

Ubuntu20.04 快速重装手册

V1.1

目录

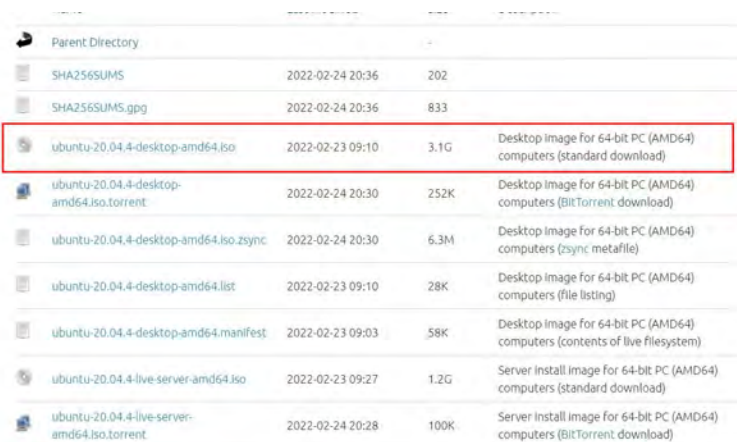
1	软件下载	1
1.1	ubuntu20.04	1
1.2	ultralso	2
1.3	diskgenius	2
2	Ultralso 制作启动盘	3
3	Bios 设置	4
4	Ubuntu 安装	5
5	Ubuntu 设置	12
5.1	ubuntu 换源	12
5.2	中文输入法	12
5.2.1	中文输入法	12
5.2.2	搜狗输入法	13
5.3	ubuntu 与 win10 时间不同步	13
5.4	ubuntu flash 插件	14
5.5	ubuntu 启动黑屏问题	14
6	ROS 安装	15
7	Opencv 4.5.5 安装	16
7.1	下载	16
7.2	安装依赖库	16
7.3	编译 OpenCV	17
7.3.1	debug 模式	17
7.3.2	release 模式	17
7.4	安装 OpenCV	17
7.5	设置库的搜索路径	17
7.6	添加 pkg-config 环境变量	18
7.6.1	debug	18
7.6.2	release	18
7.6.3	路径设置在环境变量中	18
7.7	验证安装	19
8	cuda11.5 安装	20
9	cudnn8.3.3 安装	20
10	torch 安装	21

1 软件下载

1.1 ubuntu20.04

Ubuntu20.04 安装包官网 <https://ubuntu.com/>
方法一

Ubuntu20.04 下载页 <https://releases.ubuntu.com/20.04/>



File Name	Size	Download Link
Parent Directory	-	-
SHA256SUMS	202	SHA256SUMS
SHA256SUMS.gpg	833	SHA256SUMS.gpg
ubuntu-20.04.4-desktop-amd64.iso	3.1G	ubuntu-20.04.4-desktop-amd64.iso
ubuntu-20.04.4-desktop-amd64.iso.torrent	252K	ubuntu-20.04.4-desktop-amd64.iso.torrent
ubuntu-20.04.4-desktop-amd64.iso.zsync	6.3M	ubuntu-20.04.4-desktop-amd64.iso.zsync
ubuntu-20.04.4-desktop-amd64.list	28K	ubuntu-20.04.4-desktop-amd64.list
ubuntu-20.04.4-desktop-amd64.manifest	58K	ubuntu-20.04.4-desktop-amd64.manifest
ubuntu-20.04.4-live-server-amd64.iso	1.2G	ubuntu-20.04.4-live-server-amd64.iso
ubuntu-20.04.4-live-server-amd64.iso.torrent	100K	ubuntu-20.04.4-live-server-amd64.iso.torrent

方法二

Ubuntu20.04 下载链接 <https://releases.ubuntu.com/20.04/ubuntu-20.04.4-desktop-amd64.iso>

1.2 ultraiso

用于 ubuntu 系统的 u 盘启动制作

Ultraiso 官网 <https://cn.ultraiso.net/>



1.3 diskgenius

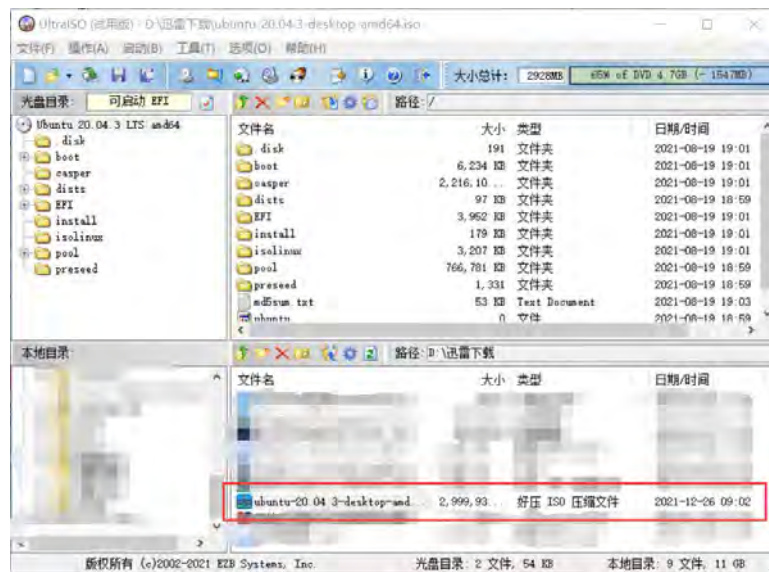
如果 ubuntu 系统无法启动并且忘记备份数据, diskgenius 用于重装之前, 在 win10 中备份 ubuntu 原来的数据



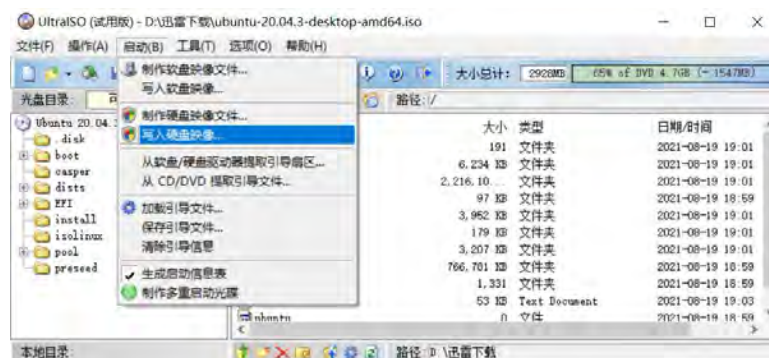
当制作好启动盘之后, 在重新启动之前需要先进行空间的压缩, 一般压缩100G就可以, 具体的步骤见:

2 Ultraso 制作启动盘

step 1 点击文件-> 打开选择下载好的 ubuntu20.04 iso 文件
或者直接在本地目录内进入到 iso 文件位置, 然后双击



step 2 再点击启动-> 写入硬盘镜像选择插入的 U 盘





最后出现刻录成功即可

3 Bios 设置

step 1 进入电脑 BIOS

(每个电脑进入 BIOS 方式不同, 请自行查询)

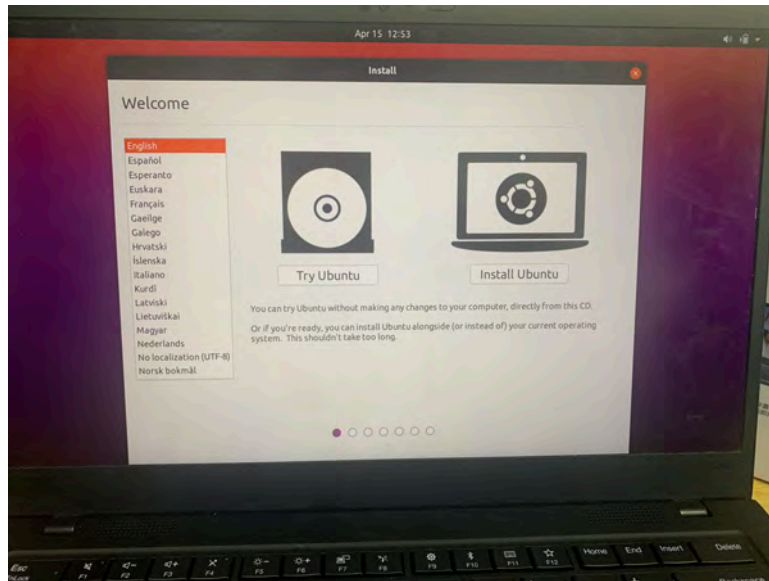
step 2 设置 U 盘启动为第一启动顺序

step 3 关闭安全模式

4 Ubuntu 安装

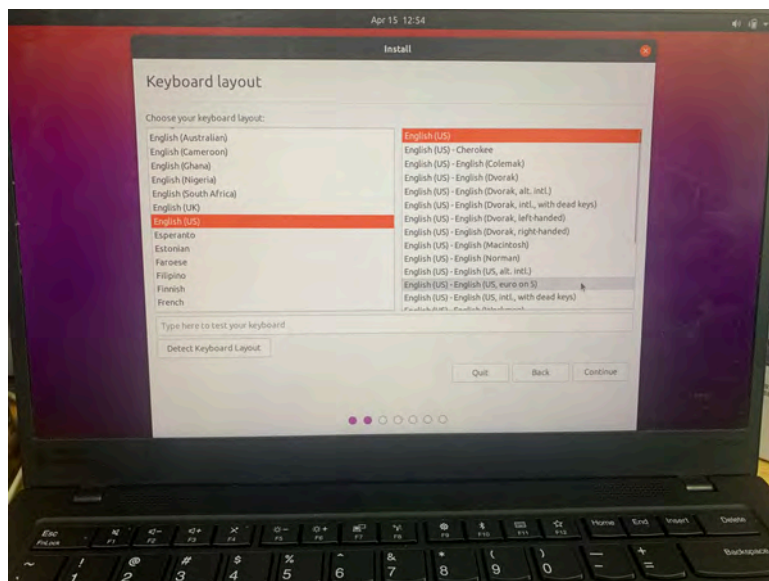
第一页 初始页面

自行选择中文或者直接选择默认英文 点击继续



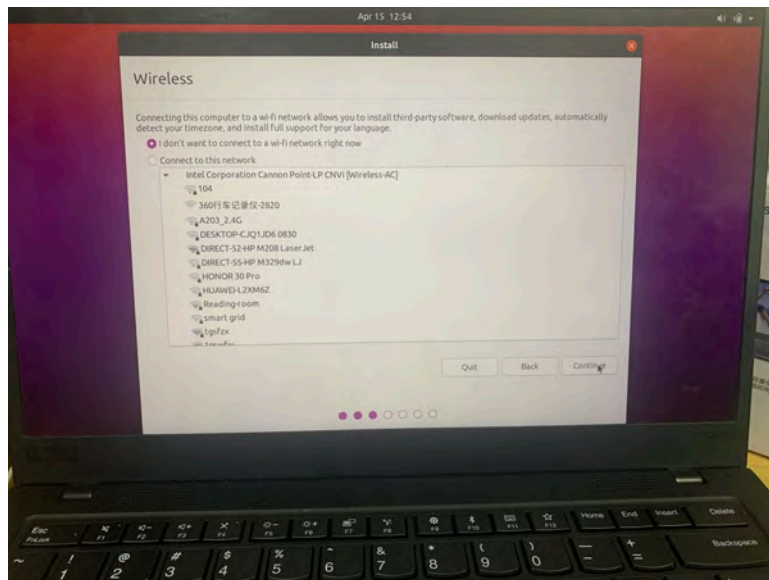
第二页 键盘布局选择

点击继续



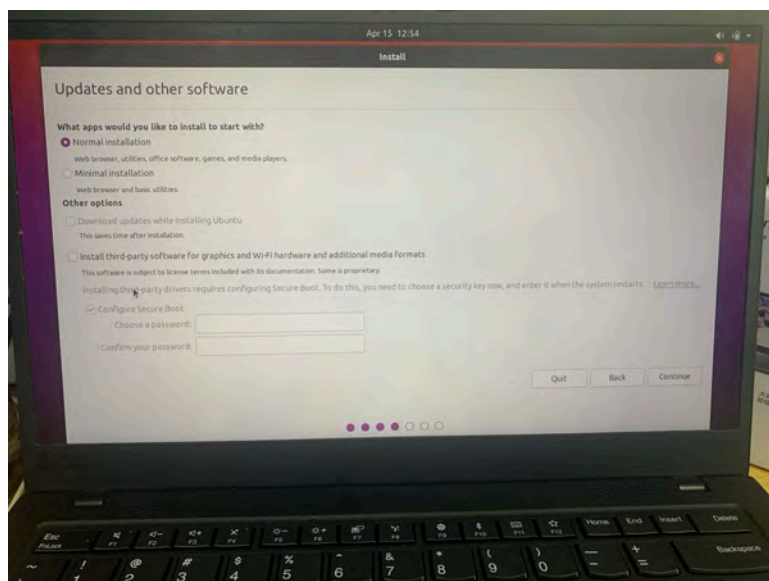
第三页 连网安装选择

可以选择连网也可以不连 点击继续

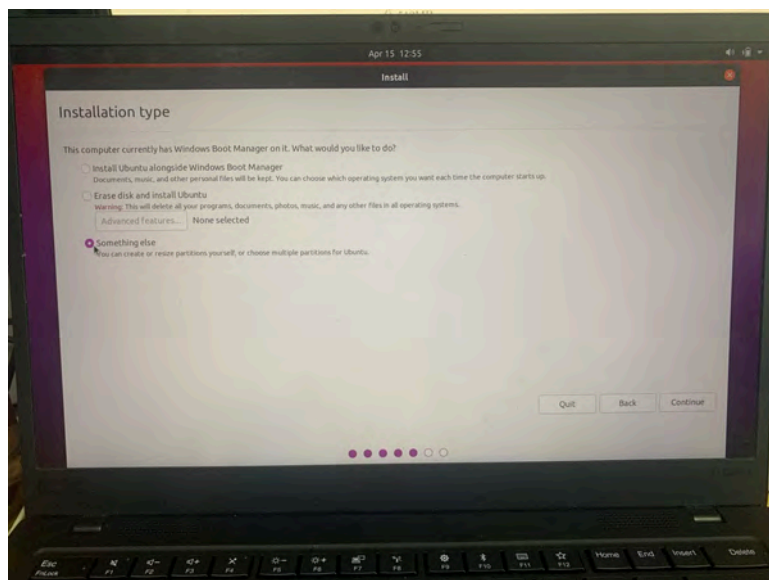


第四页 更新和其他软件

选择正常安装其他选项中都勾上 (ubuntu 没连网的情况第一个安装时更新选不上) 注意: 其他选项中第二个选项勾上后, 在安装第三方软件应该确认 BIOS 中安全模式已经关闭



第五页 安装类型 注意这是关键选项

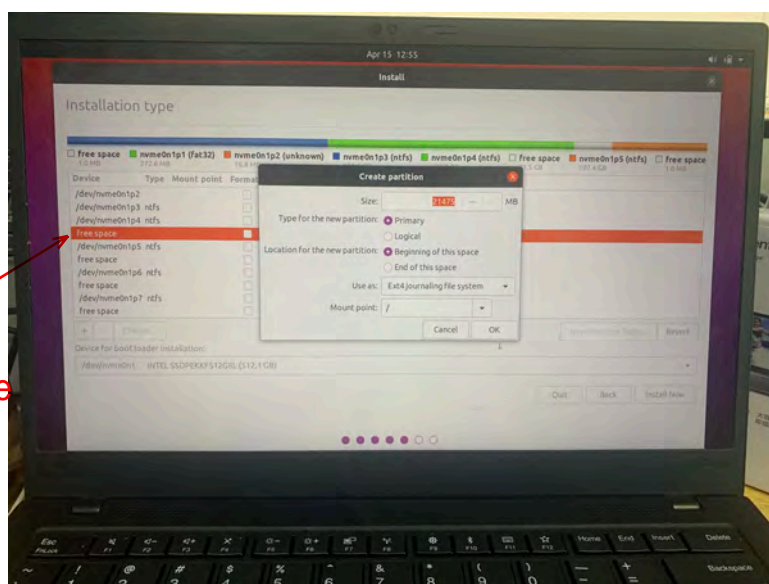


选择最下面 其他选项 (something else) 点击继续

第六页 分区 注意这是关键选项

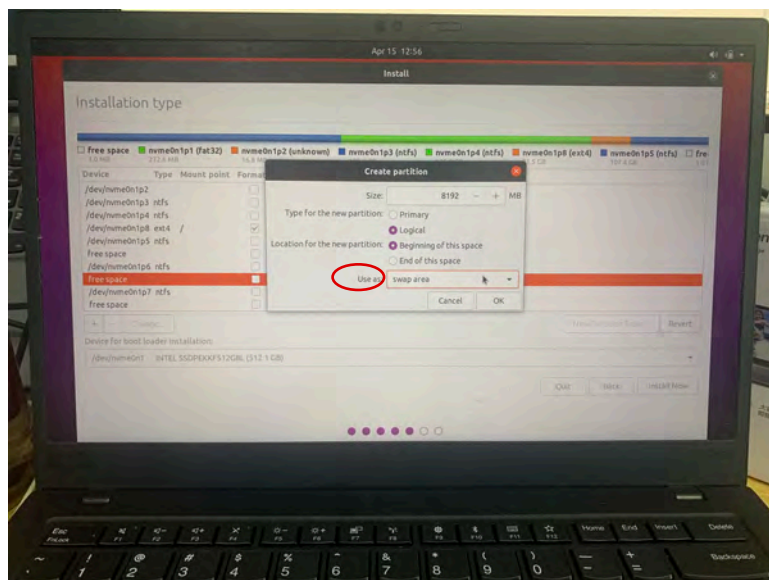
一共分为四个区: / 根目录 swap 交换空间 /boot 为启动分区
/home 主目录

/ 建议分 30G 以上

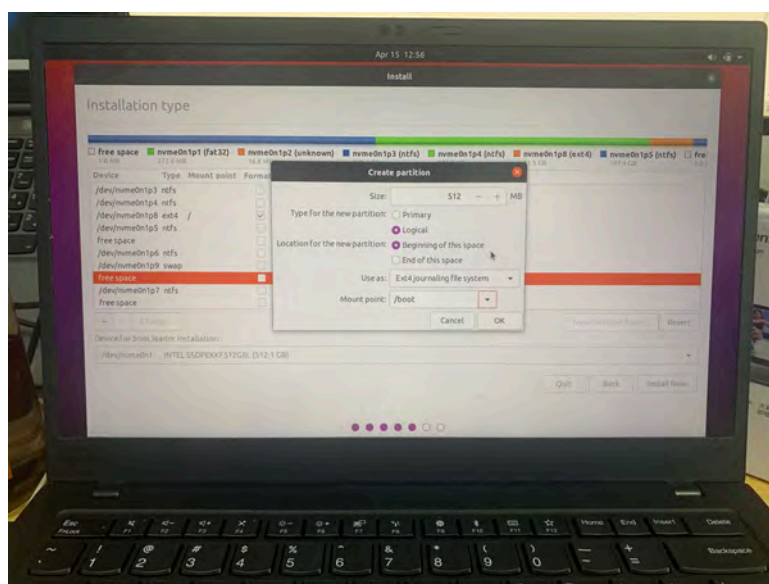


我们压缩空间的那的free space

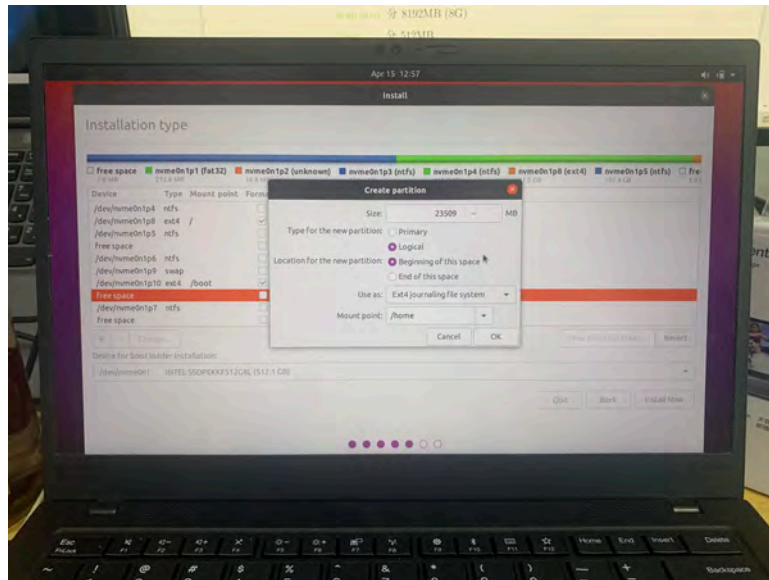
swap area 分 8192MB (8G)



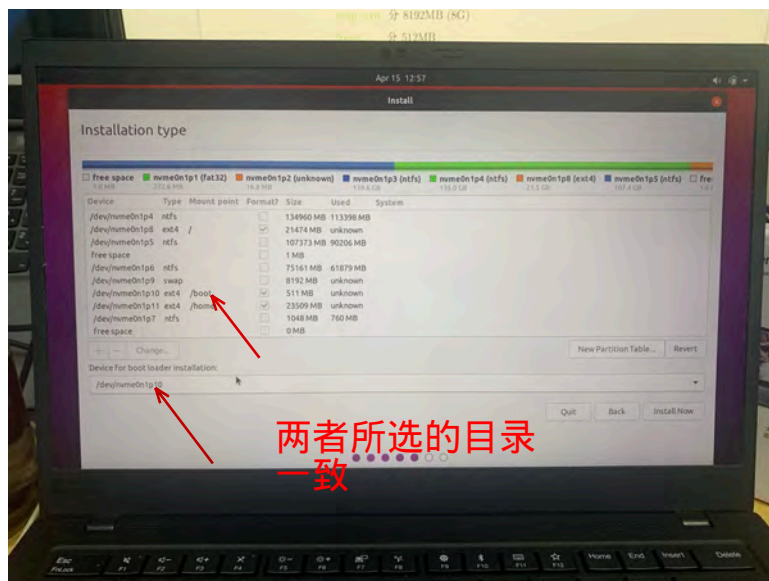
/boot 分 512MB



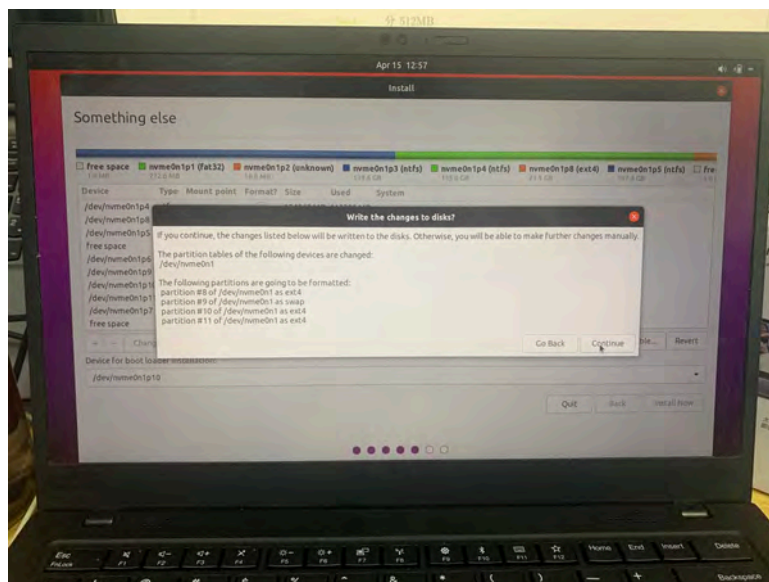
/home 剩下的空间全部



安装启动引导的设备选择 /boot 对应的设备

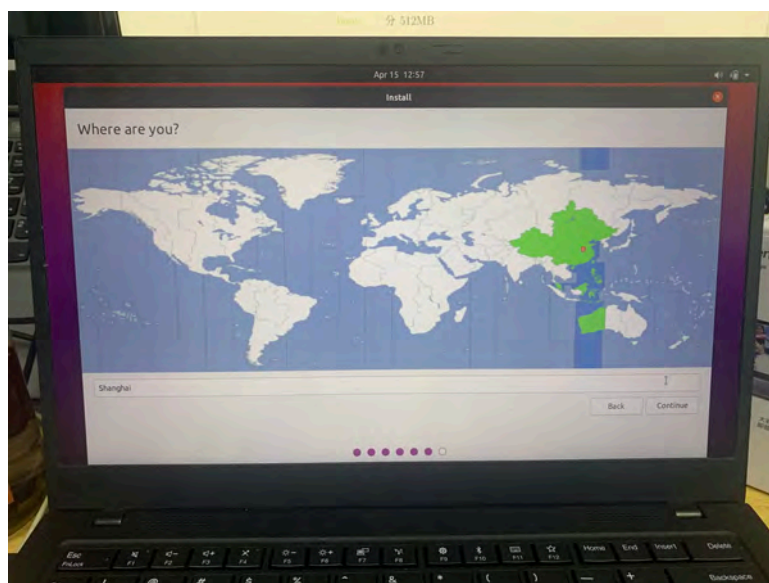


继续



第七页 选择地区

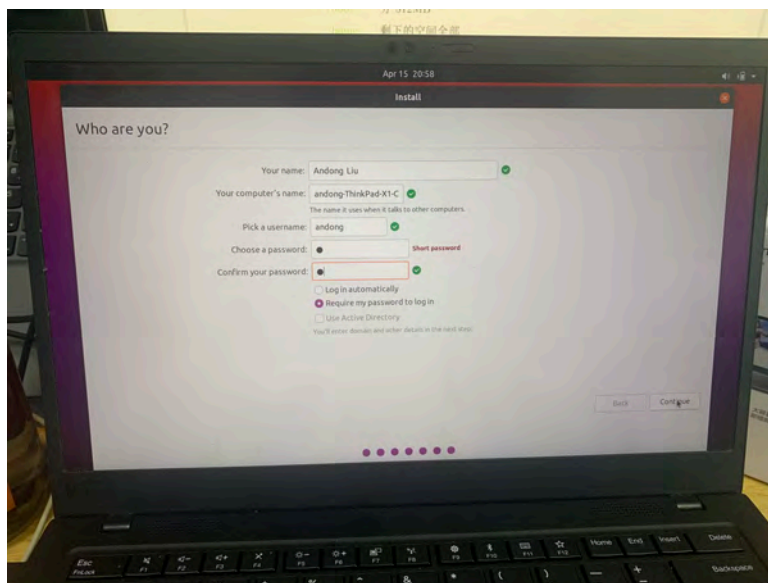
选上海 点击继续



第八页 设置用户

根据自己设置记住密码，点击继续 开始安装

username 为系统登录账户名 在 ubuntu 终端中 计算机名为 @
后 用户名为 @ 前



最后根据电脑提示, 拔下 U 盘按下回车重启电脑

5 Ubuntu 设置

5.1 ubuntu 换源

(推荐中科大源 <https://mirrors.ustc.edu.cn/>)

进入网站 <https://mirrors.ustc.edu.cn/help/ubuntu.html> 参照下图进行操作

当然也可以直接编辑 `/etc/apt/sources.list` 文件 (需要使用 `sudo`)。以下是 Ubuntu 20.04 参考配置内容:

```
# 默认注释了源码仓库，如有需要可自行取消注释
deb https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted universe multiverse
# deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted universe multiverse

deb https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted universe multiverse

deb https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted universe multiverse

deb https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse

# 预发布软件源，不建议启用
# deb https://mirrors.ustc.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
# deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
```

更改完 `sources.list` 文件后请运行 `sudo apt-get update` 更新索引以生效。

输入以下代码，更新

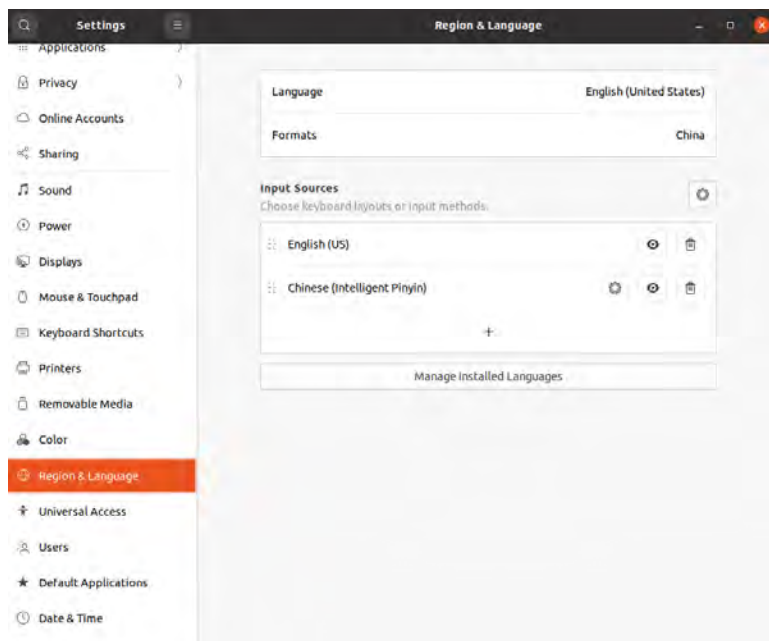
```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
```

5.2 中文输入法

5.2.1 中文输入法

设置 -> 区域和语言 -> 管理已安装的语言 -> 更新语言环境

完成以后 reboot 重启



点击加号 -> 选择中文 -> 智能拼音

5.2.2 搜狗输入法

参照官网安装教程 <https://pinyin.sogou.com/linux/help.php>

5.3 ubuntu 与 win10 时间不同步

```
//在 Ubuntu 下更新本地时间
sudo apt-get install ntpdate
sudo ntpdate time.windows.com
//将本地时间更新到硬件上
sudo hwclock --localtime --systohc
```

5.4 ubuntu flash 插件

```
sudo apt-get install flashplugin-installer
```

5.5 ubuntu 启动黑屏问题

修改两个系统文件

```
/boot/grub/grub.cfg  
/etc/default/grub
```

在上述两个文件中所有'quiet splash' 语句后添加'nomodeset'
执行以下命令, 后重启

```
sudo update-grub
```


6 ROS 安装

参考中科大源 <https://mirrors.ustc.edu.cn/help/ros.html>

参考 ROSwiki <http://wiki.ros.org/cn/noetic/Installation/Ubuntu>

设置 sources.list 和 导入 key (使用科大源)

1. 导入key:

```
gpg --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654  
gpg --export C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654 | sudo tee /usr/share/keyrings/ros.gpg > /dev/null
```

2. 将软件源添加至系统:

```
sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/ros.gpg] https://mirrors.ustc.edu.cn/ros/ubuntu
```

更新

```
sudo apt-get update
```

完整桌面版安装

```
sudo apt install ros-noetic-desktop-full
```

设置环境

Bash

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

安装成功测试

```
roscore
```

ros 安装各类包

```
sudo apt-get install ros-noetic-*
```

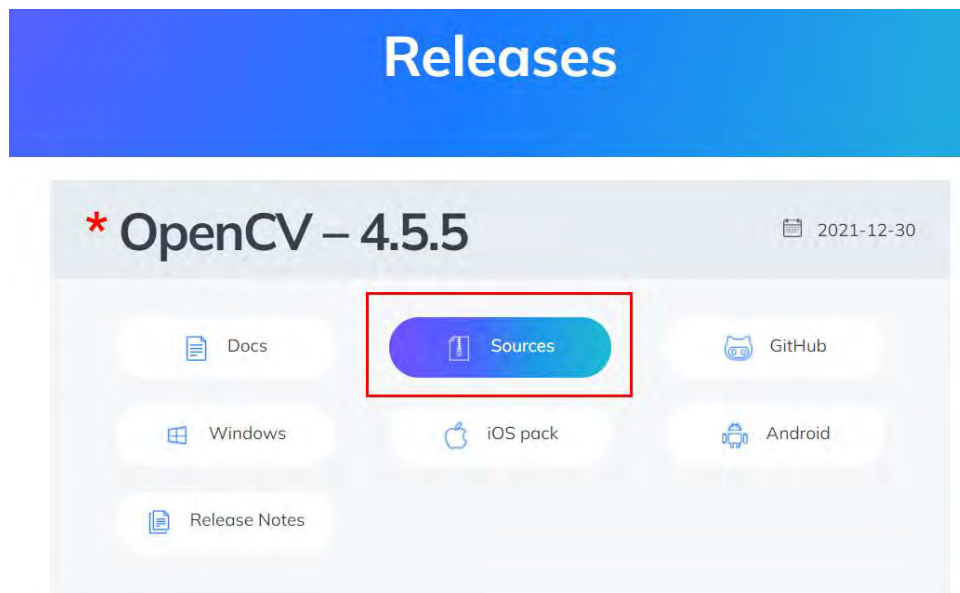
7 Opencv 4.5.5 安装

官网文档 <https://docs.opencv.org/4.5.5/index.html>

(参考 <https://blog.csdn.net/luxin5321/article/details/115369422>)

7.1 下载

点击 sources



7.2 安装依赖库

```
sudo apt-get install  
    build-essential libgtk2.0-dev  
    libgtk-3-dev libavcodec-dev  
    libavformat-dev libjpeg-dev  
    libswscale-dev libtiff5-dev
```

7.3 编译 OpenCV

进入到下载好的 opencv 目录中, 新建目录 build 并进入 build:

```
cd opencv
mkdir build
cd build
```

7.3.1 debug 模式

```
cmake -D CMAKE_BUILD_TYPE=Debug -D OPENCV_
GENERATE_PKGCONFIG=YES -D CMAKE_
INSTALL_PREFIX=/usr/local/OpenCV/
Debug -D WITH_FFMPEG=ON ..
```

7.3.2 release 模式

```
cmake -D CMAKE_BUILD_TYPE=Release -D OPENCV_
GENERATE_PKGCONFIG=YES -D CMAKE_
INSTALL_PREFIX=/usr/local/OpenCV/
Release -D WITH_FFMPEG=ON ..
```

7.4 安装 OpenCV

```
make -j8
sudo make install
```

7.5 设置库的搜索路径

```
sudo gedit /etc/ld.so.conf.d/opencv.conf
```

在 `opencv.conf` 中加入函数库所在的目录:

```
/usr/local/OpenCV/Debug/lib  
/usr/local/OpenCV/Release/lib
```

接下来利用 `ldconfig` 命令, 将 `/etc/ld.so.conf.d` 中的数据读入缓存

```
sudo ldconfig
```

7.6 添加 `pkg-config` 环境变量

7.6.1 debug

`/usr/local/OpenCV/Debug/lib/pkgconfig`

```
sudo mv opencv4.pc opencv4_debug.pc
```

7.6.2 release

`/usr/local/OpenCV/Release/lib/pkgconfig`

```
sudo mv opencv4.pc opencv4_release.pc
```

7.6.3 路径设置在环境变量中

```
sudo gedit /etc/profile
```

把以下代码写入文件的最后并保存

```
export PKG_CONFIG_PATH=/usr/local/OpenCV/  
Debug/lib/pkgconfig:$PKG_CONFIG_PATH  
export PKG_CONFIG_PATH=/usr/local/OpenCV/  
Release/lib/pkgconfig:$PKG_CONFIG_PATH
```

```
source /etc/profile
```

7.7 验证安装

在目录:

```
/opencv/samples/cpp/example_cmake
```

依次执行

```
cmake .  
make  
./opencv_example
```

8 cuda11.5 安装

不需要安装显卡驱动,cuda 会自动安装上对应的显卡驱动

官网 <https://developer.nvidia.com/cuda-toolkit-archive>



9 cudnn8.3.3 安装

官方下载页 <https://developer.nvidia.com/rdp/cudnn-archive>

官方文档 <https://docs.nvidia.com/deeplearning/cudnn/archives/>

cuDNN Archive

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

Download cuDNN v8.3.1 (March 2022)

for CUDA 11.5

Local Installers for Windows and Linux, Ubuntu(x86_64, armsbsa)

Local Installer for Windows (Exe)

Local Installer for Windows (Zip)

Local Installer for Linux x86_64 (Tar)

Local Installer for Linux PPC (Tar)

Local Installer for Linux SBC (Tar)

Local Installer for Ubuntu19.04 x86_64 (Deb)

Local Installer for Ubuntu18.04 arm64v8 (Deb)

Local Installer for Ubuntu18.04 cross-armsa (Deb)

Local Installer for Ubuntu20.04 x86_64 (Deb)

Local Installer for Ubuntu20.04 arm64v8 (Deb)

Local Installer for Ubuntu20.04 cross-armsa (Deb)

Local Installers for Red Hat (x86_64, armsbsa, Power architecture)



10 torch 安装

官网下载页 <https://download.pytorch.org/whl/>

下载完成以后

```
pip3 install *
```

附录

跑火车的友人 A

2022-03-22 v1.0 创建

2022-04-26 v1.1 新增目录 增加 opencv cuda cudnn