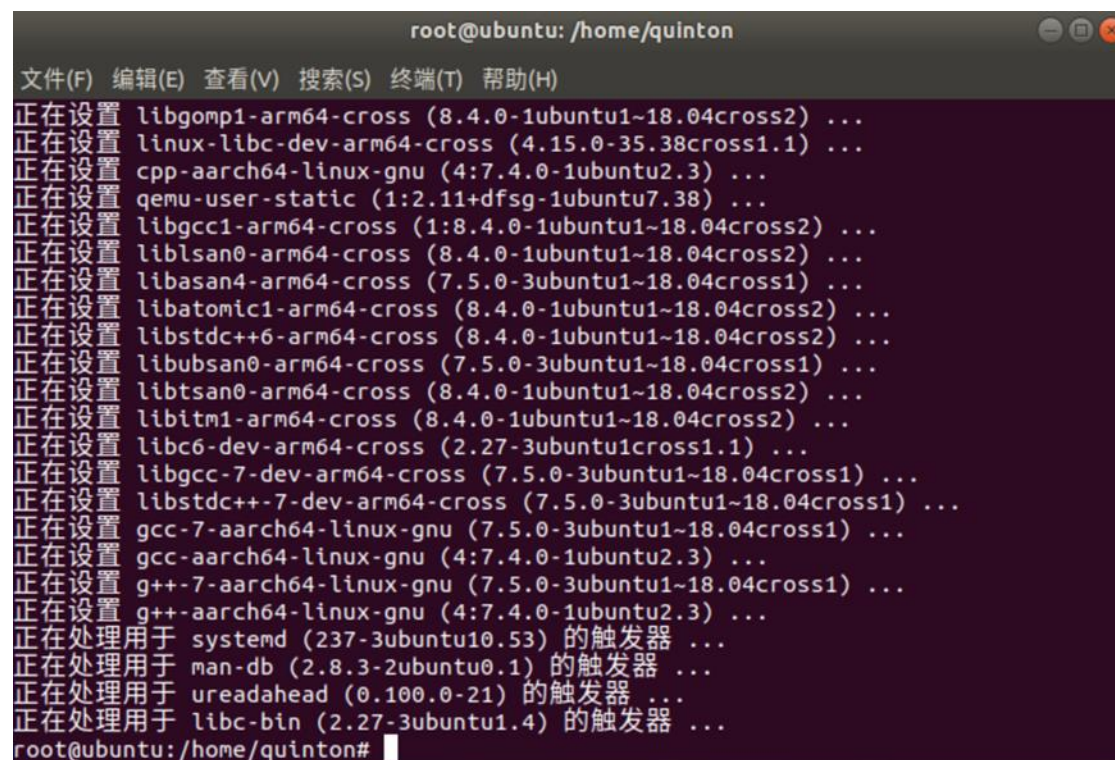
# 安装 python 相关依赖

pip3 install pyyaml

# 安装 qemu-user-static、binfmt-support、yaml、squashfs-tools 与交叉编译器

apt-get install qemu-user-static binfmt-support python3-yaml squashfs-tools gcc-aarch64-linux-gnu g++-aarch64-linux-gnu

安装完成



```
root@ubuntu: /home/quinton
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
正在设置 libgomp1-arm64-cross (8.4.0-1ubuntu1~18.04cross2) ...
正在设置 linux-libc-dev-arm64-cross (4.15.0-35.38cross1.1) ...
正在设置 cpp-aarch64-linux-gnu (4:7.4.0-1ubuntu2.3) ...
正在设置 qemu-user-static (1:2.11+dfsg-1ubuntu7.38) ...
正在设置 libgcc1-arm64-cross (1:8.4.0-1ubuntu1~18.04cross2) ...
正在设置 liblsan0-arm64-cross (8.4.0-1ubuntu1~18.04cross2) ...
正在设置 libasan4-arm64-cross (7.5.0-3ubuntu1~18.04cross1) ...
正在设置 libatomic1-arm64-cross (8.4.0-1ubuntu1~18.04cross2) ...
正在设置 libstdc++6-arm64-cross (8.4.0-1ubuntu1~18.04cross2) ...
正在设置 libubsan0-arm64-cross (7.5.0-3ubuntu1~18.04cross1) ...
正在设置 libtsan0-arm64-cross (8.4.0-1ubuntu1~18.04cross2) ...
正在设置 libitm1-arm64-cross (8.4.0-1ubuntu1~18.04cross2) ...
正在设置 libc6-dev-arm64-cross (2.27-3ubuntu1cross1.1) ...
正在设置 libgcc-7-dev-arm64-cross (7.5.0-3ubuntu1~18.04cross1) ...
正在设置 libstdc++-7-dev-arm64-cross (7.5.0-3ubuntu1~18.04cross1) ...
正在设置 gcc-7-aarch64-linux-gnu (7.5.0-3ubuntu1~18.04cross1) ...
正在设置 gcc-aarch64-linux-gnu (4:7.4.0-1ubuntu2.3) ...
正在设置 g++-7-aarch64-linux-gnu (7.5.0-3ubuntu1~18.04cross1) ...
正在设置 g++-aarch64-linux-gnu (4:7.4.0-1ubuntu2.3) ...
正在处理用于 systemd (237-3ubuntu10.53) 的触发器 ...
正在处理用于 man-db (2.8.3-2ubuntu0.1) 的触发器 ...
正在处理用于 ureadahead (0.100.0-21) 的触发器 ...
正在处理用于 libc-bin (2.27-3ubuntu1.4) 的触发器 ...
root@ubuntu: /home/quinton#
```

创建制卡工作目录

mkdir /home/ascend/mksd

# 制卡目录可任意指定

#ascend 为自己的用户名

将软件包准备获取的 Ubuntu 操作系统镜像包、开发者板所有驱动包上传到制卡工作目录  
在制卡工作目录下获取制卡脚本。可点击链接直接下载或在 Ubuntu 内输入命令下载

[make\\_sd\\_card.py](#)

[make\\_ubuntu\\_sd.sh](#)

# 下载制卡入口脚本“make\_sd\_card.py”。

# 从 gitee 下载：

```
wget https://gitee.com/ascend/tools/raw/master/makesd/generic_script/make_sd_card.py
```

# 从 github 下载：

```
wget
```

```
https://raw.githubusercontent.com/Ascend/tools/master/makesd/generic_script/make_sd_card.py
```

# 下载制作 SD 卡操作系统的脚本“make\_ubuntu\_sd.sh”。

# 从 gitee 下载：

```
wget https://gitee.com/ascend/tools/raw/master/makesd/generic_script/make_ubuntu_sd.sh
```

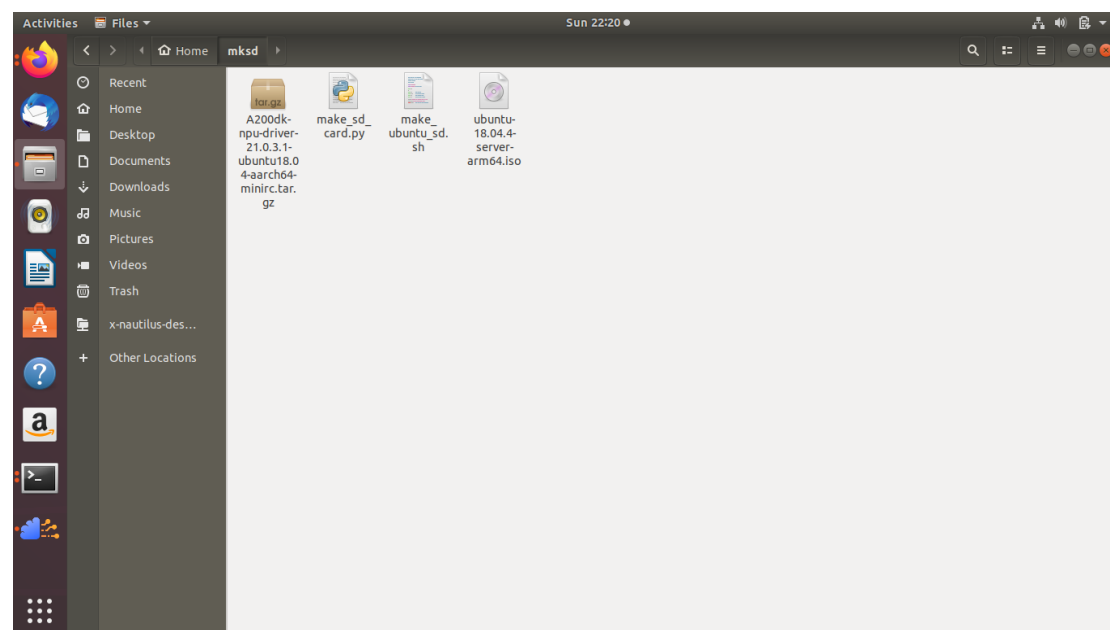
# 从 github 下载：

```
wget
```

```
https://raw.githubusercontent.com/Ascend/tools/master/makesd/generic_script/make_ubuntu_sd.sh
```

下载完成

mksd 文件夹内容如下：



执行制卡脚本。

以 root 用户执行如下命令查找 SD 卡所在的 USB 设备名称。

```
fdisk -l
```

例如，SD 卡所在 USB 设备名称为“/dev/sda”，可通过插拔 SD 卡或者按容量推断的方式确定设备名称。

图中的 sd 卡为/dev/sdb

运行 SD 制卡脚本“make\_sd\_card.py”。

```
python3 make_sd_card.py local /dev/sda1
```

“local”表示使用本地方式制作 SD 卡。

“/dev/sda”为 SD 卡所在的 USB 设备名称。

正在制卡

制卡完成

拔出 sd 卡，插入开发板并上电，六个 led 灯亮起则成功启动开发板