Autonomous vehicle control project

User Guide

Table of Content

1. Ubuntu Introduction 4
   1.1 Linux 4
   1.2 Ubuntu Linux 4
   1.3 Ubuntu Desktop 5
   1.4 Ubuntu Commands 9
   1.5 Supplementary Materials 15
2. Setup Ubuntu System 15
   2.1 Virtual Machine For Windows User 15
   2.2 Create New Virtual Machine in VMware 19
   2.3 Install Ubuntu in Virtual Machine 25
   2.4 Configure your Ubuntu repositories 29
   2.5 Set up your keys 31
3. ROS Introduction 32
   3.1 System structure 32
   3.2 Basic tools 33
   3.3 Supplementary Materials 33
4. ROS Practice 33
   4.1.1 ROS Install 33
      4.1.1 Install ROS 34
      4.1.2 Environment setup 34
      4.1.3 Dependencies Install & Initialization 34
   4.2 First Demo & Test 34
      4.2.1 Create a ROS Workspace 34
      4.2.2 Test 35
5. PID Basis 35
   5.1 Introduction 35
   5.2 Position PID 36
   5.3 Position PID Code 37
   5.3 STM32 Program Download via USB 38
6. Robot Car 40

6.1 Connect with Robot Car 40
6.2 Remote Login by SSH 41
6.3 Robot Car Program Mount and Compile 42
6.3.1 Mount the Robot Car to Ubuntu 42
6.3.2 Modify the System time 42
6.3.3 Install Sublime and open ROS code 43
6.3.4 ROS Code Modification and Compile 43
6.4 Control the Robot by Keyboard 44
6.4.1 The Initialization Node 44
6.4.2 The Keyboard Control Node 45
6.4.3 The code of Keyboard Control Function 46
Bug ListReference 47
Reference 48
Appendix A Detailed Comment of turtlebot_teleop_key.py 49

1. Ubuntu Introduction

1.1 Linux

In 1983, under the leadership of Richard Stallman, the GNU Project was set up to coordinate activities to develop a free operating system. They could develop all of the components required for an operating system except a kernel. The kernel is considered as the heart of a computer operating system. The kernel controls the resources on a computer and makes it available to various functions. The kernel acts as an intermediary between computer hardware and application software. For example, when you watch a movie on a computer when we press the volume button to increase the level of sound, the media player application passes the instruction to the operating system kernel. As per the instruction, the kernel increases the volume by interfering with the parts controlled by the sound system. The attempts of the GNU project to develop an operating system did not manifest due to various reasons. The kernel used in Linux-based operating systems developed in 1991 by Linus Torvalds, a computer science student from Helsinki, Finland. The new kernel was named Linux. The Linux kernel began to be used with other parts of the operating system developed for the GNU project. Such systems are known as the GNU Linux. With the development of the X-Window module in 1992, the Linux operating system became more and more accessible. Initially, the popularity of Linux was limited to the academic community and computer experts. Two Linux-based operating systems, Slackware and Debian, started in 1993. The Linux kernel-based operating systems became known as the Linux operating system. The development of the Linux kernel is still led by Linus Torvalds. The mascot of Linux kernel is a penguin, and its name is Tux. GNU General Public License developed by Free Software Foundation for the use and distribution of Free software. The GNU General Public license promotes the idea of copyleft, opposite of copyright. This license gives users the freedom to run, distribute, copy, modify, study, and develop software. So, free software proposes more freedom free of cost. Linux kernel is a Free software in nature with the adoption of GNU General Public License. As a result, there are no restrictions on building the operating systems using the Linux kernel. A large number of operating systems have been built on Linux kernels.

1.2 Ubuntu Linux

Although there were plenty of Linux-based operating systems, there was no one that could be used easily by the public. Mark Shuttleworth, a South African entrepreneur, came with the idea of Ubuntu Linux. Microsoft Windows was famous as an easy to use operating system and became popular among computer users. Mark Shuttleworth's goal was to develop a simple, Linux-based operating system for all sections of the community. He established a company named Canonical. Initially, the Ubuntu operating system was started as a project by the Canonical company. The Ubuntu Operating System is an almost Free software project. Ubuntu has gained immense popularity because of its simplicity of use compared to other Linux-based operating systems. Community involvement is there in all stages of Ubuntu development. The term Ubuntu borrowed from an African

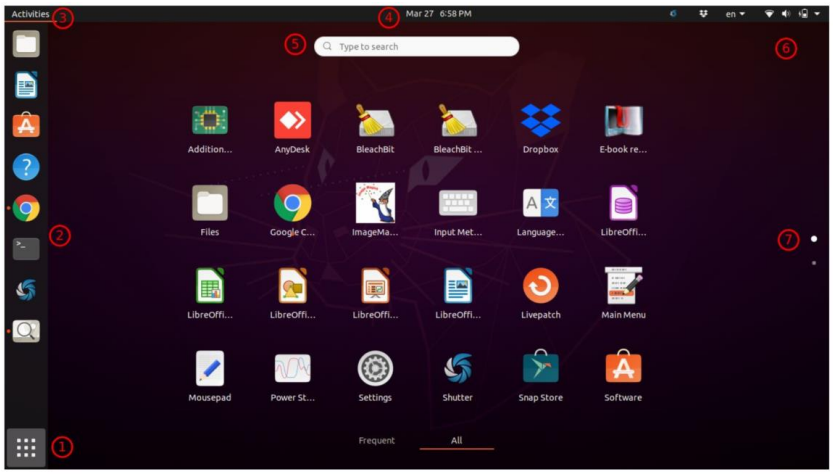
philosophical thought. The basic idea of Ubuntu is 'Humanity towards others'. Ubuntu Linux is first released on 20 October 2004. Ubuntu Linux copies are made available on the Internet for download. Ubuntu CDs were sent by post because of the slow speed of the Internet at that time. Installing Linux-based operating systems on the computer was complicated. Only those with the right expertise could successfully install the Linux-based operating systems. Complex installation procedures were a significant hindrance to the popularity of the Linux operating systems. Ubuntu introduced a simple installation method, and it attracted users. Customised GNOME-based desktop added the attractiveness of Ubuntu. Both of these factors boosted the popularity of Ubuntu Linux. Ubuntu became synonymous with the Linux operating system. Ubuntu is based on Debian Linux. Debian is a large Linux project with active community participation. Ian Murdock, a young computer expert, started the Debian Linux project in 1993. Debian is a popular Linux-based operating system among programmers and system administrators. Using Debian Linux was not simple for everyday use by ordinary people. Ubuntu has made several changes to the Debian Linux for public utility. Statistics show that Ubuntu is the most popular Linux-based operating system. Ubuntu mainly uses the GNU General Public License. Several operating systems were again developed based on Ubuntu. Linux Mint, Zorin OS, Elementary OS, Peppermint OS and many more operating systems built on the Ubuntu platform.

1.3 Ubuntu Desktop

The desktop is a critical component of an operating system that enables the user to communicate efficiently with the system. The desktop also has the functionality of making the operating system more attractive and comfortable to use. Themes, file Manager, wallpaper, panels and app launcher determine the look and feel of the desktop. Proprietary operating systems use the inbuilt desktop developed exclusively for the purpose. The desktop of the Windows operating system is not allowed to replace with another. Linux-based operating systems enable the users to change the default desktop and add another. There are plenty of desktops available for use on Linux-based operating systems. A Linux-based operating system always has a default desktop environment. You can install other desktops and switch to any desktop than your default desktop. Desktops give different looks for operating systems, the same as a person wears a different style of costumes. Ubuntu uses the GNOME desktop by default. Users can easily be familiar with the different graphical interface of Linux-based operating systems. Let's get acquainted with the GNOME desktop with Ubuntu.

● GNOME home

Ubuntu desktop consists of different elements. Get familiar with the different parts of the GNOME desktop.



An overview of the GNOME desktop. 1. Application launcher 2. Dashboard 3. Activities button 4. Notification area 5. Search bar 6. System menu 7. Workspace list.

Start applications

Find the favourite application icons on the dashboard located on the left side of the GNOME home. All of the applications can be visible on the centre of the desktop. Press the App Launcher (1) button at the bottom left of the panel. Users can search and find apps using the search box (5). It is possible to move to the next set of applications by pressing the workspace (two small dots) (7) on the right-hand side of the home page.

Favourite applications

Frequently used applications can be placed on the dashboard on the left side. Easy to launch favourite apps from the panel, and there is no need to find and open the application from the group every time.

Adding apps to the dashboard.

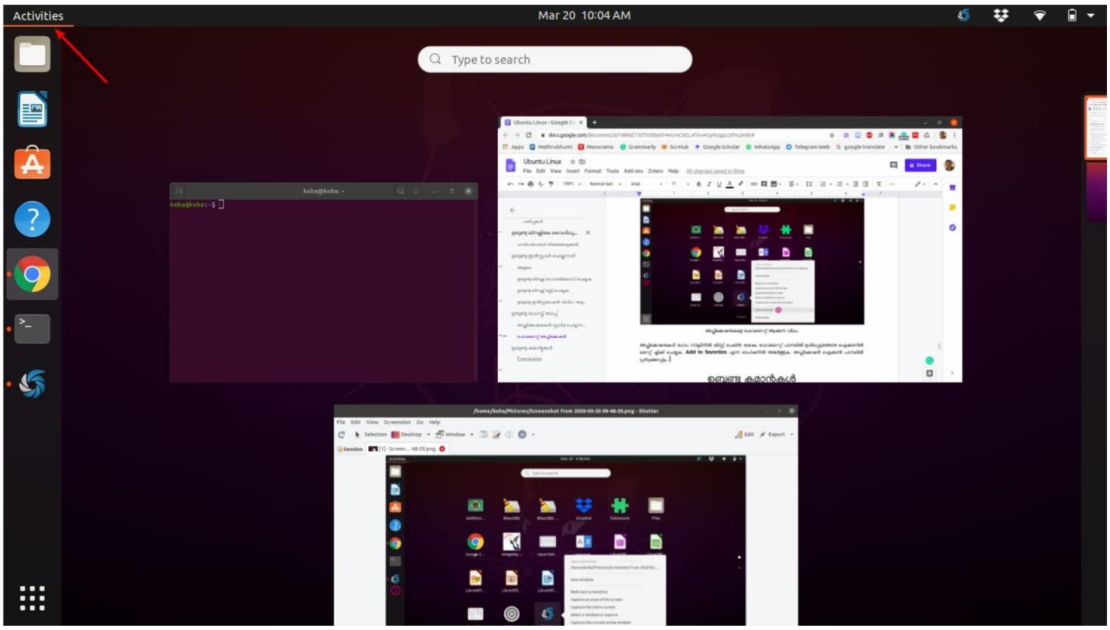
Right-click on the app icon on the dashboard and select **Add to favourites** option to include in the favourite applications. The app icon will appear on the panel.

Remove an app from favourites

Right-click on the app icon and select the option **Remove from Favorites**.

View running applications

Often users work with many applications and need to switch from one app to another. Click on the Activities option in the upper left corner of the GNOME home to view all the running applications.

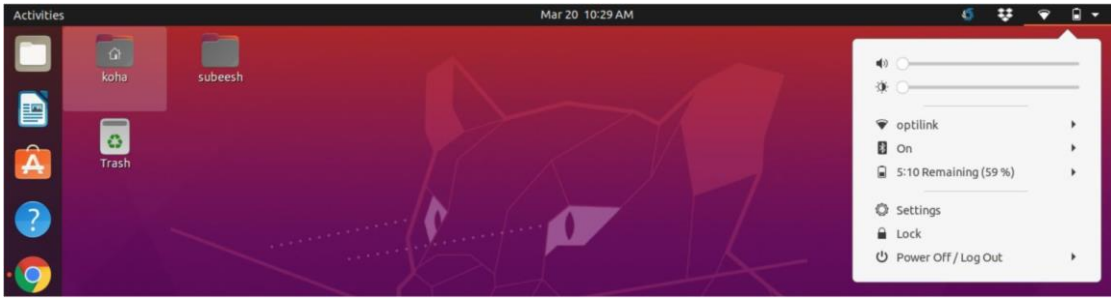


Activities menu

Press on the snapshots to maximize and open the applications. It's possible to close the unwanted apps from here by pressing on the **Close** button on apps windows right corner.

System menu

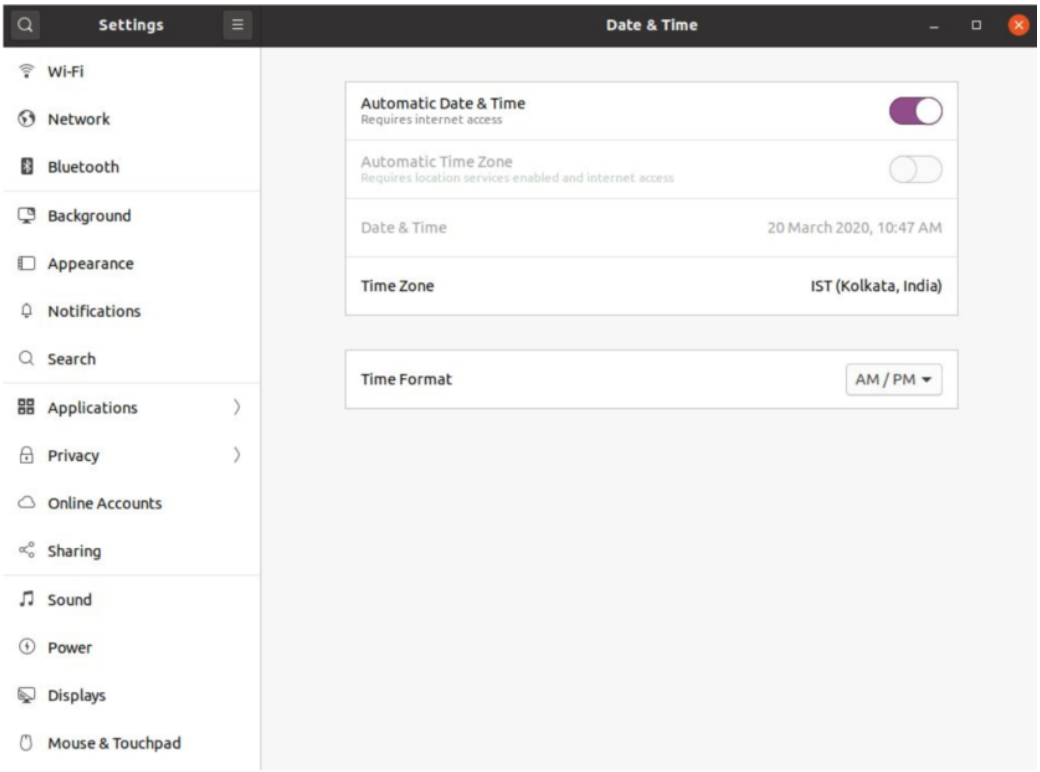
The System Menu is located on the upper right corner of the GNOME home screen. The system menu appears when pressing the mouse. Screen Lock, Restart, Shutdown and System Settings are arranged inside the **System Menu**.



System menu

System settings

To handle all Ubuntu settings, visit **System Menu > Settings**.



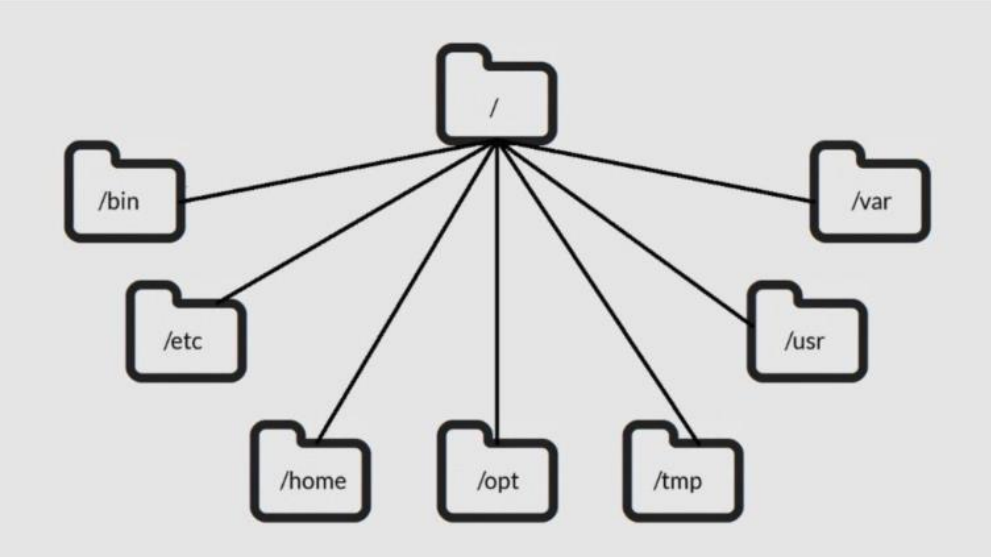
Ubuntu Setting

It can handle all kinds of settings, including network, wallpaper, printer, time and date, online account, voice and more. Online accounts with Ubuntu helps to sync cloud storage to your computer. Users can add Google Drive to access files from the Ubuntu File Manager.

Linux directory structure

Understanding the Ubuntu system is only possible if you are familiar with the Linux directory structure. The hard disk is partitioned to store data at the time of operating system installation. The data relating to the operating system are stored in files and folders. Each operating system has its directory structure. Linux directory structure is not like the Windows operating system. Windows operating system divides the entire storage space into C, D and E. C is the leading directory where system files are stored by default. Files are usually stored in C: \ Program Files when a new application is installed.

The directory structure of Linux can be compared to a tree root. Many branches start from the root. The Linux operating system directory structure starts from the Root directory. The / used to refer to the Root directory. There are other subdirectories within the Root directory.



Linux directory structure

The function of each directory in the Linux directory structure is described here very briefly.

Name of the directory	Functions
	The origin of all the directories is from here.
/bin	Here the binary files are stored. Details of functions described in the binary files associated with the application of various commands.
/etc	This directory contains the configuration files of all application software.
/home	Linux users store their personal files here.
/opt	Here store the directories and files of applications software installed from out of Ubuntu repository.
/tmp	Storage place of the temporary files when the applications software is running. Temporary files get cleared while restarting the computer.
/usr	Storage place of the application software installed and its source code.
/var	A log or history of various currently running programs.

Each subdirectory in the Linux directory structure has its own function. The home directory contains directories for the Linux users. The home directory for each Linux user includes the files and folders they create. When a Linux user account with the name Tom is created, a folder named Tom will be created inside /home directory. The location will be called /home/tom. The symbol / in the middle of folder names to indicate the hierarchy of directories. If Tom writes a love letter using LibreOffice; it is stored in his home folder. If Tina has another account, the /home/teena folder is created inside the home directory. Tina cannot enter Tom's home directory without permission to access his files.

When installing application software, associated files and folders stored in different system folders of Linux.

1.4 Ubuntu Commands

- Introduction

Commands are extensively used to do a specific task in the early days of the Linux operating system. At that time, the majority of the operating systems had no graphical user interface like a desktop. To print a document on modern computers, one needs to press the Print button. Then the software sends the instruction to the printer to start the print process. The Linux operating system provides the convenience of using both commands and graphical interface. System administrators, network administrators, database administrators, programmers, and Linux developers use Linux commands for the day to day activities. There are hundreds of Linux commands to apply in various contexts. In this chapter, we are going to learn the basic Linux commands that the beginners should know while using Ubuntu.

● How to apply Linux commands

The tool that helps to apply commands on a Linux operating system is called Terminal. Type 'Terminal' in the search box on the GNOME home page to open the Terminal tool. Add the Terminal to the dashboard as a favourite application for quick launching. If you are using GNOME Classic Desktop, find the Terminal from, **Applications> System Tools**. The Terminal can also be opened using the short key, **Ctrl + Alt + T** buttons together.

Linux command Terminal

There are specific rules to be followed while using Linux commands. Otherwise, the Linux operating system will refuse to apply the commands. Beginners may find it a bit difficult when dealing with Linux commands. The commands for Linux must be case-sensitive. Whether the commands are in lower or upper case, it must be entered as such. Otherwise, the command will misinterpret and stop the action without implementing. Linux commands can have multiple parts. Space should be placed between each part. Otherwise, the Linux operating system will stop the action without applying commands. Linux users should be careful to follow syntax while applying commands. Type the command on the terminal and press the **Enter** button to apply commands. If the command is applied correctly, the result would be displayed on the Terminal screen or jump to the next prompt. If the command is not valid, the error message will be displayed.

● Sudo command

The ultimate power on Linux operating systems is focused on the administrator; he is called the Super User / Root User. The sovereign of a nation will be the head of state. All power is concentrated at the head of the country. The same concept is applied to the **Super User** concept in the Linux operating system. A user must have the privilege to perform such tasks as installing, removing software, upgrading, opening files, modifying them, copying and moving file-folders. A user account is created while installing Ubuntu Linux. A user name and a password are assigned to the Linux account. This prime account holder is the **Super User** of Ubuntu Linux.

To work as a superuser, you must add the word sudo to all of the commands. The full form of the word sudo is **Super User Do**. The Password will be asked when the sudo command is applied. Type the password assigned to the account you created at the time of Ubuntu installation. The * special character is symbolically displayed when a password enters on the Windows operating system. While typing a password in a Linux terminal, nothing will be displayed. Type the password in the blank space and press the **Enter** button. The command will start running and show the result. Once you have added the sudo command, you will not be asked a password till 15 minutes.

● APT commands

Advanced Packaging Tool (APT) is used by Ubuntu to perform new tasks such as installing new software packages, upgrading the already installed software packages, updating the package list, and upgrading. Let us get acquainted with a set of commands used to control the APT system.

● Commands for Ubuntu update

Ubuntu Linux is a community project. Many experts are working on the development of Ubuntu, and the new updates come out frequently. Minimum five years are needed for the release of a new edition of Microsoft Windows. A new version of Ubuntu is released every six months. Ubuntu receives updates for each version regularly. Users need to know the commands to collect and add new updates to Ubuntu installation.

- Update the package index

Ubuntu software repositories are available all over the world where they store collection of software packages. Ubuntu software repositories are available in India. Ubuntu users in India receive updates from software repositories dedicated to India. The software repositories in the country help the users to download the updates faster. Addresses of the software repositories are stored on the computer where Ubuntu is installed. For example, the new version/update of LibreOffice Software first appears in the software repository. The Ubuntu user needs to fetch the latest version packages from the software repository. Type the following command to update the package index on your computer. It fetches the latest package information available in the repository. While using update commands, the user should make sure that your computer has Internet access.

```
$ sudo apt update
```

Enter the password when prompted. This command checks the repositories and gathers information about the latest versions of the packages available.

- Upgrade

Use the following command to download and install the latest version of the software and associated packages that are currently installed.

```
$ sudo apt upgrade
```

The above-mentioned commands need to apply regularly to keep Ubuntu Linux new.

- Install GNOME Classic desktop

Apply the following command to install the GNOME Classic desktop.

```
$ sudo apt install gnome-shell-extensions
```

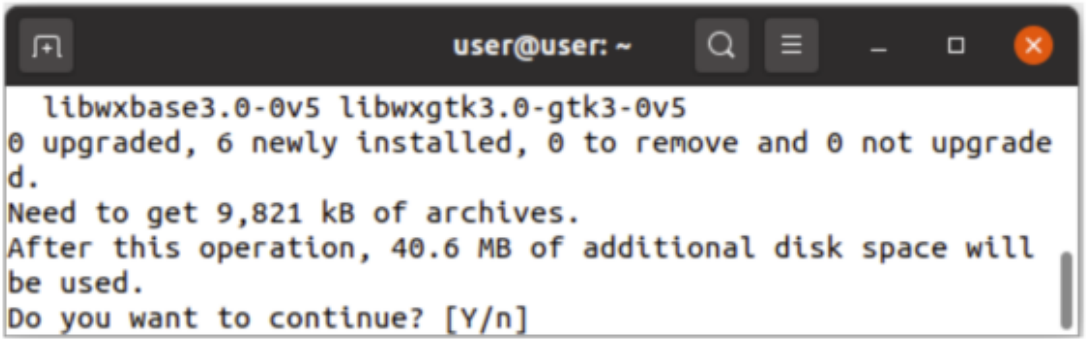
Once the installation is complete, log out from the GNOME desktop to enter into the classic desktop.

Select the Log Out option from the **System Menu** on the top right of the GNOME desktop. Pressing the button at the bottom of the login screen will pop up the list of desktops. Select the **GNOME Classic** from the list. Enter the password and log in.

- Install application software

New applications software can be downloaded using either software tool or using commands. Installing new software using the Ubuntu Software Tool is a simple process for beginners. Those who are familiar with Linux prefer installing software using commands. Install a new software Use the APT command to install new software from the Terminal. When applying the command, packages began to download from the Ubuntu Software repository and get installed on the computer. Apply the following command to install the image editing software GIMP.

```
$ sudo apt install gimp
```



Ubuntu will ask for confirmation on whether to install the software or not. Type **Y** to start the installation of GIMP. The packages began to download and install the GIMP software. GIMP software icon will appear on the **Applications > Graphics**.

- Uninstall software

Use the following command to remove the installed software.

```
$ sudo apt remove gimp
```

- Clean the packages

When updating Ubuntu and installing new software, several packages are downloaded to your computer. These packages are not required after the completion of update and installation processes. Cleaning such packages can save so much storage space. Here is the command to erase the obsoleted packages.

```
$ sudo apt clean
```

Use this command to remove associated packages that exist after uninstalling the software entirely.

```
$ sudo apt autoremove
```

- Install the software from the Debian package

There is also a way to install software that is not in the Ubuntu software repository. Software such as Google Chrome and Team Viewer is not available in the Ubuntu software repository. The package file for installing such software should be downloaded from the website. Open the File Manager and move the package file from the Download folder to the Home folder. The extension of the Debian package is .deb and is similar to the .exe extension with Windows application installer.

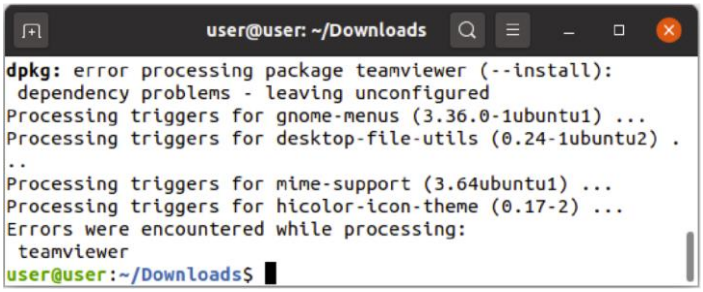
Apply the following command to install the Debian package of Google Chrome browser.

```
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

The package **google-chrome-stable_current_amd64.deb** is lengthy and there is a chance for making mistakes while typing. If the package name is not appropriately given, the attempt of installation will fail. There is an easy way to enter the package name without mistakes. Press the Tab button after typing the first few lines of the package name (e.g. **sudo dpkg -i goo**) and will autofill the rest of the name. Another method to get the full file name put the * symbol after the few letters. It will install all Debian packages starting with google.

```
sudo dpkg -i google*
```

In some cases, the installation of a Debian package may fail due to the lack of additional software packages.



Installation fails due to the lack of associated packages

In such a situation, apply the following command to download the missing packages to finish the installation of Google Chrome browser.

```
sudo apt install -f
```

The Google Chrome software icon will appear at **Applications > Internet** after the installation. Open the application software by pressing the mouse on the icon.

- Basic Linux commands

Here are a few more commands to know for those who want to use Linux.

To know the Ubuntu version

When using a computer with Ubuntu Linux installed, this command will help to know the version.

```
lsb_release -a
```

To know the system information

This command displays details of the system architecture.

```
$ lscpu
```

To know the date and time

This command will show the current date and time using this command.

```
$ date
```

To know the user name

This command can be used to determine which account is logged in.

```
$ whoami
```

To know the working directory

Each Linux account has its own folder. Once everyone is logged in, they will run from within that folder. This is called the Home folder of that account. The files and folders they create are stored in the Home folder. This command helps you to understand the location and name of the home folder.

```
$ pwd
```

To know the contents inside a folder

Apply the following command to list the contents inside the current folder you are in.

```
$ ls
```

To create a new folder

```
sudo mkdir tom
```

The above-mentioned command will create a folder called tom.

Remove/delete a folder

```
sudo rmdir tom
```

The above command will delete the folder called tom.

To enter inside a folder

There is a folder with the name Public in the home folder. Apply the following command to enter inside the Public folder using a terminal.

```
cd Public
```

Apply pwd command to check whether you are inside the folder, Public.

To exit from a folder

Apply the following command to exit from the Public folder.

```
cd ..
```

Apply pwd command to check the current location.

- Open a text file using Terminal

Many occasions Linux users need to open files and make changes in it. First, make sure which text editor application is available on your computer. Text editor applications can be found at **Applications > Accessories**. Linux-based operating systems are using different applications for the default text editor. For example, GNOME uses **Gedit**. Install a new text editor, if you are not sure about the name of the default text editor. The **Mousepad** is a lightweight text editor. To install Mousepad, apply the following command,

```
$ sudo apt install mousepad
```

- Open a text file using Mousepad

First, find the location of the file to open. For example, apply the following command to open the **syslog** file located at **/var/log**. We use the **Mousepad** text editor to open the file.

```
$ sudo mousepad /var/log/syslog
```

Syslog file contains the running history of each movement in Linux.

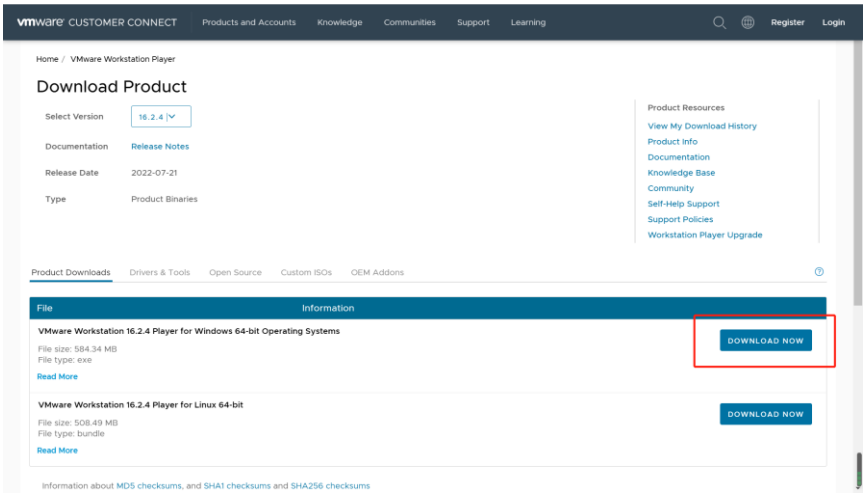
1.5 Supplementary Materials

(1) Linux Tutorial for Beginners: Introduction to Linux Operating System
https://www.youtube.com/watch?v=V1y-mbWM3B8
(2) Ubuntu Complete Beginners Guide
https://www.youtube.com/watch?v=D4WyNjt_hbQ

2. Setup Ubuntu System

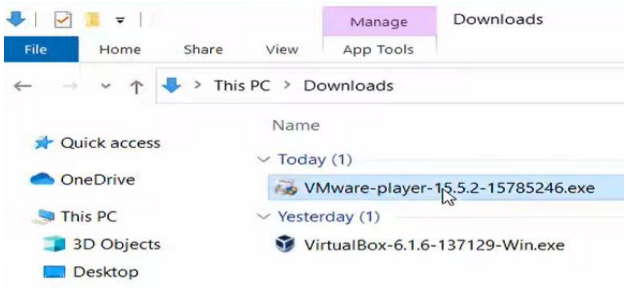
2.1 Virtual Machine For Windows User

If your computer is running on the Ubuntu System, please start with section 1.4.
If your computer is on Windows system and you do not want to reboot it to Linux directly, please install a virtual machine software 'VMware' first.
(1) Download VMware Installer
Download the VMware from the below link.
https://customerconnect.vmware.com/en/downloads/details?downloadGroup=WKST-PLAYER-1624&productId=1039&rPId=91446
After the down finished, you may need to restart you computer.

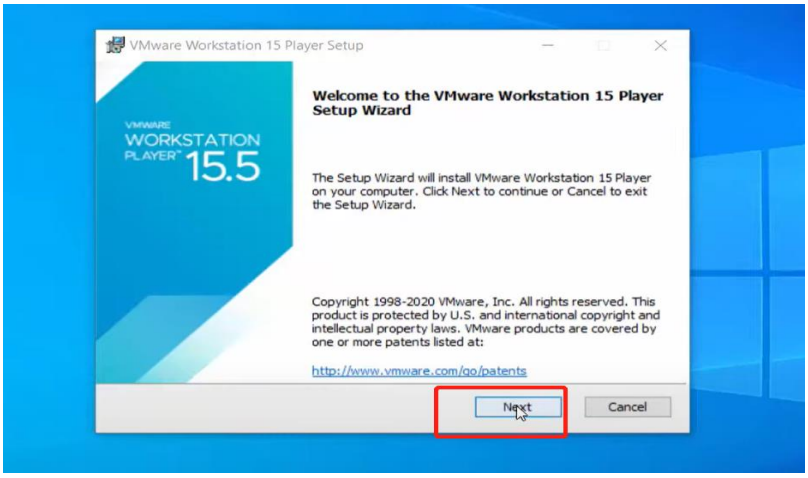


(2) Start Installation

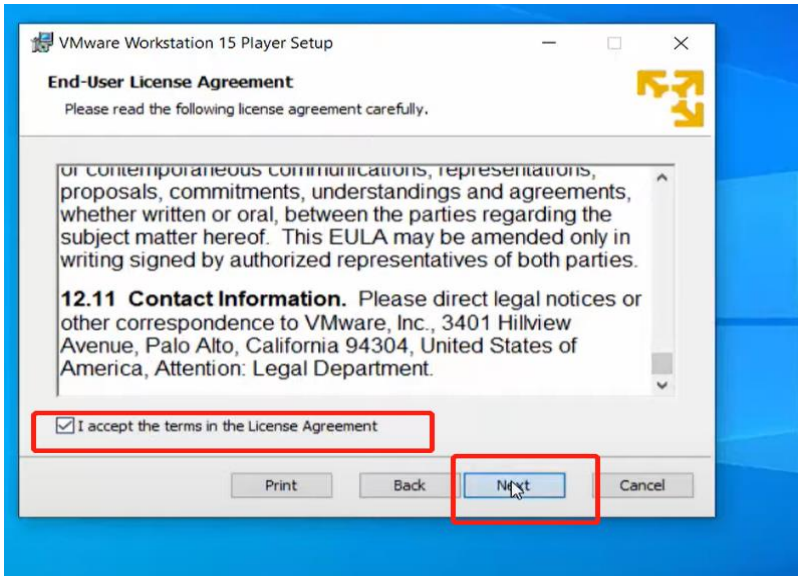
- Double click the installer icon to start it.



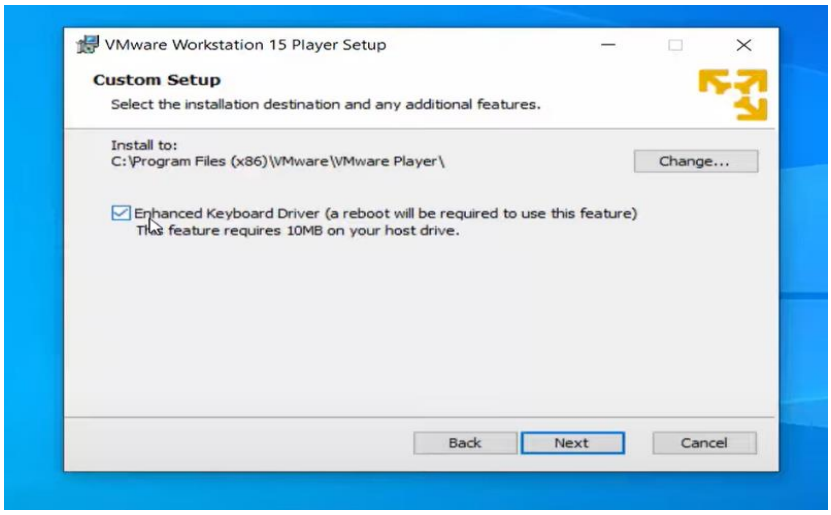
- Click next



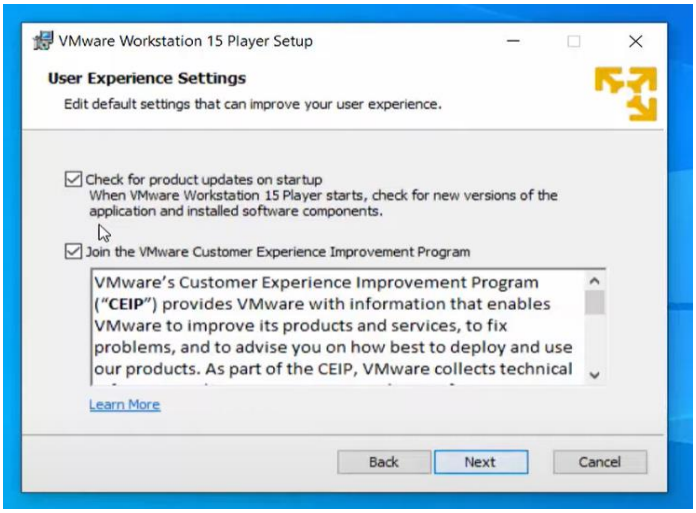
- Accept the license and click next



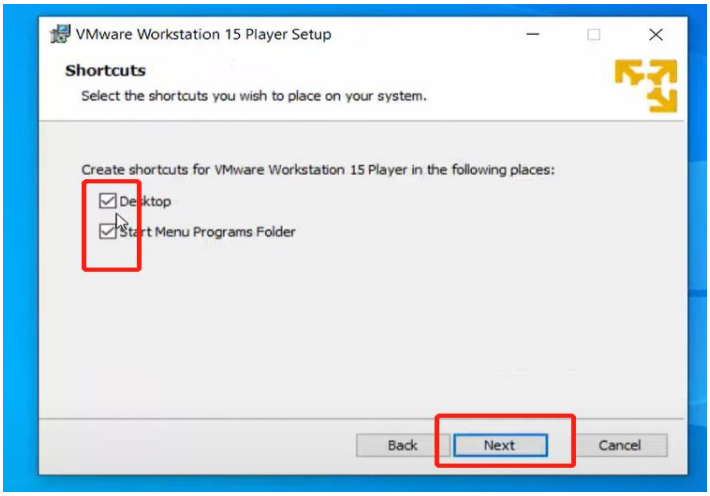
- Change your install directory if it is necessary. Tick at the 'Enhanced Keyboard Driver' Option. And click next.

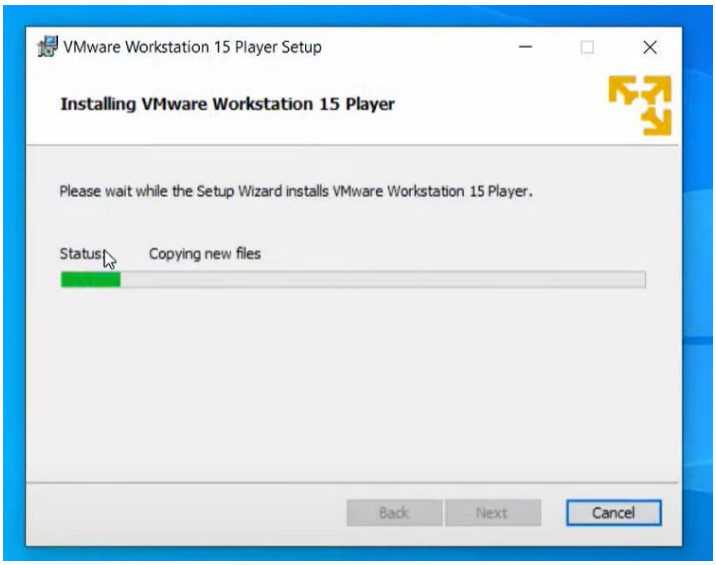


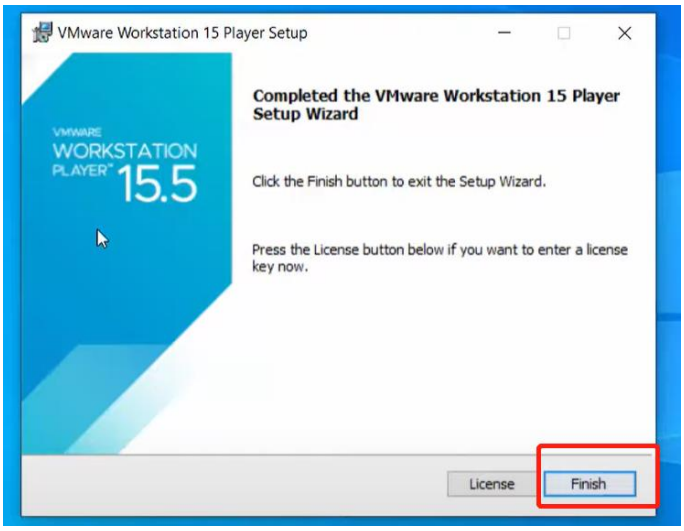
- Tick at the 'Check for product updates on startup' and 'Join the VMware Customer Experience Improvement Program' options. And click next.



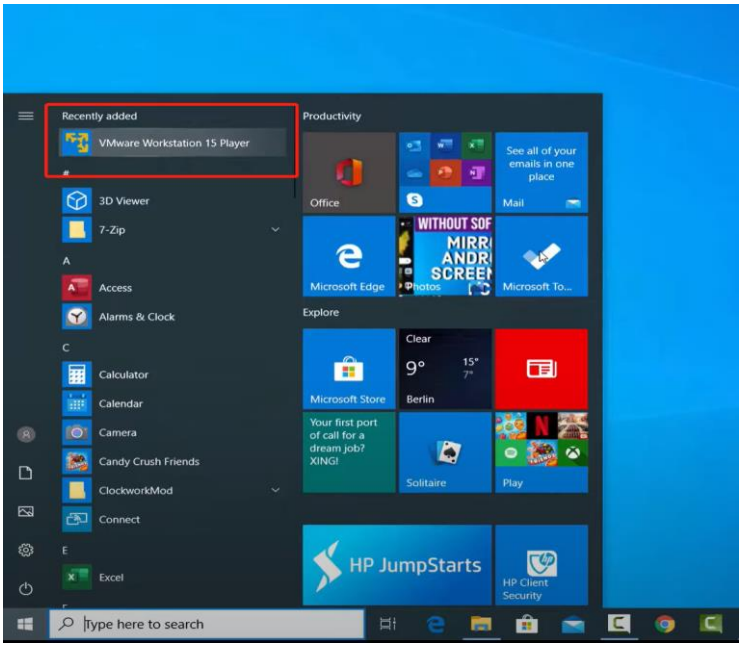
- Configure the shortcuts as default. And click next. And the installation starts automatically.

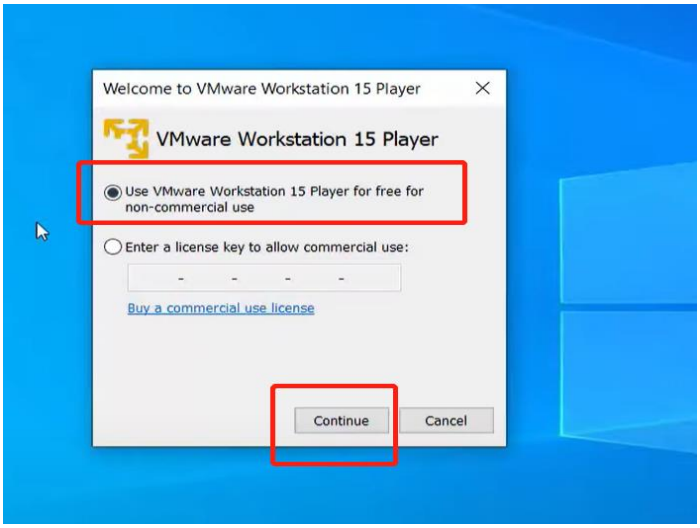


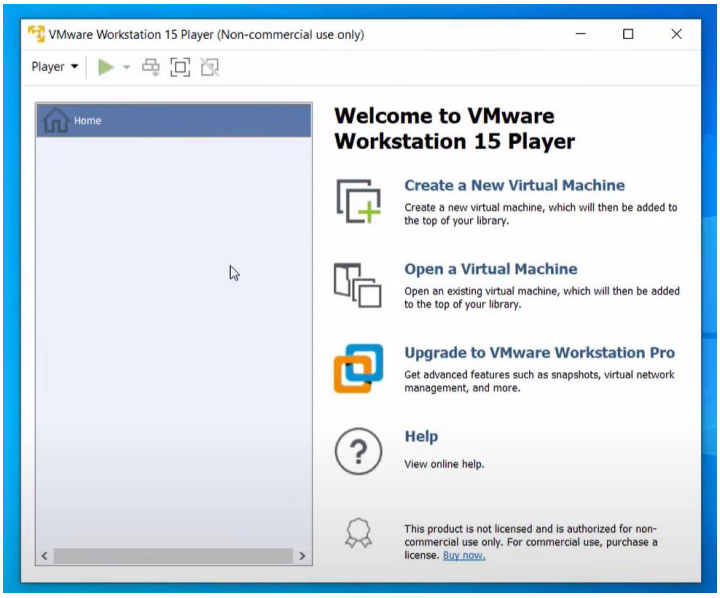




- Find your VMware and start it to use.







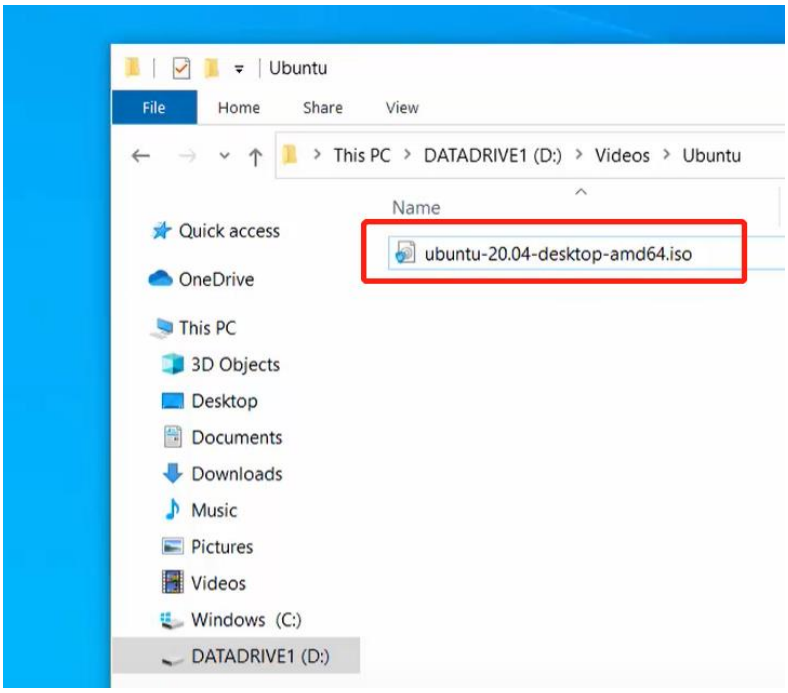
2.2 Create New Virtual Machine in VMware

(1) download the Ubuntu 20.04 LTS .iso file from the following link: https://releases.ubuntu.com/20.04.4/?_ga=2.102817728.13376874.1660447457-1634720039.1654648890

A full list of available files, including BitTorrent files, can be found below.

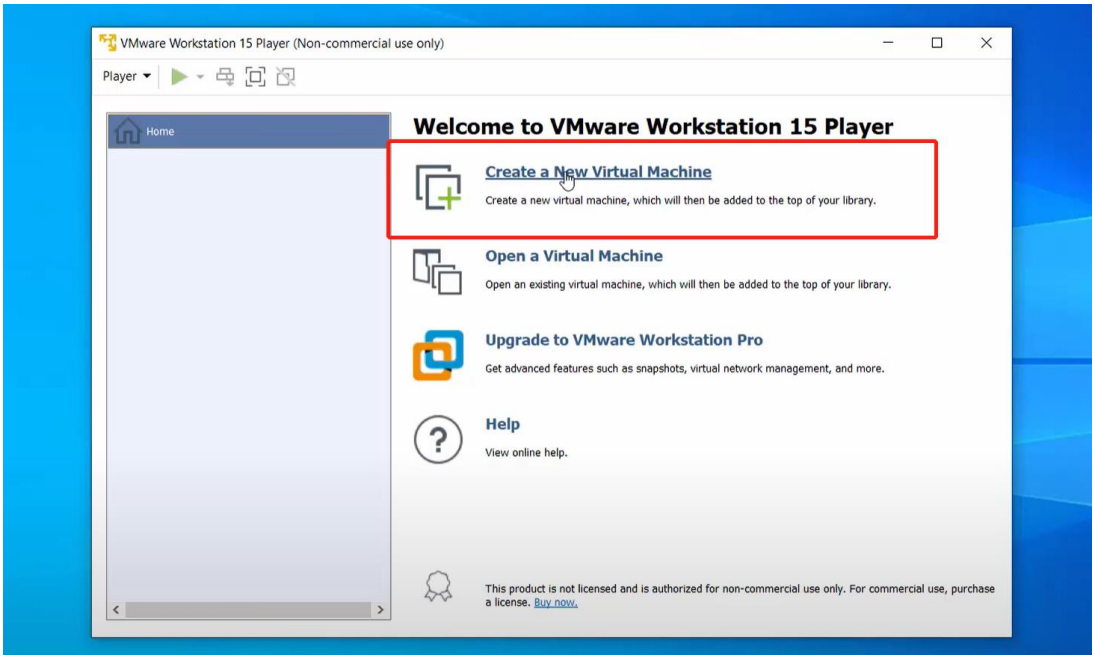
If you need help burning these images to disk, see the Image Burning Guide.

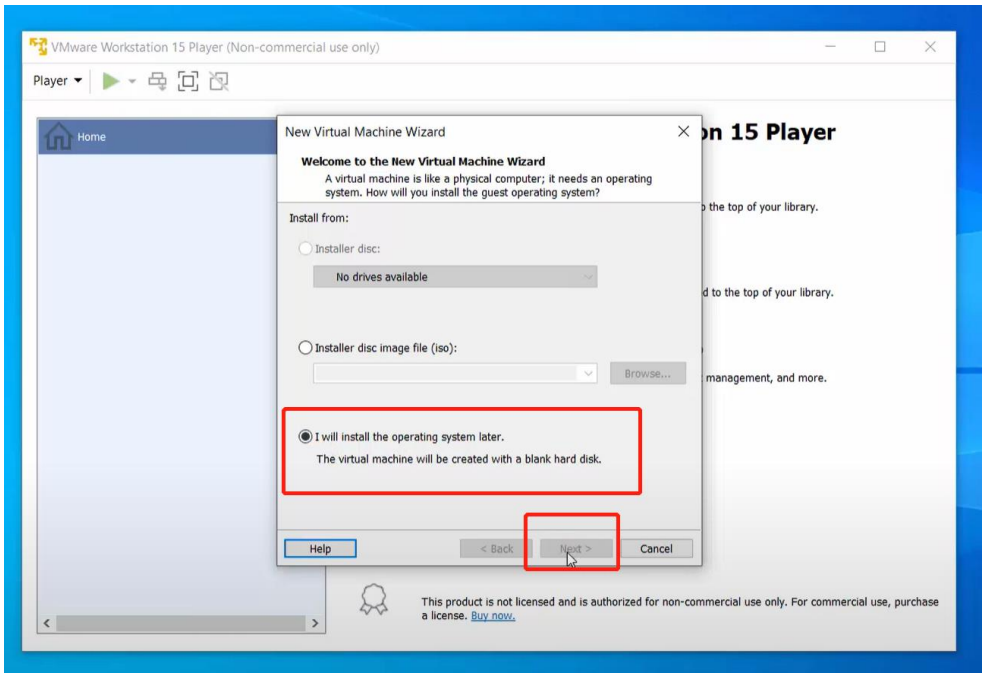
Name	Last modified	Size	Description
Parent Directory		-	
SHA256SUMS	2022-02-24 20:36	202	
SHA256SUMS.gpg	2022-02-24 20:36	833	
ubuntu-20.04.4-desktop-amd64.iso	2022-02-23 09:10	3.1G	Desktop image for 64-bit PC (AMD64) computers (standard download)
ubuntu-20.04.4-desktop-amd64.iso.torrent	2022-02-24 20:30	252K	Desktop image for 64-bit PC (AMD64) computers (BitTorrent download)
ubuntu-20.04.4-desktop-amd64.iso.zsync	2022-02-24 20:30	6.3M	Desktop image for 64-bit PC (AMD64) computers (zsync metafile)
ubuntu-20.04.4-desktop-amd64.list	2022-02-23 09:10	28K	Desktop image for 64-bit PC (AMD64) computers (file listing)
ubuntu-20.04.4-desktop-amd64.manifest	2022-02-23 09:03	58K	Desktop image for 64-bit PC (AMD64) computers (contents of live filesystem)
ubuntu-20.04.4-live-server-amd64.iso	2022-02-23 09:27	1.2G	Server install image for 64-bit PC (AMD64) computers (standard download)
ubuntu-20.04.4-live-server-amd64.iso.torrent	2022-02-24 20:28	100K	Server install image for 64-bit PC (AMD64) computers (BitTorrent download)
ubuntu-20.04.4-live-server-	2022-02-24 20:28	2.5M	Server install image for 64-bit PC (AMD64)

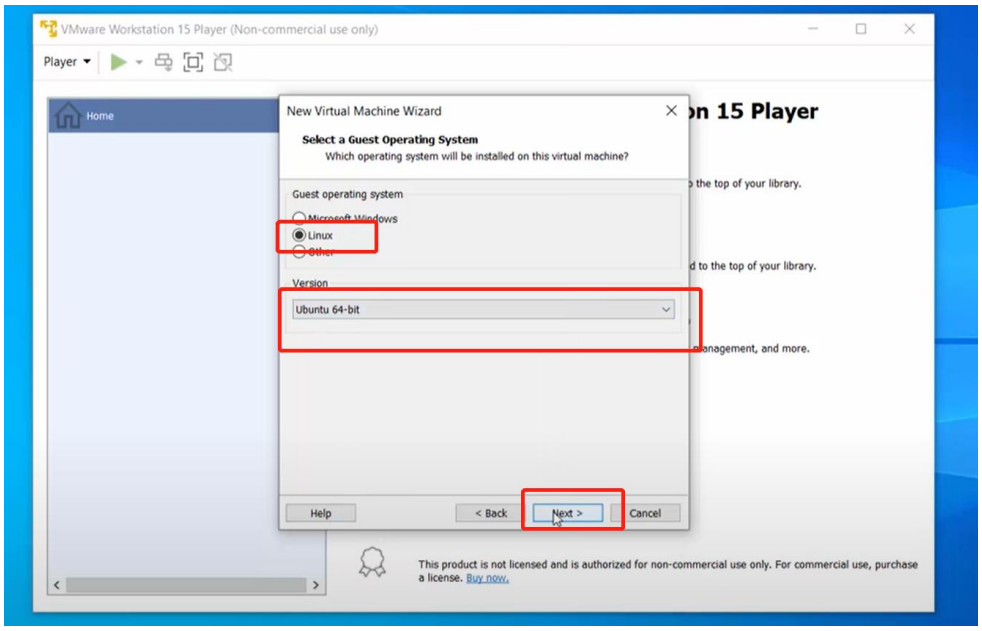


(3) create a new Virtual Machine in VMware

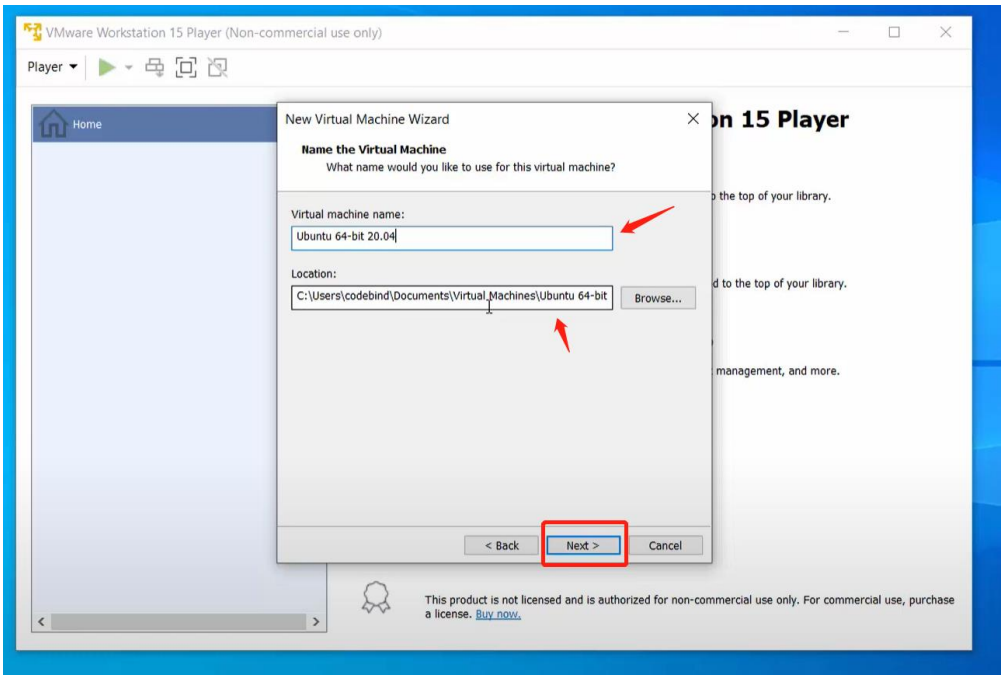
- select a guest operation system



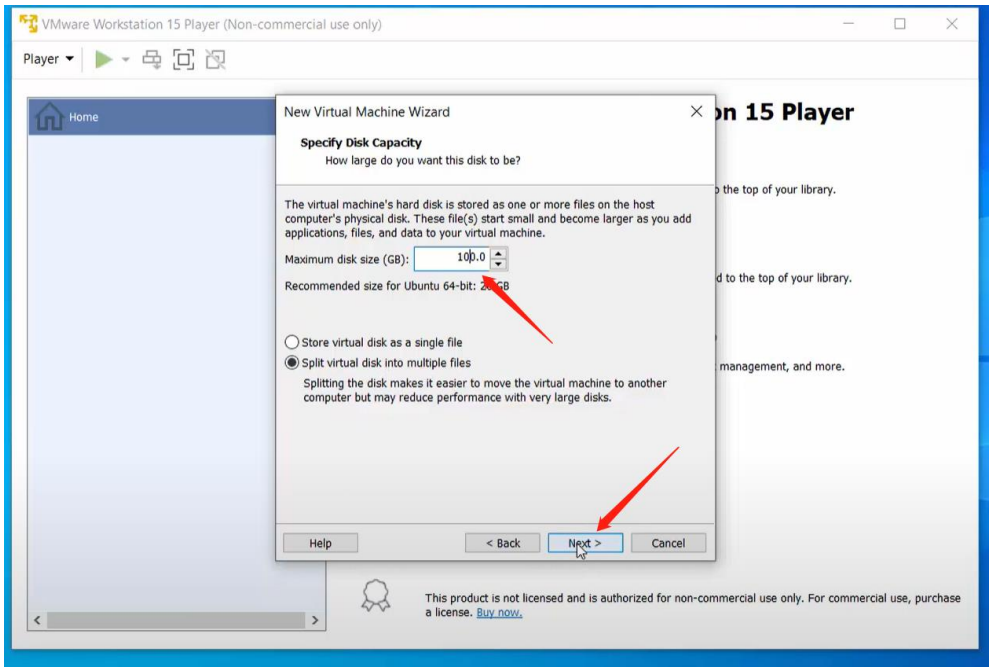




- Configure the name and location of your Virtual Machine

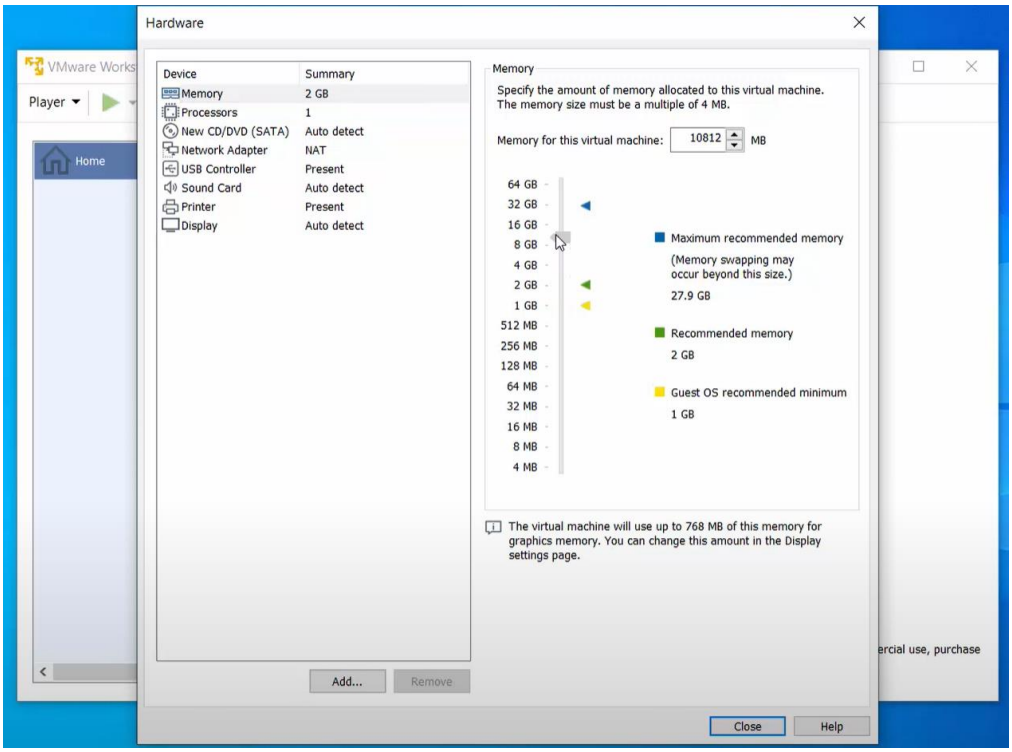


- Specify the Disk capacity

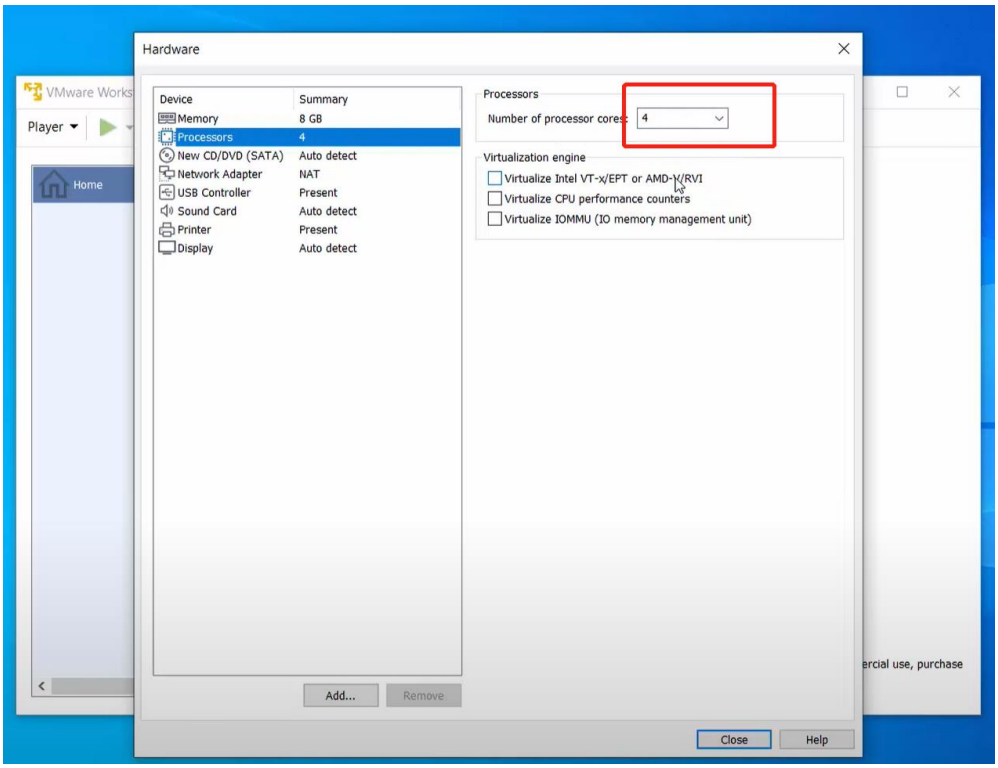


- Customize Hardware

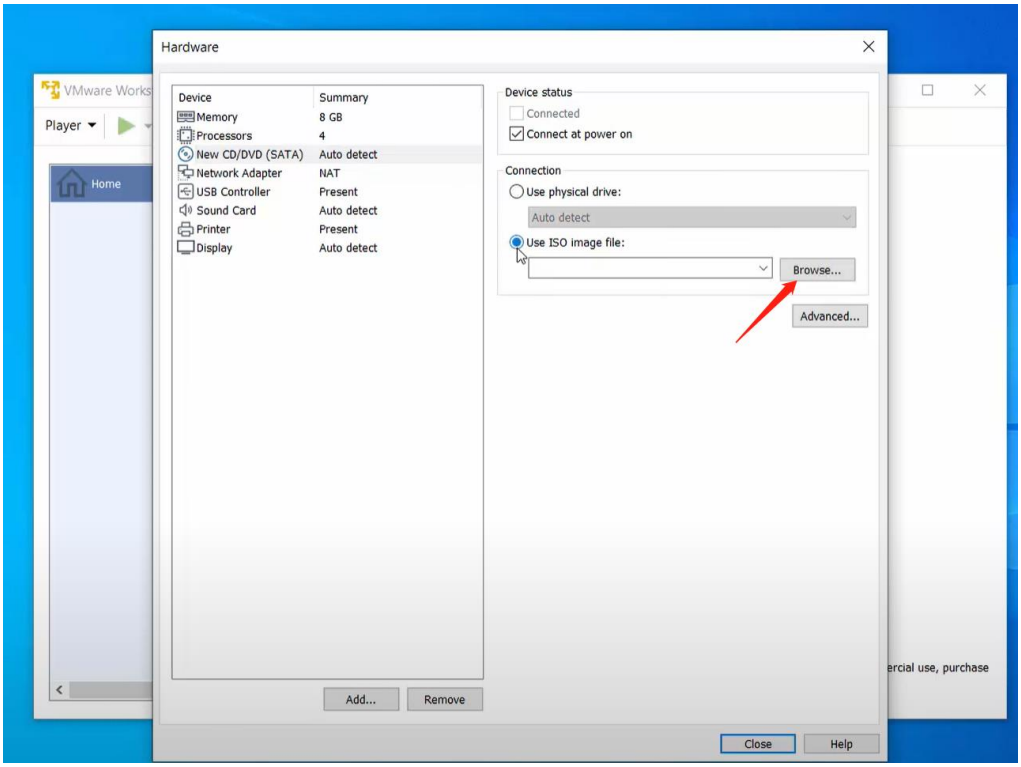
  Change the ‘Memory’ to 8GB

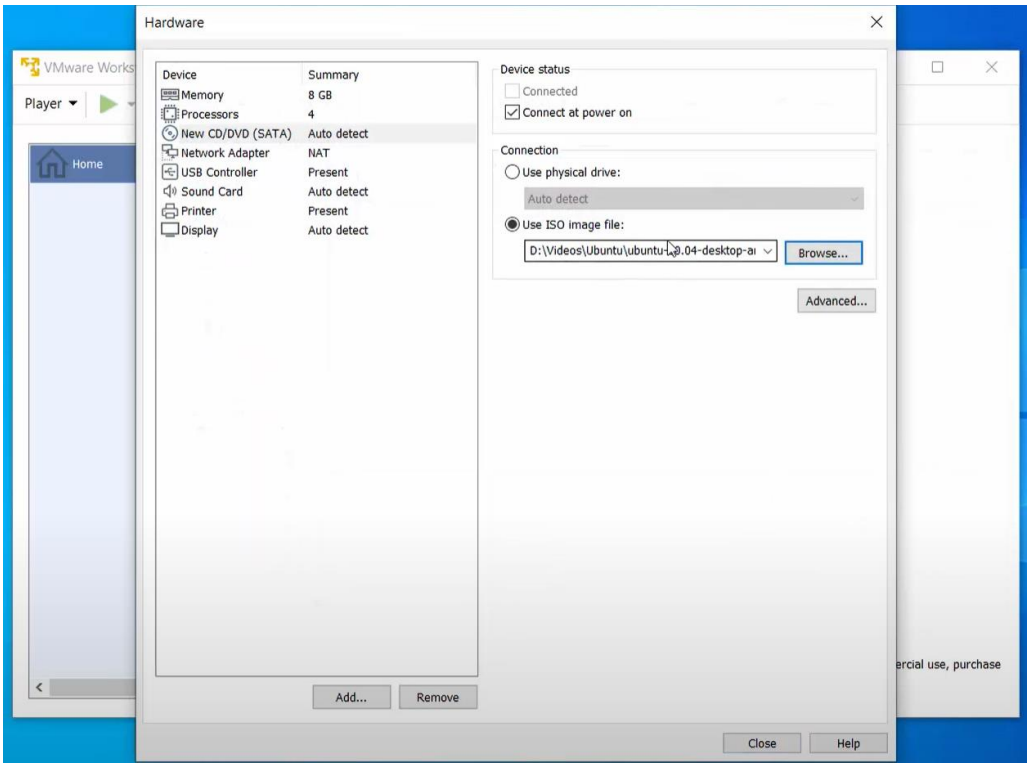


Change the Processors Number to 4

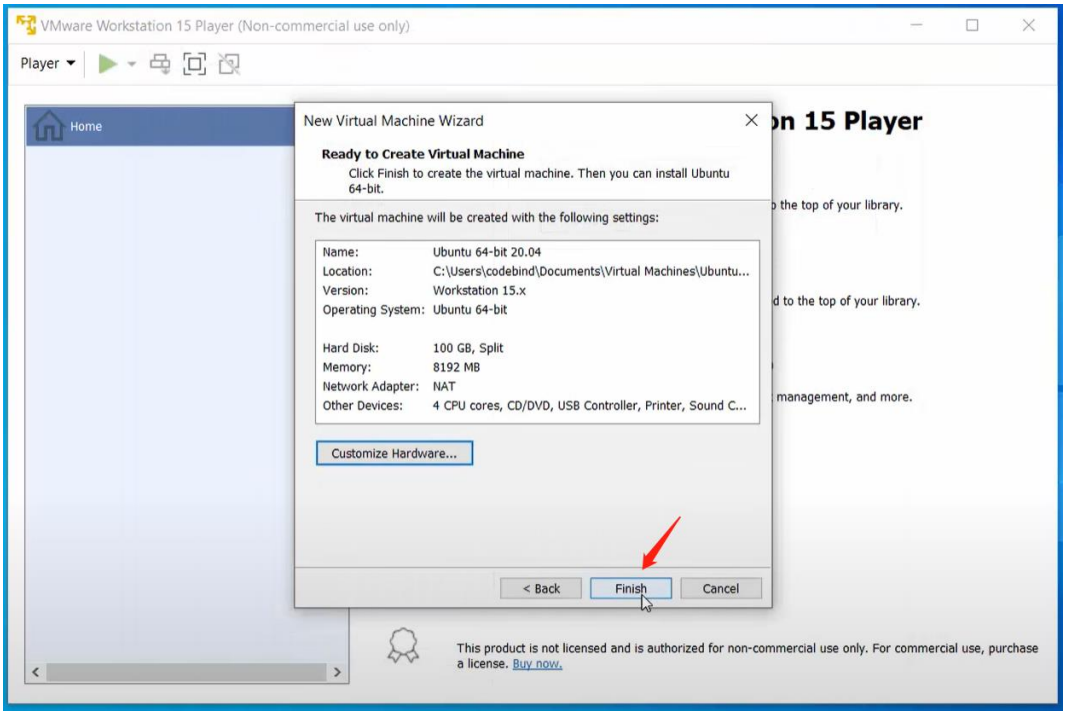


Find your ISO file in the ‘New CD/DVD(SATA)’ option

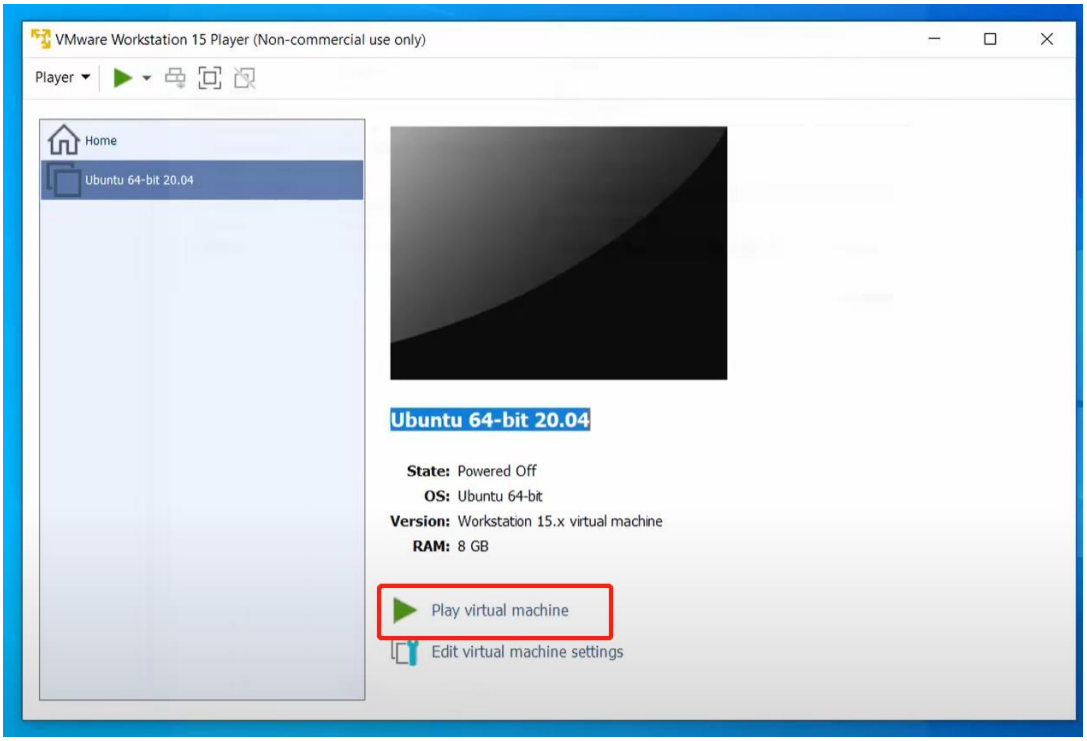




After the above configuration, leave the other Hardware keeping default and click ‘Close’ to return ‘New Virtual Machine Wizard’ and click ‘Finish’.

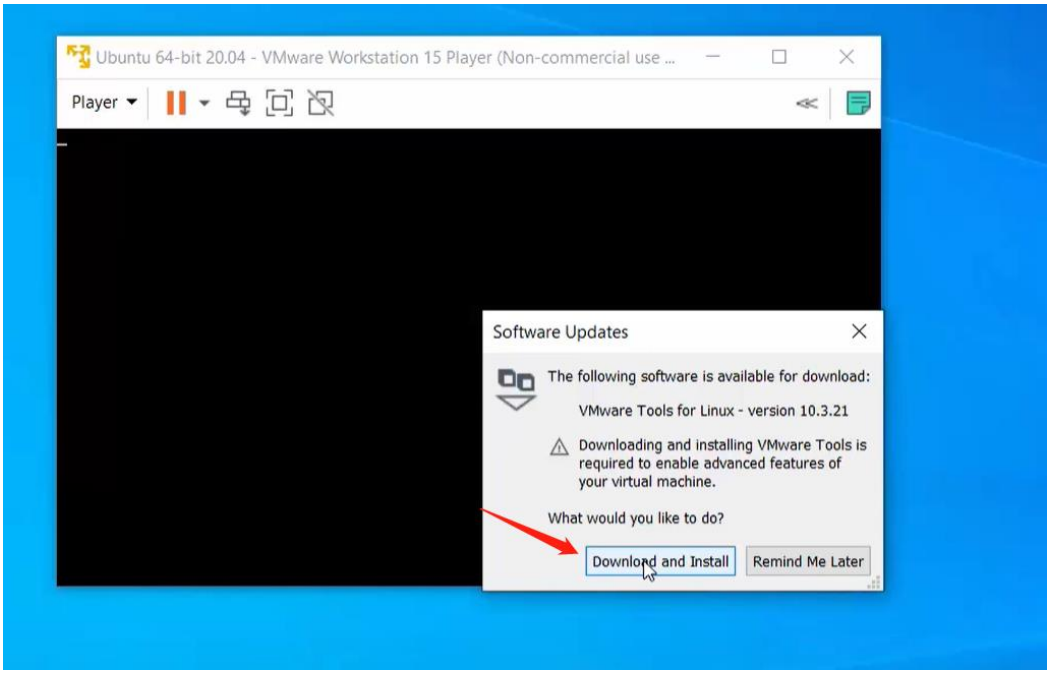


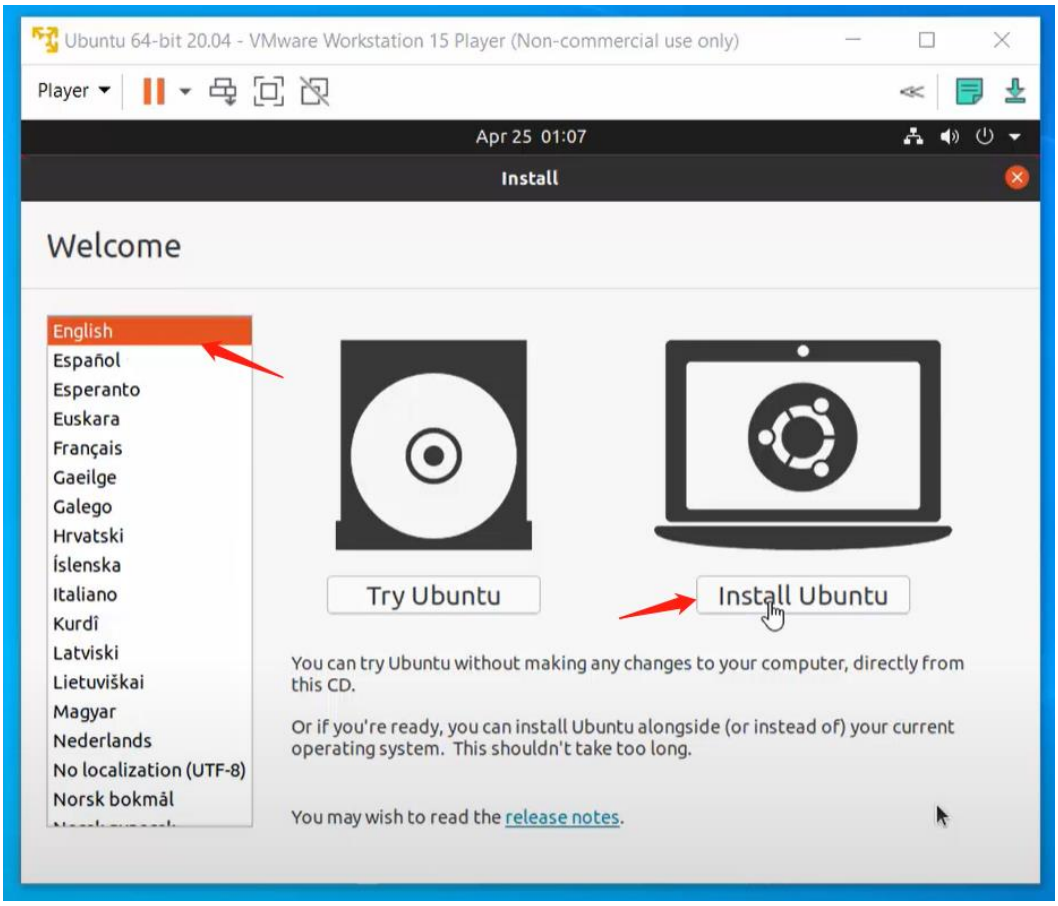
Then your Ubuntu Virtual Machine is created as the following figure shows. Click ‘Play virtual machine’ to finish Ubuntu Installation.

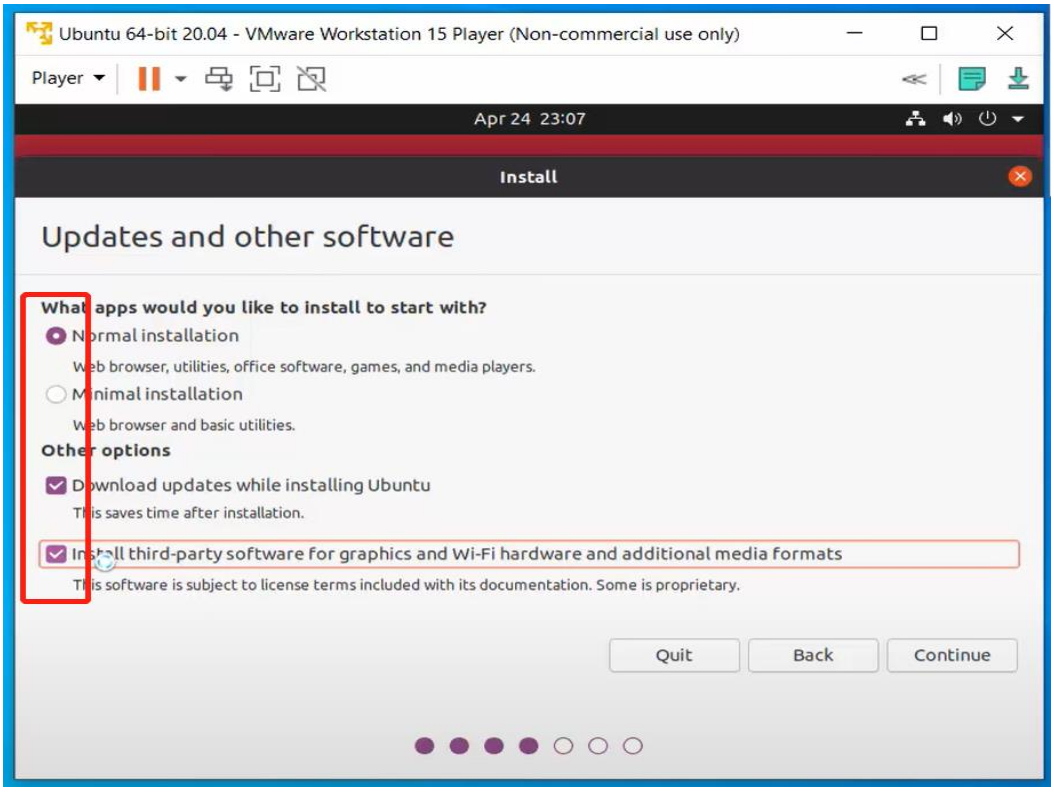


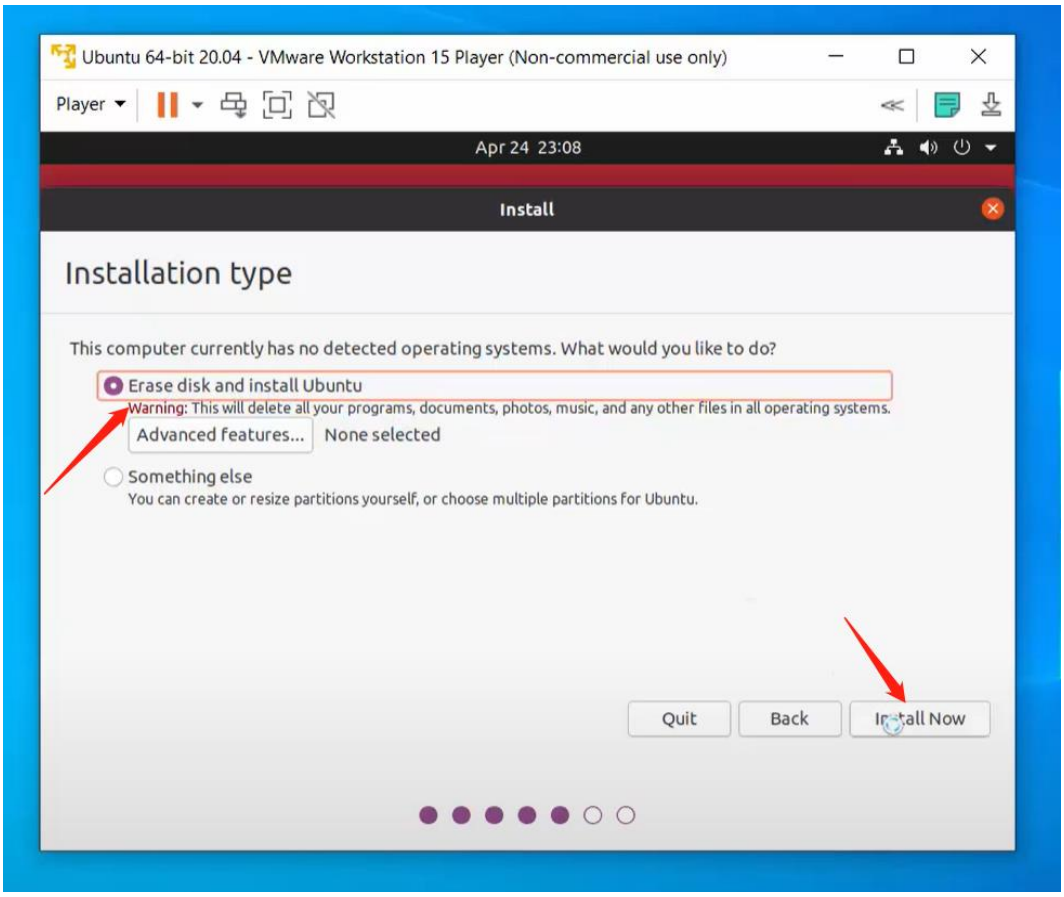
2.3 Install Ubuntu in Virtual Machine

- If you are get a reminder to update VMware, you can click ‘Download and Install’ to continue.

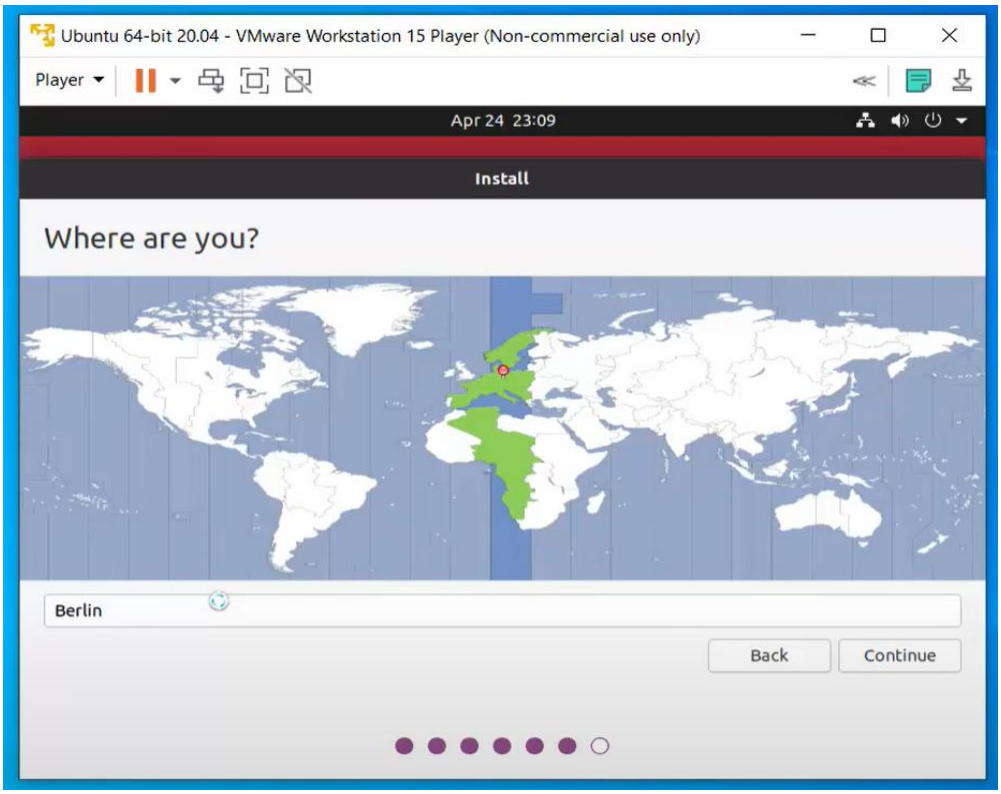

Ubuntu 64-bit 20.04 - VMware Workstation 15 Player (Non-commercial use ...
Player
Software Updates
The following software is available for download:
VMware Tools for Linux - version 10.3.21
Downloading and installing VMware Tools is required to enable advanced features of your virtual machine.
What would you like to do?
Download and Install
Remind Me Later


Ubuntu 64-bit 20.04 - VMware Workstation 15 Player (Non-commercial use only)
Player
Apr 25 01:07
Install
Welcome
English
Español
Esperanto
Euskara
Français
Gaeilge
Galego
Hrvatski
Íslenska
Italiano
Kurdî
Latviski
Lietuviškai
Magyar
Nederlands
No localization (UTF-8)
Norsk bokmål
Try Ubuntu
Install Ubuntu
You can try Ubuntu without making any changes to your computer, directly from this CD.
Or if you're ready, you can install Ubuntu alongside (or instead of) your current operating system. This shouldn't take too long.
You may wish to read the release notes.



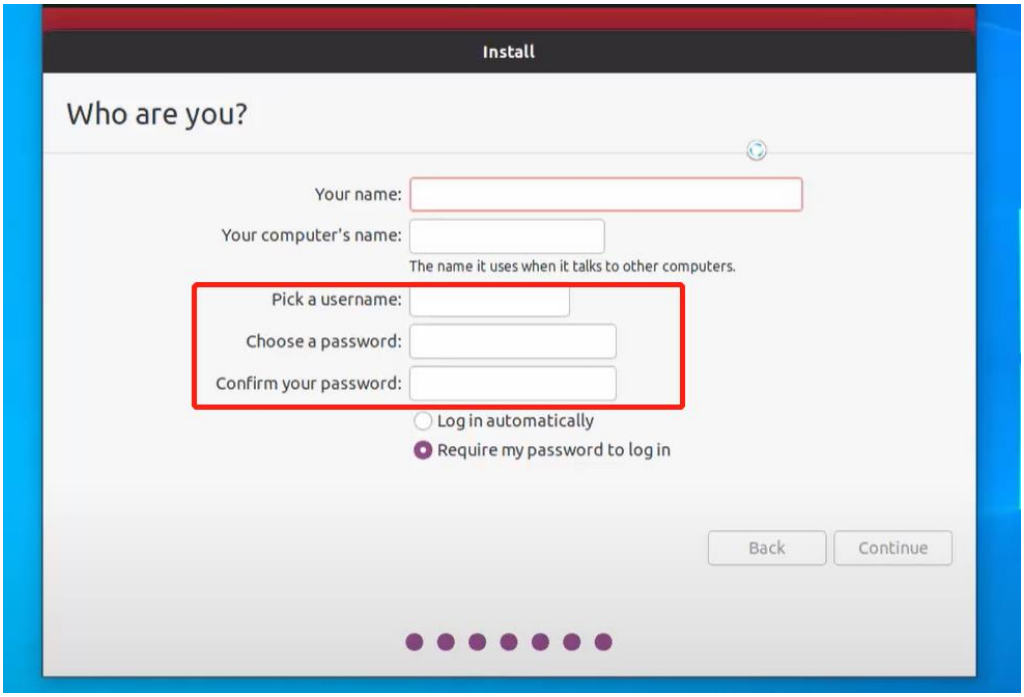


- Chose the location to ‘HongKong’

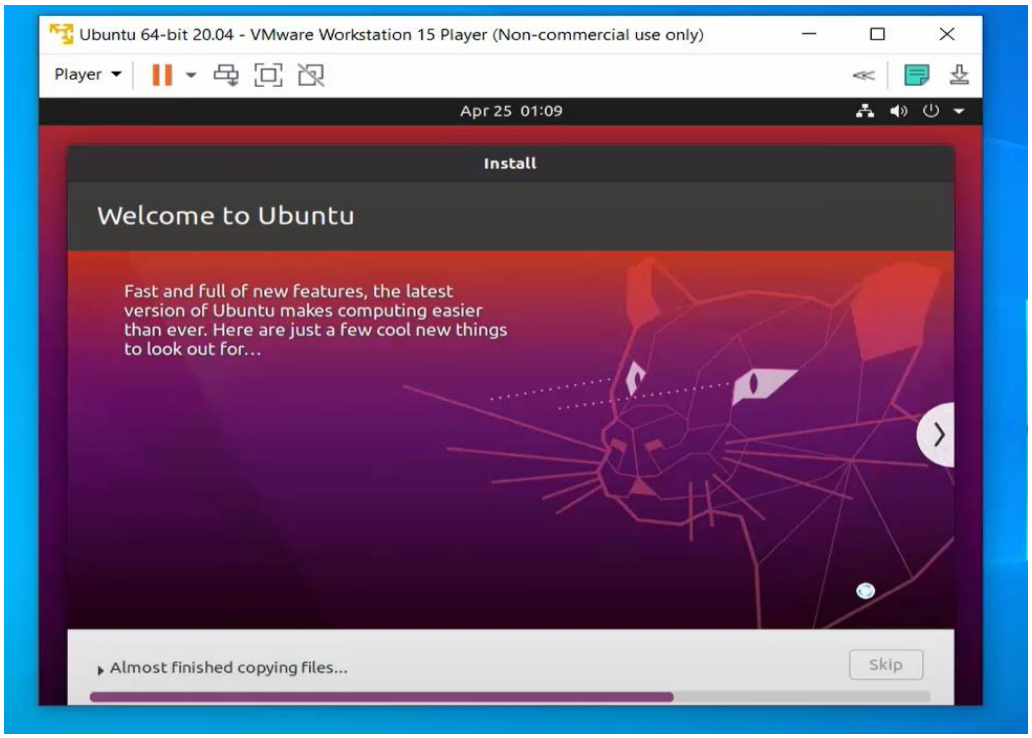


- Set the username and password for your Ubuntu System

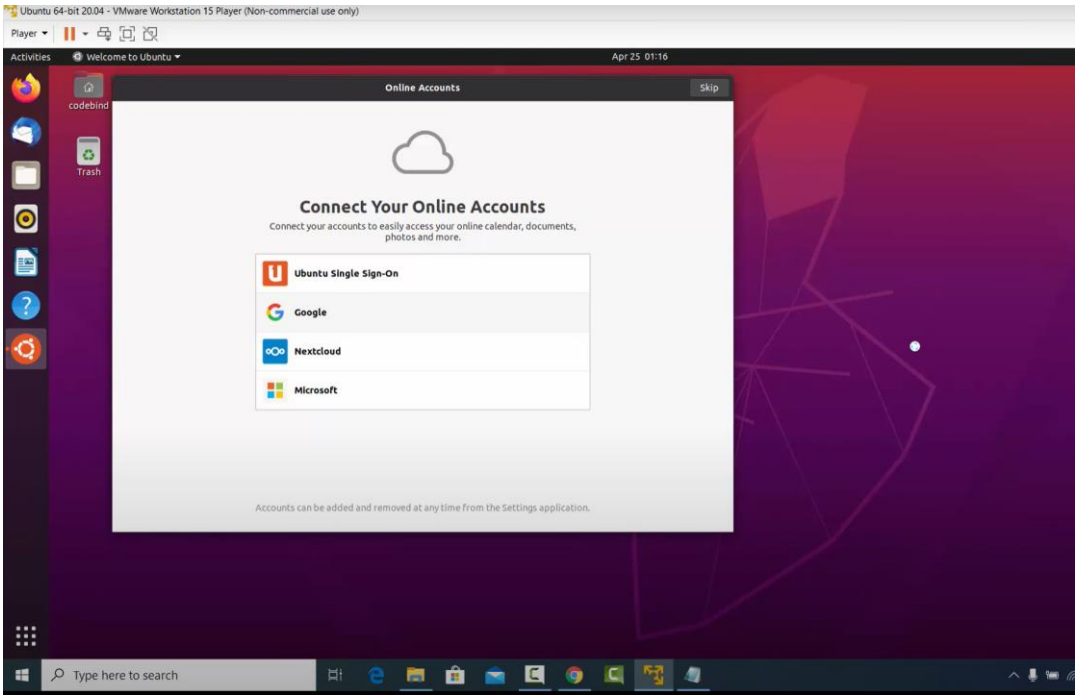
  Please do not forget the 'username' and 'password'.



- The ubuntu system starts to install automatically.



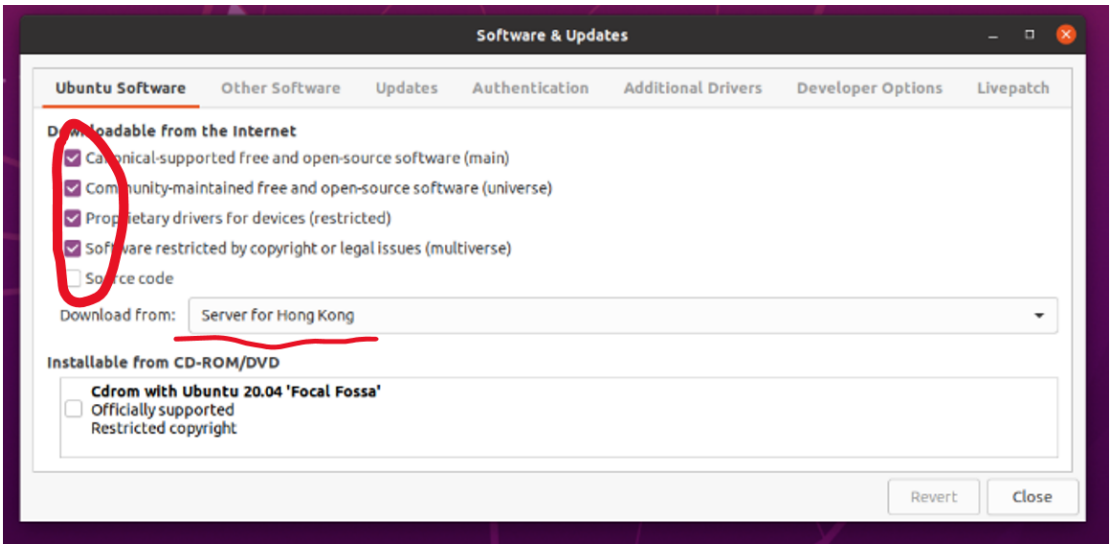
After the installation, you need to restart and re-enter the Virtual Machine. If you see the following figure, it means that the virtual machine is installed successfully.



2.4 Configure your Ubuntu repositories

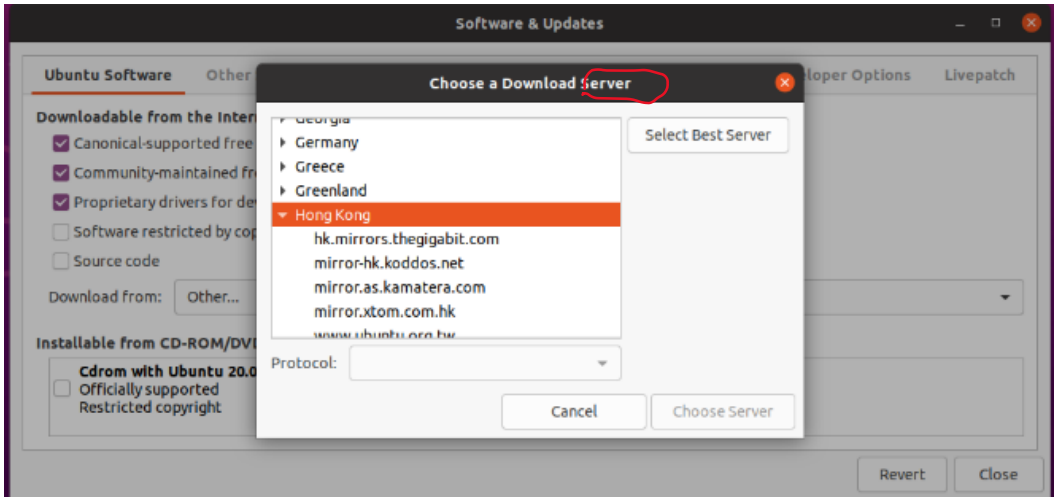
(1) enter Ubuntu ‘Software & Updates’ Tab.

(2) Configure your Ubuntu repositories to allow "restricted," "universe," and "multiverse."



(3) config the download source

Automatic configure: Click the drop-down menu of 'Download from' tab ---> 'Other..' ---> 'Select Best Server'



(4) set up your sources.list

It is also suggested to set up the source.list as follows:

- sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak
- sudo gedit /etc/apt/sources.list
- Append the following content into aources.list, do not delete the original content.

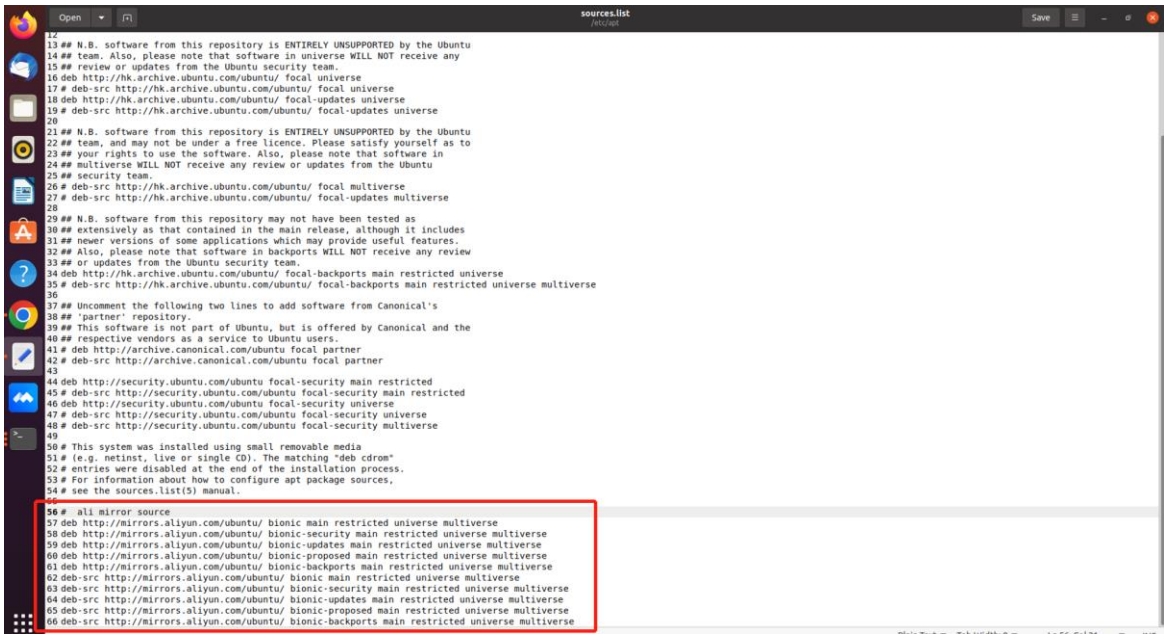
```
#    ali mirrior source
deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main restricted universe multiverse
```

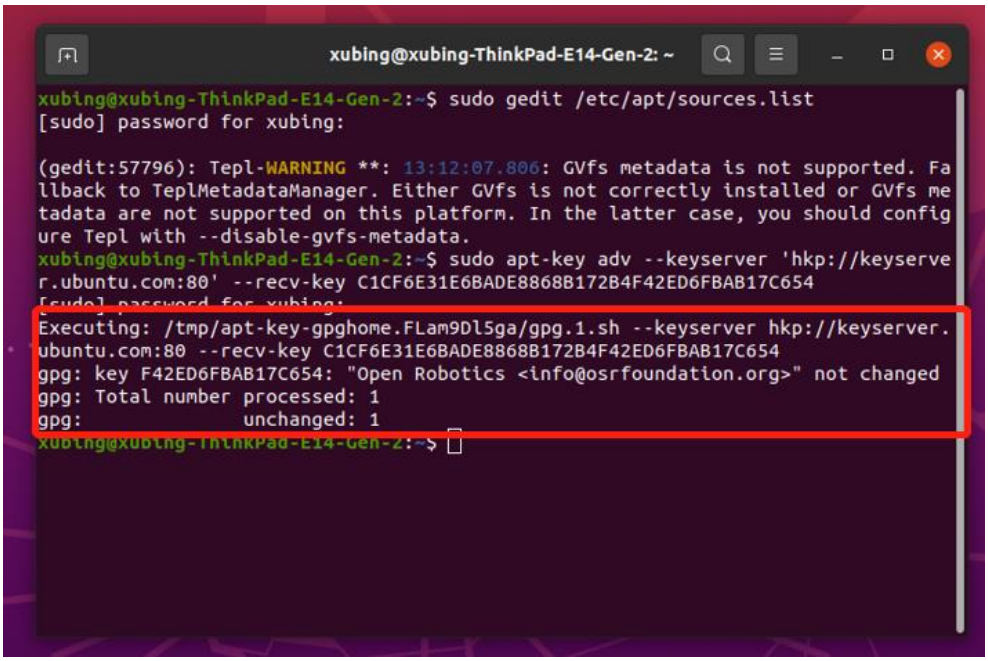
```
deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main restricted universe multiverse

deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main restricted universe multiverse
```



2.5 Set up your keys

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```



If you can not connect to the keyserver, you can change the above command as follows:

```
$ sudo apt-key adv --keyserver 'hkp://pgp.mit.edu:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

If you enable the proxy, you can use ‘curl’ instead of ‘apt-key’:

```
$ sudo apt install curl # if you haven't already installed curl
$ curl -sSL 'http://keyserver.ubuntu.com/pks/lookup?op=get&search=0xC1CF6E31E6BADE8868B172B4F42ED6FBAB17C654' | sudo apt-key add –
```

3. ROS Introduction

3.1 System structure

ROS is a software suite that allows for the quick and easy building of autonomous robotic systems. ROS should be considered as a set of tools for creating new solutions or adjusting already existing ones. A major advantage of this system is a great set of drivers and implemented algorithms widely used in robotics.

- Nodes

The base unit in ROS is called a node. Nodes are in charge of handling devices or computing algorithms - each node for a separate task. Nodes can communicate with each other using topics or services. ROS software is distributed in packages. A single package is usually developed for performing one type of task and can contain one or multiple nodes.

- Topics

In ROS, a topic is a data stream used to exchange information between nodes. They are used to send frequent messages of one type. This could be a sensor readout or motor goal speed. Each topic is registered under a unique name and with a defined message type. Nodes can connect with it to publish messages or subscribe to them. For a given topic, one node can not publish and subscribe to it at the same time, but there are no restrictions on the number of different nodes publishing or subscribing.

- Services

Communication by services resembles the client-server model. In this mode, one node (the server) registers service in the system. Later any other node can ask for that service and get a response. In contrast to topics, services allow for two-way communication, as a request can also contain some data.

3.2 Basic tools

While working with ROS some tools are most useful. They are intended to examine nodes and topics.

- rosnode

Rosnode is a command line application for examining which nodes are registered in the system and also checking their statuses.
Using the application looks as follows:

```
rosnode command [node_name]
```

The command could be:
list - display list of running nodes
info - display info regarding selected node
kill - stop selected node
Detailed info could be found in ROS documentation.

- rostopic

Rostopic is a command line application for examining which topics are already being published and subscribed, checking details of the selected topic or reading messages being sent in it.
Using the application looks as follows:

```
rostopic command [topic_name]
```

The command could be:
list - display list of topics
info - display info regarding selected topic
echo - display messages published in the topic
Detailed info could be found in ROS documentation.

- rqt_graph

rqt_graph is a graphical tool for visualization of data flow across different nodes in the system.
Using the application looks as follows:

```
rqt_graph
```

3.3 Supplementary Materials

(1) Programming for Robotics (ROS)
https://www.youtube.com/playlist?list=PLE-BQwvVGf8HOvwXPgtDfWoxd4Cc6ghiP
(2) ROS Introduction Book
https://husarion.com/tutorials/ros-tutorials/1-ros-introduction/

4. ROS Practice

4.1.1 ROS Install

The ROS is consistent with the release progress of Ubuntu. The ‘ROS Noetic (Ninjemys)’ is the

latest distribute ROS version, which can be run on Ubuntu 20.04.

4.1.1 Install ROS

```
$ sudo apt update
$ sudo apt install ros-noetic-desktop-full
$ sudo apt install ros-noetic-slam-gmapping
```

4.1.2 Environment setup

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

4.1.3 Dependencies Install & Initialization

```
$ sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool
build-essential
$ sudo apt install python3-rosdep
$ sudo rosdep init
$ rosdep update
```

4.2 First Demo & Test

4.2.1 Create a ROS Workspace

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
$ source devel/setup.bash
$ source devel/setup.bash
```

Use your username to replace the <username> below:

```
$ echo $ROS_PACKAGE_PATH /home/<username>/catkin_ws/src:/opt/ros/ noetic /share
```

For example:

```
xubing@xubing-ThinkPad-E14-Gen-2:~/catkin_ws$ echo $ROS_PACKAGE_PATH /home/xubing/catkin_ws/src:/opt/ros/ noetic /share
/opt/ros/noetic/share /home/xubing/catkin_ws/src:/opt/ros/ noetic /share
xubing@xubing-ThinkPad-E14-Gen-2:~/catkin_ws$
```

4.2.2 Test

(1) Open three terminals (CTRL+ALT+T),
In the first terminal, type in the following command:

```
$ roscore
```

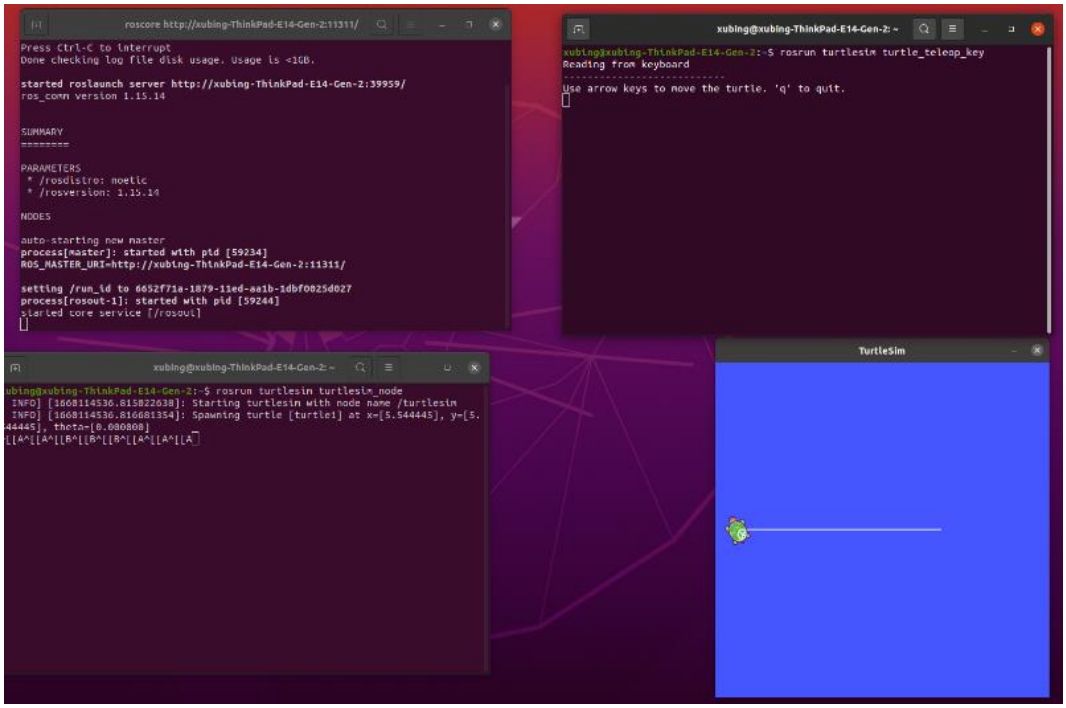
(2) In the second terminal, type in the following command:

```
$ rosrun turtlesim turtlesim_node
```

(3) In the third terminal, type in the following command:

```
$ rosrun turtlesim turtle_teleop_key
```

Keep the third terminal active, and you can control the turtlesim through keyboard by ←↑↓→



5. PID Basis

5.1 Introduction

- How Does a PID Controller Work?

A proportional integral derivative (PID) controller can be used as a means of controlling temperature, pressure, flow, and other process variables. As its name implies, a PID controller combines proportional control with additional integral and derivative adjustments which help the unit automatically compensate for changes in the system.

- PID Controller Basics

The purpose of a PID controller is to force feedback to match a setpoint, such as a thermostat that forces the heating and cooling unit to turn on or off based on a set temperature. PID controllers are best used in systems that have a relatively small mass and those which react quickly to changes in

the energy added to the process. It is recommended in systems where the load changes often and the controller is expected to compensate automatically due to frequent setpoint changes, the amount of energy available, or the mass to be controlled.

- PID Controller Working Principle

The working principle behind a PID controller is that the proportional, integral, and derivative terms must be individually adjusted or "tuned." Based on the difference between these values a correction factor is calculated and applied to the input. For example, if an oven is cooler than required, the heat will be increased. Here are the three steps:

Proportional tuning involves correcting a target proportional to the difference. Thus, the target value is never achieved because as the difference approaches zero, so too does the applied correction.

Integral tuning attempts to remedy this by effectively cumulating the error result from the "P" action to increase the correction factor. For example, if the oven remained below temperature, “I” would act to increase the head delivered. However, rather than stop heating when the target is reached, "I" attempts to drive the cumulative error to zero, resulting in an overshoot.

Derivative tuning attempts to minimize this overshoot by slowing the correction factor applied as the target is approached.

5.2 Position PID

Position PID is the algorithm typically used to perform closed-loop motion control on a position feedback axis. PID stands for the central gains used in this mode: Proportional, Integral, and Differential. The Position PID provides very good control and is suitable for nearly all motion control systems with position feedback.

- Algorithm

Each closed loop motion command issued to the RMC specifies a target profile, which defines where the axis should be at any given moment. For each loop time when the axis is in closed loop control, the RMC uses the specified target profile to calculate the desired position of the axis at that moment (called the Target Position) and subtracts the Actual Position to determine the Position Error. The Position PID algorithm then uses this information, together with the gains and feed forwards, to calculate how much Control Output should be generated to move the axis to the Target Position. The values of the gains and feed forwards must be set to achieve proper control. The process of setting the gains is called tuning and is done as part of the setup procedure.

The Position PID uses the gains and feeds forwards listed below. Each gain or feed-forward is multiplied by some quantity related to the Target Position and Actual Position to come up with a percentage. The resulting percentages are all summed and then multiplied by the maximum output (typically 10V), to come up with the Control Output voltage for that loop time.

- Proportional Gain

The Proportional Gain is multiplied by the Position Error. This is the most important gain.

- Integral Gain

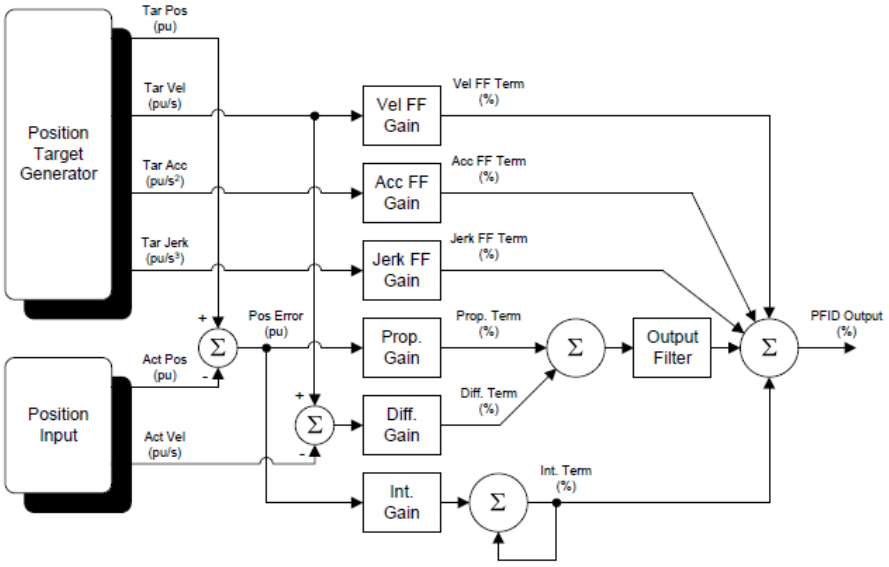
The Integral Gain is multiplied by the accumulated Position Error. This helps the axis get into position over time.

- Differential Gain

The Differential Gain is multiplied by the difference between the Target and Actual Velocities. This helps the axis keep up with quick changes in velocity.

- Velocity Feed Forward
  The Velocity Feed Forward is multiplied by the Target Velocity.
- Acceleration Feed Forward
  The Acceleration Feed Forward is multiplied by the Target Acceleration.
- Jerk Feed Forward
  The Jerk Feed Forward is multiplied by the Target Jerk. The Jerk Feed Forward is not necessary for most applications.

In addition, higher-order gains may be used if Acceleration Control or Active Damping are selected.



5.3 Position PID Code

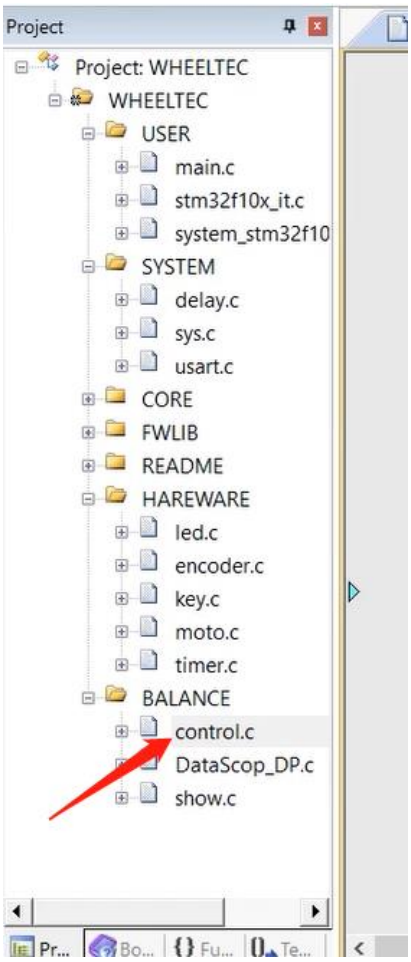
```
int Position_PID (int position,int target)
{
      static float Bias,Pwm,Integral_bias,Last_Bias;
      Bias=target-position;
      Integral_bias+=Bias;
     if(Integral_bias>3000)Integral_bias=3000;
     if(Integral_bias<-3000)Integral_bias=-3000;
      Pwm=Position_KP*Bias+Position_KI*Integral_bias+Position_KD*(Bias-Last_Bias);
      Last_Bias=Bias;
      return Pwm;
}
```

The inlet parameter is the measured value of the encoder position and the target value of the position control, and the return value is the motor control PWM (now look at the above control block diagram to see if it is easier to understand).

The 1st line is the definition of relevant internal variables.

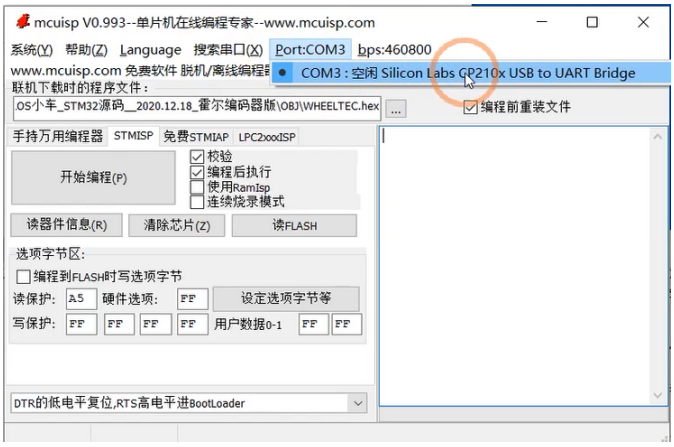
The 2nd line is to find the position deviation and subtract the target value from the measured value.

The 3rd line calculates the integral of the deviation by accumulation.
The 4th line uses the position PID controller to calculate the motor PWM.
The 5th line saves the last deviation for the next call.
The last line is returned.
You can find the code as shown in the following figure.

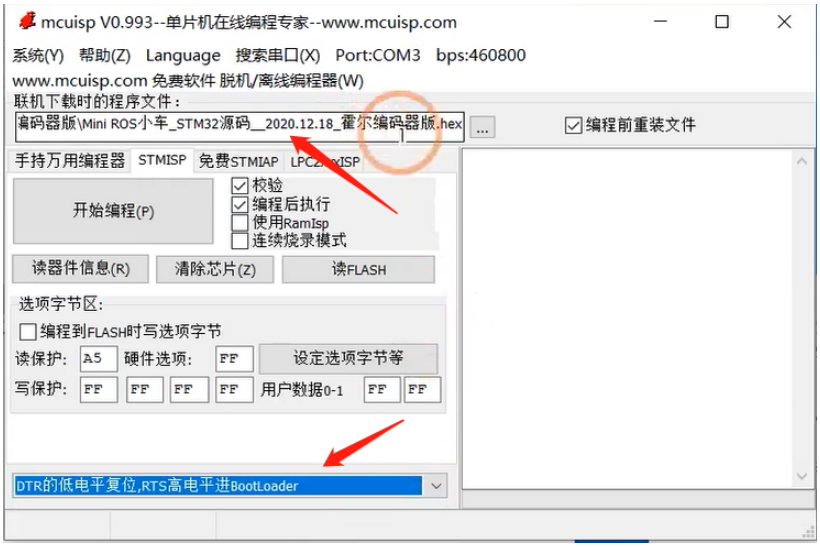


5.3 STM32 Program Download via USB

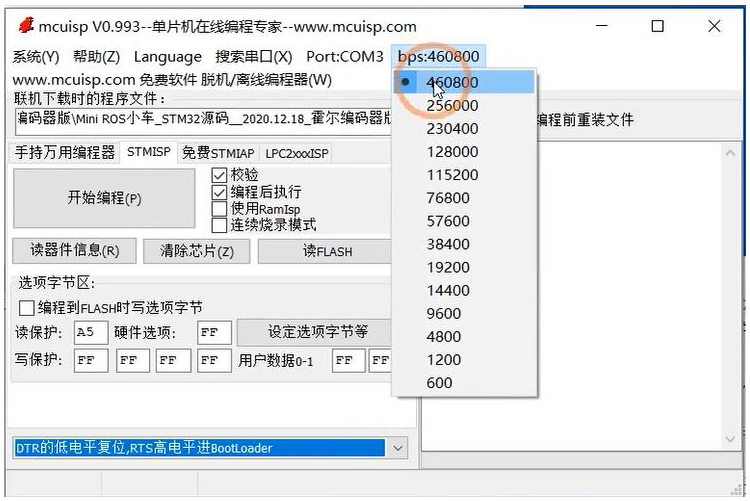
- Connect your computer and USB interface of the STM32 via a cable
- Chose the active serial port in ‘mcuisp’



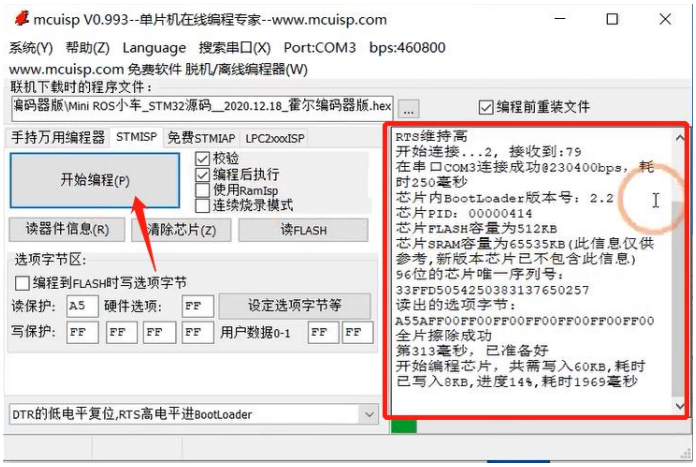
- Choose the .hex file which is have been compiled and need to download into STM32. Choose the download mode at the bottom of mcuisp. Ensure the download mode keep consistent with the below figure.



- Choose the highest baud rate '460800'



- Click the button indicated by the arrow in the following picture. The code download log can be found on the right of the dialog.



6. Robot Car

6.1 Connect with Robot Car

(1) Power on the Robert car

(2) Connect the WIFI on the Robot Car

WIFI Name: WHEELTECXX_XX

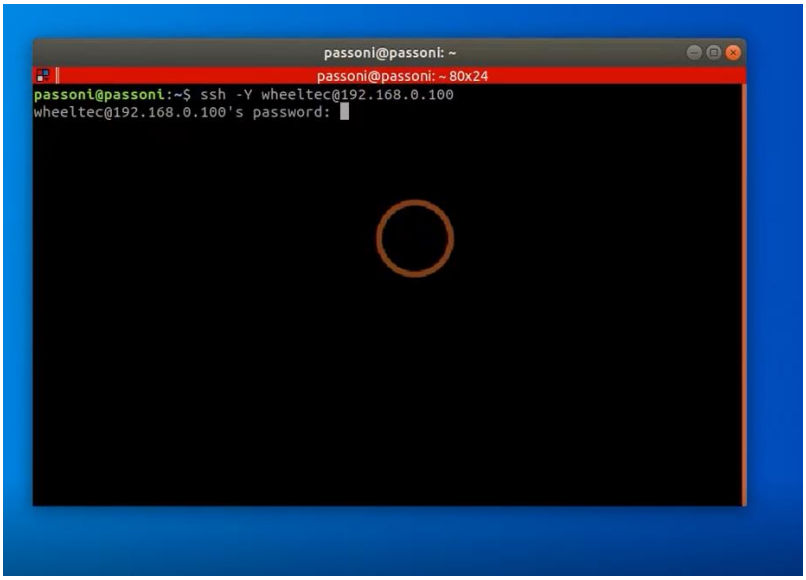
WIFI Password: dongguan

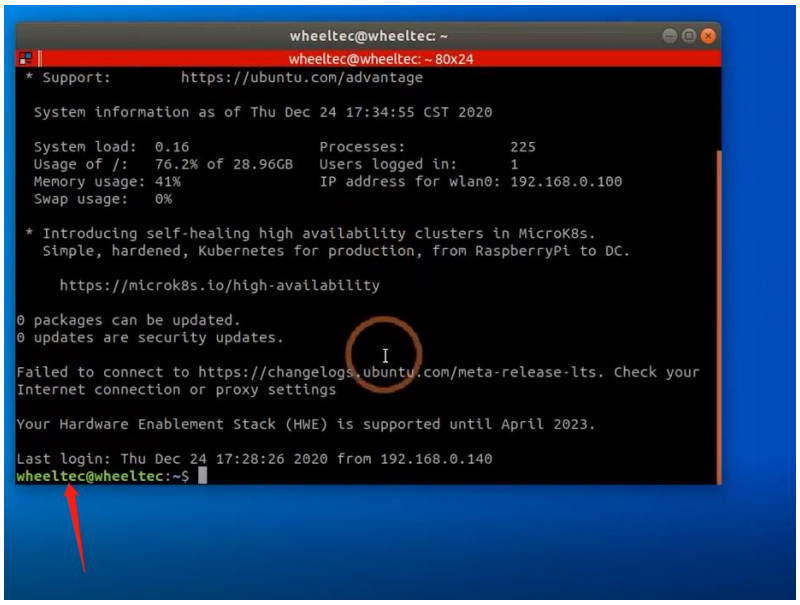
6.2 Remote Login by SSH

(1)SSH Login

- Secure shell (SSH) is a secure network protocol developed by IETF (the Internet Engineering Task Force) based on the application layer.
- We are using the free and open-source implementation of SSH in Ubuntu. It is mainly used to facilitate the remote use of the ROS car.
- In general, the SSH client is automatically installed in Ubuntu by default.
- The server needs to install by the following command:

```
 $ sudo apt-get install openssh-server
$ ssh -Y wheeltec@192.168.0.100
```



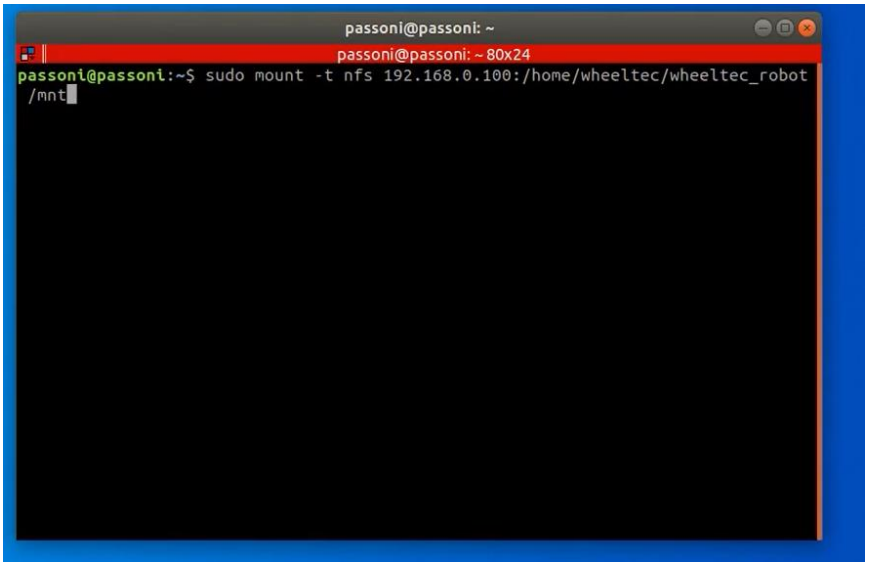


Now, if you type in a command in the terminal in ubuntu, the command will execute on the ROS robot car.

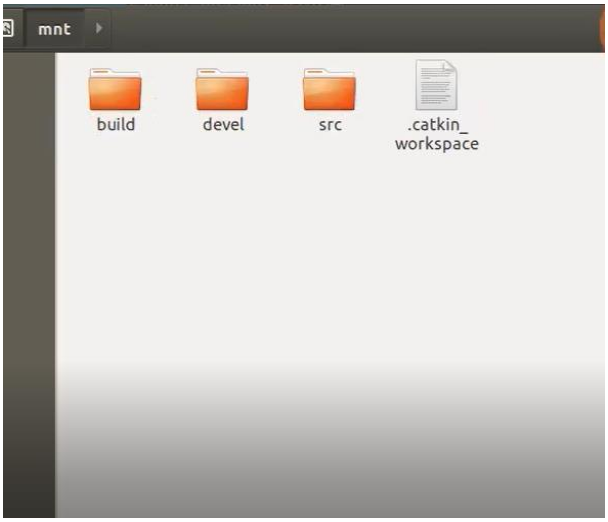
6.3 Robot Car Program Mount and Compile

6.3.1 Mount the Robot Car to Ubuntu

```
$ sudo mount -t nfs 192.168.0.100:/home/wheeltec/wheeltec_robot /mnt
```



If the source on the Robot car is successfully mounted, you can find and check the ROS workspace in directory /mnt on your Ubuntu system.



6.3.2 Modify the System time

To ensure the normal compilation of program modification, the system time of the Robot Car should be changed to the current time of your computer. The command is:

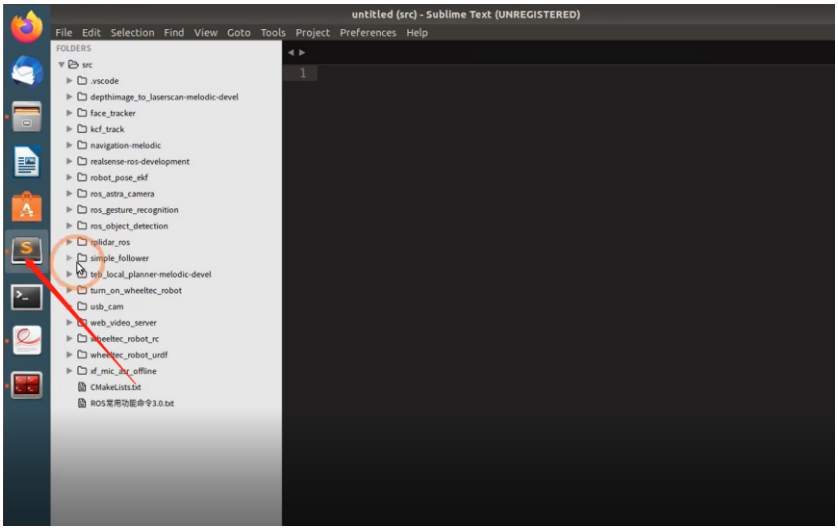
```
$ sudo date -s "xxxx-xx-xx xx:xx:00"
```

The time format is YYYY-MM-DD HH:MM:SS

6.3.3 Install Sublime and open ROS code

```
$ sudo add-apt-repository ppa:webupd8team/sublime-text-3
$ sudo apt-get update
$ sudo apt-get install sublime-text
```

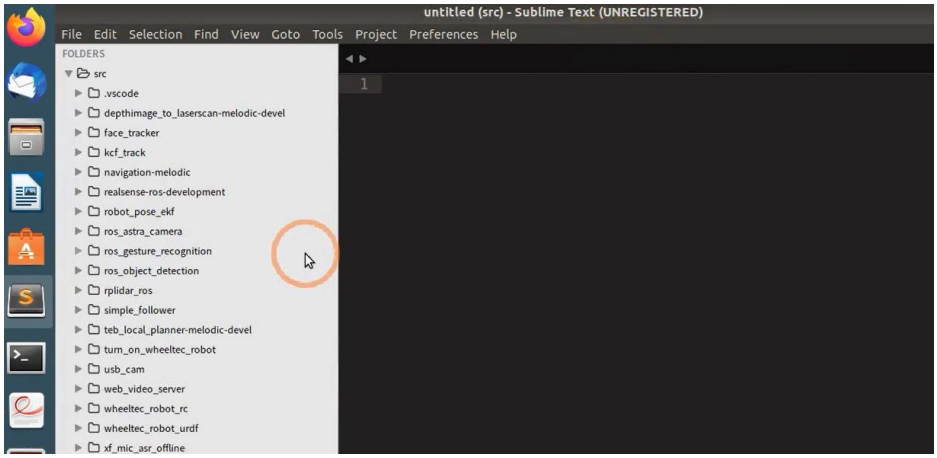
- After the installation, you can find the icon of sublimeText in your Ubuntu and open it.



- Open the mounted file

  In SublimeText, you can open the mounted file source by the menu on the GUI as follows:

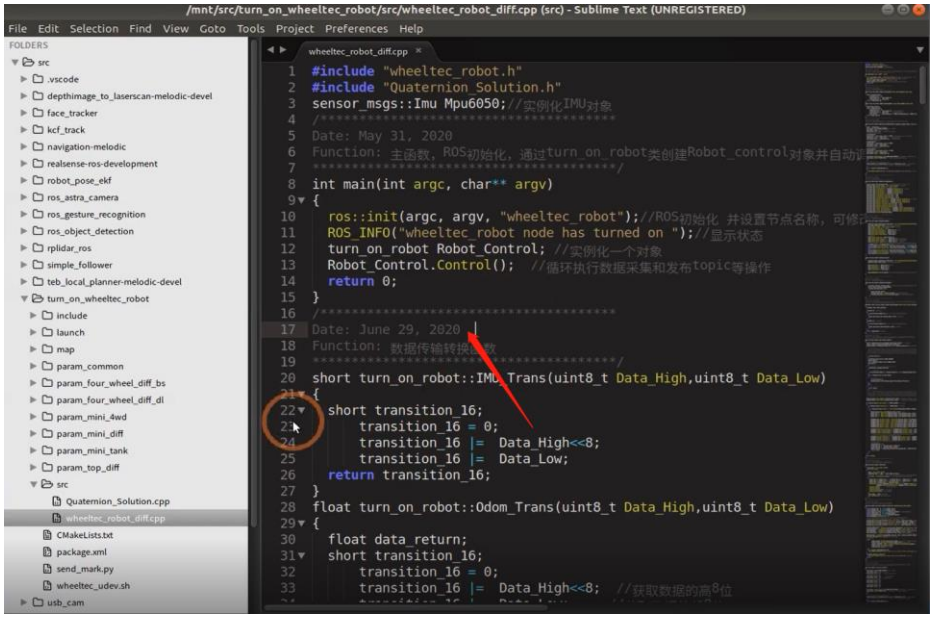
  “File” →”Select Folder” →find the ubuntu directory ‘/mnt’ →open the folder ‘/src’



6.3.4 ROS Code Modification and Compile

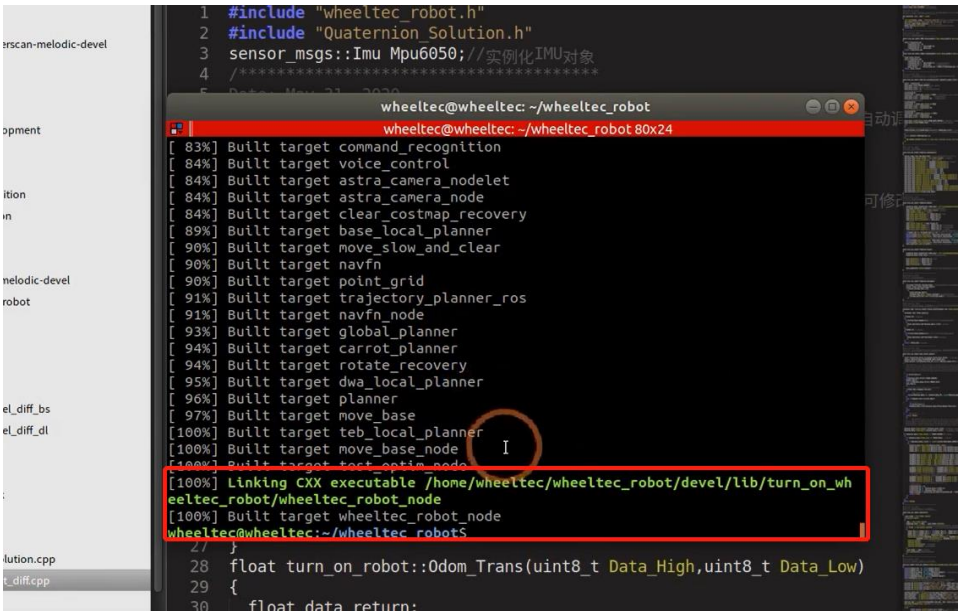
Here we show an example of how to check, modify and compile the ROS code.

- Open the catkin package ‘turn_on_wheeltec_robot’
- Find the cpp code in the directory ‘/src’
- Make a small change in line 17, here we just type in some spaces. And save the file by ‘Ctrl+S’.



- Open your ssh terminal and enter the ROS workspace and compile the ROS code

```
$ cd /home/wheeltec/wheeltec_robot/
$ catkin_make
```



If the compilation is completed without bugs, you can find the above log in the terminal.

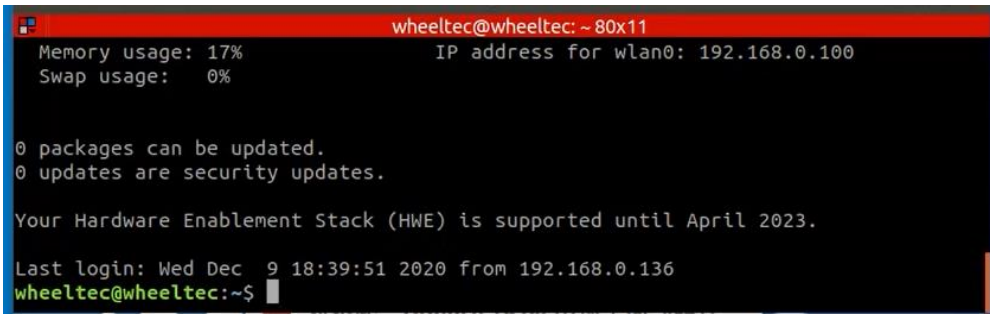
6.4 Control the Robot by Keyboard

6.4.1 The Initialization Node

This node receives the speed command and controls the movement of the Robot car.

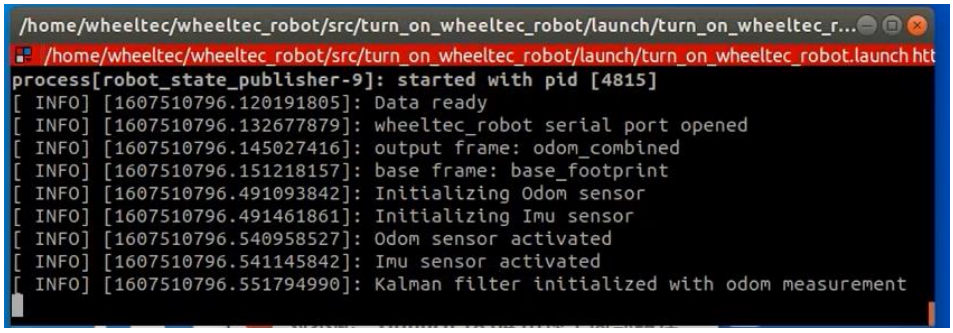
- Open a new terminal (CTRL+ALT+T)
- Login to the Robot car by SSH

```
$ ssh -Y wheeltec@192.168.0.100
```



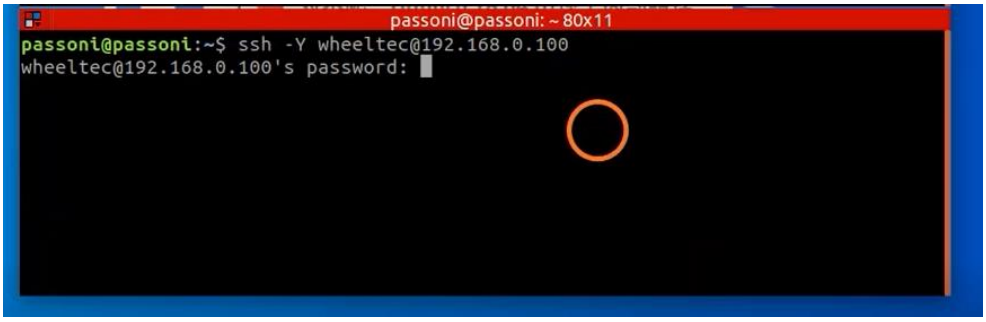
- Enable the node by ROS command

```
$ roslaunch turn_on_wheeltec_robot turn_on_wheeltec_robot.launch
```



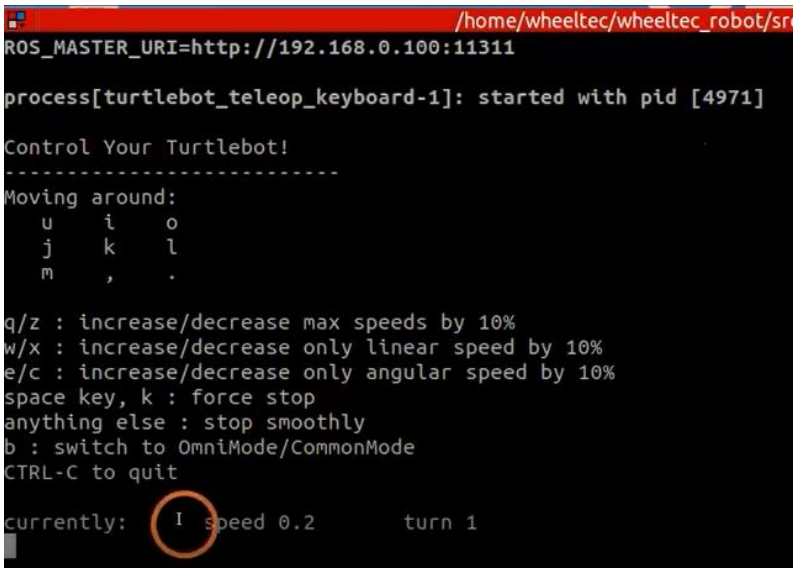
6.4.2 The Keyboard Control Node

- Open another new terminal (CTRL+ALT+T)
- Login to the Robot car by SSH



- Enable the node by ROS command

```
$ roslaunch wheeltec_robot_rc keyboard_teleop.launch
```



The 'Moving around' log in the terminal give the button on the keyboard, and each button relates to one direction.

" i " :↑

" , " :↓

" j " : ←

" l " : →

" u " : ↖

" o " : ↗

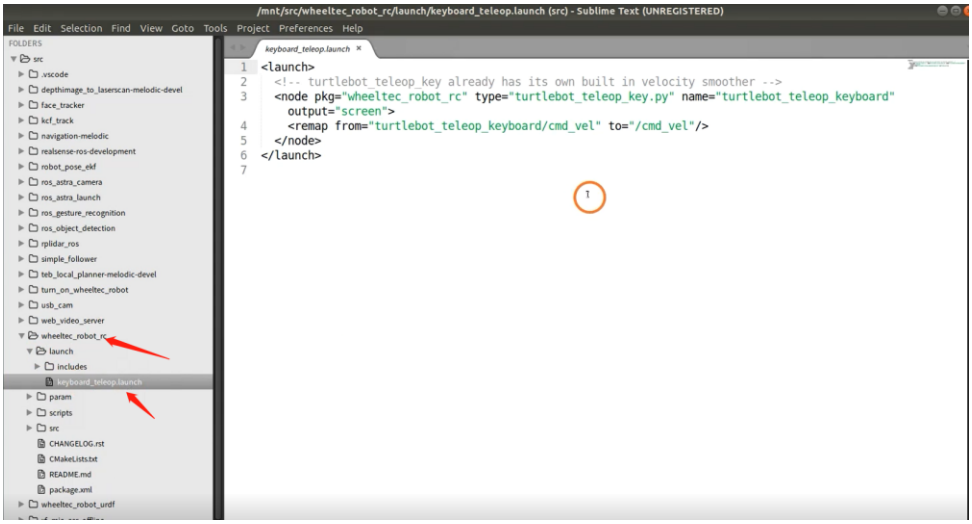
" m " : ↙

" . " : ↘

" k " : emergency brake

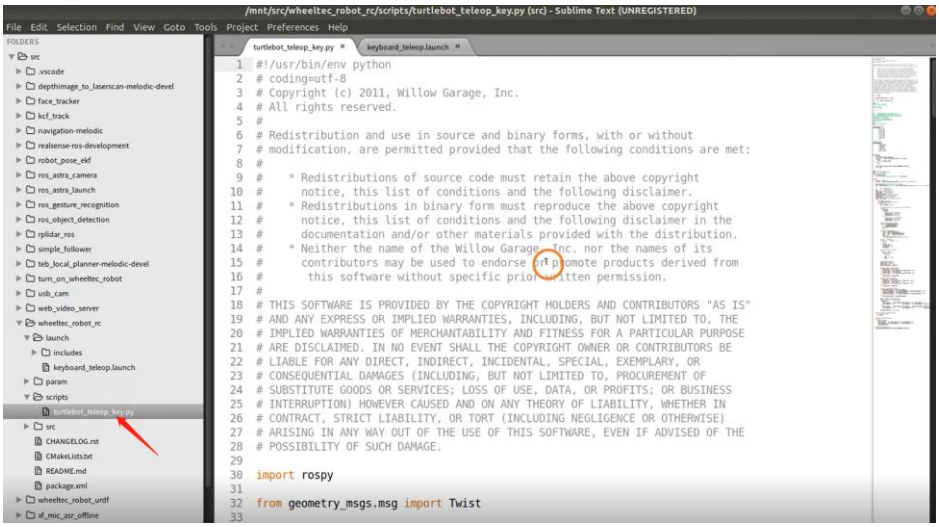
6.4.3 The code of the Keyboard Control Function

- Check the .launch



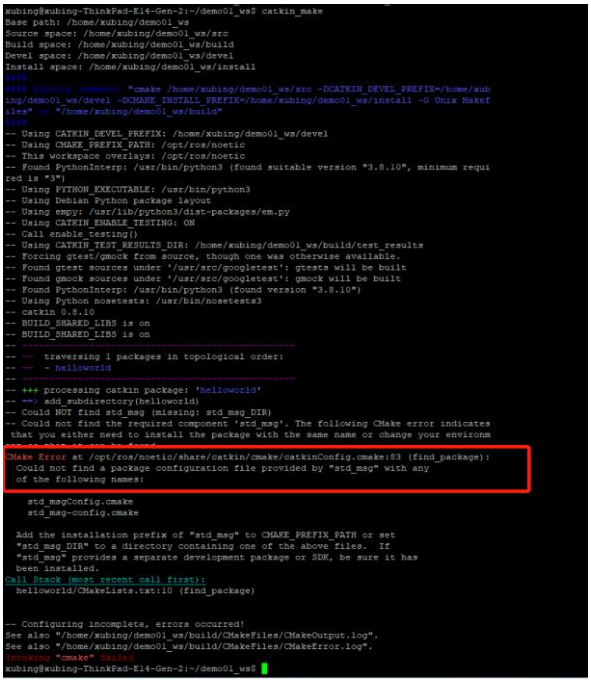
It tells that the code is saved in .py file 'turtlebot_teleop_key.py', and the topic is renamed as '/com_vel'

- Open the .py source code
  A detailed commented file can be found in Appendix A.



Bug ListReference

(1) Could not find a package configuration file provided by "xxx" with any of the following names:



Solution: try to install the missing package use the following command

```
sudo apt-get install <ros version>-xxx
```

For the example here, the below command should be executed:

```
sudo apt-get install noetic std_msg
```

Reference

http://wiki.ros.org/
http://wiki.ros.org/cn

Appendix A Detailed Comment of turtlebot_teleop_key.py

```
#!/usr/bin/env python
# coding=utf-8
# Copyright (c) 2011, Willow Garage, Inc.
# All rights reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions are met:
#
#       * Redistributions of source code must retain the above copyright
#           notice, this list of conditions and the following disclaimer.
#       * Redistributions in binary form must reproduce the above copyright
#           notice, this list of conditions and the following disclaimer in the
#           documentation and/or other materials provided with the distribution.
#       * Neither the name of the Willow Garage, Inc. nor the names of its
#           contributors may be used to endorse or promote products derived from
#             this software without specific prior written permission.
#
# THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
# AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
# IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
# ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
# LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
```

```
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
# SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
# INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
# CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
# ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
# POSSIBILITY OF SUCH DAMAGE.

import rospy

from geometry_msgs.msg import Twist

import sys, select, termios, tty

msg = """
Control Your Turtlebot!
---------------------------
Moving around:
   u    i    o
   j    k    l
   m    ,    .

q/z : increase/decrease max speeds by 10%
w/x : increase/decrease only linear speed by 10%
e/c : increase/decrease only angular speed by 10%
space key, k : force stop
anything else : stop smoothly
```

```
b : switch to OmniMode/CommonMode
CTRL-C to quit
"""
Omni = 0 #Omnidirectional movement mode

#The key-value corresponds to the movement / steering direction
moveBindings = {
        'i':( 1, 0),
        'o':( 1,-1),
        'j':( 0, 1),
        'l':( 0,-1),
        'u':( 1, 1),
        ',':(-1, 0),
        '.':(-1, 1),
        'm':(-1,-1),
           }

#the value corresponds to speed increment
speedBindings={
        'q':(1.1,1.1),
        'z':(0.9,0.9),
        'w':(1.1,1),
        'x':(0.9,1),
        'e':(1,    1.1),
        'c':(1,    0.9),
          }
```

```
#this function get value by key
def getKey():
        tty.setraw(sys.stdin.fileno())
        rlist, _, _ = select.select([sys.stdin], [], [], 0.1)
        if rlist:
                key = sys.stdin.read(1)
        else:
                key = ''

        termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
        return key

speed = 0.2 #the default moving speed m/s
turn    = 0.5      #the default steering speed rad/s
#Returns the current speed in string format
def vels(speed,turn):
        return "currently:\tspeed %s\tturn %s " % (speed,turn)

#main function
if __name__=="__main__":
        settings = termios.tcgetattr(sys.stdin) #Get key value initialization and read terminal related properties

        rospy.init_node('turtlebot_teleop') #crate a ROS node
        pub = rospy.Publisher('~cmd_vel', Twist, queue_size=5) #Create speed topic publisher，'~cmd_vel'='<node name>/cmd_vel'
```

```
        x           = 0     #Forward backward direction
        th          = 0     #Steering / lateral movement direction
        count       = 0     #
        target_speed = 0 #Forward / backward target speed
        target_turn    = 0 #Steering target speed
        target_HorizonMove = 0 #Lateral moving target speed
        control_speed = 0 #Forward and reverse actual control speed
        control_turn    = 0 #Steering actual control speed
        control_HorizonMove = 0 #Lateral movement actual control speed
        try:
            print(msg) #Print control instructions
            print(vels(speed,turn)) #Print current speed
            while(1):
                key = getKey() #get key

                #Switch whether it is the omni-directional movement mode. The omni-directional wheel / wheat wheel trolley can join the omni-directional
movement mode
                if key=='b':
                    Omni=~Omni
                    if Omni:
                        print("Switch to OmniMode")
                        moveBindings['.']=[-1,-1]
                        moveBindings['m']=[-1, 1]
                    else:
                        print("Switch to CommonMode")
                        moveBindings['.']=[-1, 1]
```

```
            moveBindings['m']=[-1,-1]

#Determine whether the key value is within the movement / steering direction key value
if key in moveBindings.keys():
    x    = moveBindings[key][0]
    th = moveBindings[key][1]
    count = 0

#Determine whether the key value is within the speed increment key value
elif key in speedBindings.keys():
    speed = speed * speedBindings[key][0]
    turn    = turn    * speedBindings[key][1]
    count = 0
    print(vels(speed,turn)) #Speed changes, log it out to the terminal

#Empty key value /'k ', relevant variable set to 0
elif key == ' ' or key == 'k' :
    x    = 0
    th = 0
    control_speed = 0
    control_turn    = 0
    HorizonMove     = 0

#Unknown key value is recognized for a long time, and relevant variables are set to 0
else:
    count = count + 1
```

```
        if count > 4:
            x    = 0
            th = 0
        if (key == '\x03'):
            break

#Calculate target speed according to speed and direction
target_speed = speed * x
target_turn    = turn * th
target_HorizonMove = speed*th

#Smooth control, calculate the actual control speed of forward and backward
if target_speed > control_speed:
        control_speed = min( target_speed, control_speed + 0.1 )
elif target_speed < control_speed:
        control_speed = max( target_speed, control_speed - 0.1 )
else:
        control_speed = target_speed

#Smooth control, calculate the actual steering control speed
if target_turn > control_turn:
        control_turn = min( target_turn, control_turn + 0.5 )
elif target_turn < control_turn:
        control_turn = max( target_turn, control_turn - 0.5 )
else:
        control_turn = target_turn
```

```
            #Smooth control, calculate the actual control speed of lateral movement
            if target_HorizonMove > control_HorizonMove:
                control_HorizonMove = min( target_HorizonMove, control_HorizonMove + 0.1 )
            elif target_HorizonMove < control_HorizonMove:
                control_HorizonMove = max( target_HorizonMove, control_HorizonMove - 0.1 )
            else:
                control_HorizonMove = target_HorizonMove

            twist = Twist() #Create ROS speed topic variable
            #Assign a value to the speed topic variable according to whether the omni-directional movement mode is available
            if Omni==0:
                twist.linear.x  = control_speed; twist.linear.y = 0;  twist.linear.z = 0
                twist.angular.x = 0;                  twist.angular.y = 0; twist.angular.z = control_turn
            else:
                twist.linear.x  = control_speed; twist.linear.y = control_HorizonMove; twist.linear.z = 0
                twist.angular.x = 0;                  twist.angular.y = 0;                  twist.angular.z = 0

            pub.publish(twist) #ROS release speed topic

        #If there is a problem in the operation, the program terminates and prints the relevant error information
        except Exception as e:
            print(e)

        #Before the end of the program, release the speed topic with the speed of 0
        finally:
```

```
        twist = Twist()
        twist.linear.x = 0;    twist.linear.y = 0;    twist.linear.z = 0
        twist.angular.x = 0; twist.angular.y = 0; twist.angular.z = 0
        pub.publish(twist)

#Set the terminal related properties before the program ends
termios.tcsetattr(sys.stdin, termios.TCSADRAIN, settings)
```