Operating System Principle

Lab Course Instruction Book
Version 0.1

Hengjun ZHAO
School of Computer and Information Science, Southwest University, China

Contents

List of Figures

List of Tables

Chapter 1

Basic Operations and C Programming in Linux

Throughout this lab course, we will experiment with Linux[1] to deepen your understanding of the basic concepts and principles of operating systems we have learned in the theory classes. There are many popular distributions of Linux, e.g., Debian, Fedora, Ubuntu, Red Hat, etc. In this course, all labs will be carried out in the Ubuntu environment for no other reason but that it is simple enough for beginners. In chapter 1, we will learn

- how to install Ubuntu in a virtual machine;
- the basic commands for operating Linux system;
- how to do C programming in Linux system, and how to manage a project with the assistance of the `make` tool;
- how to work with kernel modules in Linux.

1.1 Install Ubuntu in Virtualbox

1.1.1 Preparations

We will install the Ubuntu system as a virtual machine in the `VirtualBox` manager. It is assumed that you have made the following preparations:

i. Download the latest long-term-support (LTS) version of Ubuntu at `https://ubuntu.com/download/desktop`. Up to March 9, 2021, it is `Ubuntu 20.04.2.0 LTS`.

[1] `https://en.wikipedia.org/wiki/Linux`

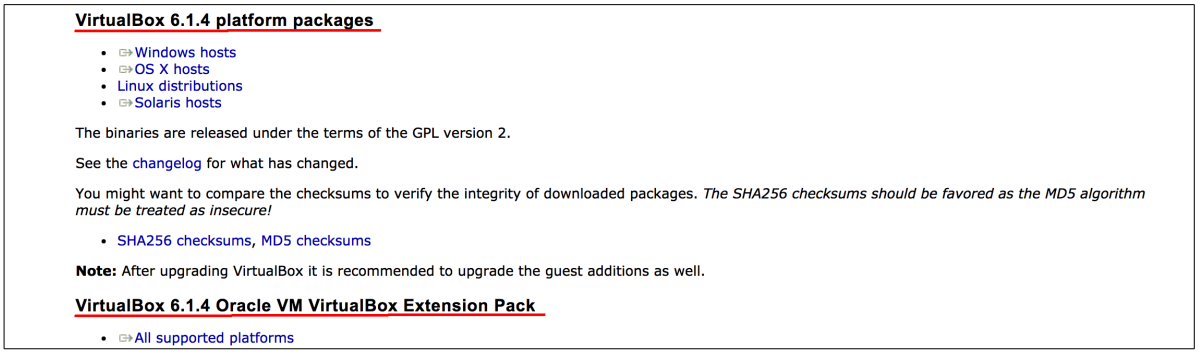

Figure 1.1: Download VirtualBox and its extension pack

Figure 1.2: Create a new machine in VirtualBox

ii. Download the latest version of VirtualBox, a freely available virtual machine manager, as well as its Extension Pack for supporting USB 2.0/3.0 devices, at `https://www.virtualbox.org/wiki/Downloads`, cf. Fig. 1.1. As of March 9, 2021, the latest version is `VirtualBox 6.1.18`.

iii. Install VirtualBox and its extension pack on your computer.

1.1.2 Create a Virtual Machine in VirtualBox

Now we show the process of creating a bare machine in VirtualBox.

i. First, start VirtualBox, and then click the `new` icon, cf. Fig. 1.2;

ii. Next, name your system and choose the right type and version, and then allocate virtual main memory for your system, cf. Fig. 1.3 (left);

iii. In the following, allocate virtual disk space for your system, cf. Fig. 1.3 (right).

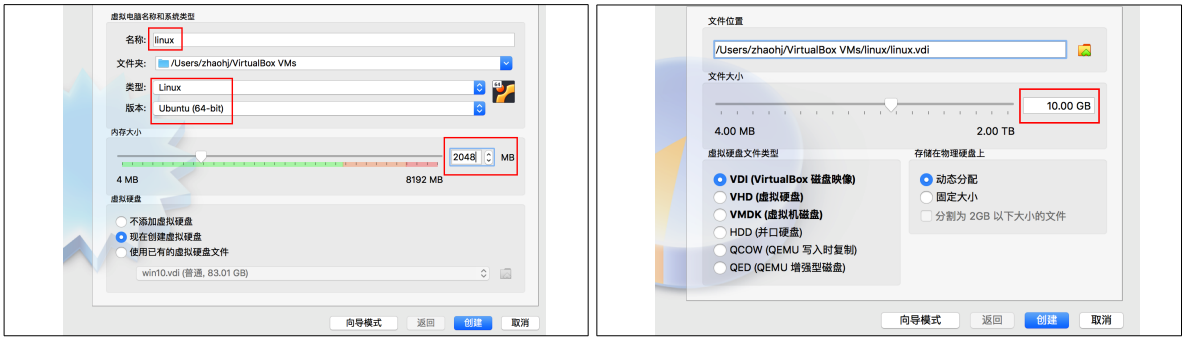

Figure 1.3: Allocate virtual main memory (left) and disk space (right) for your machine

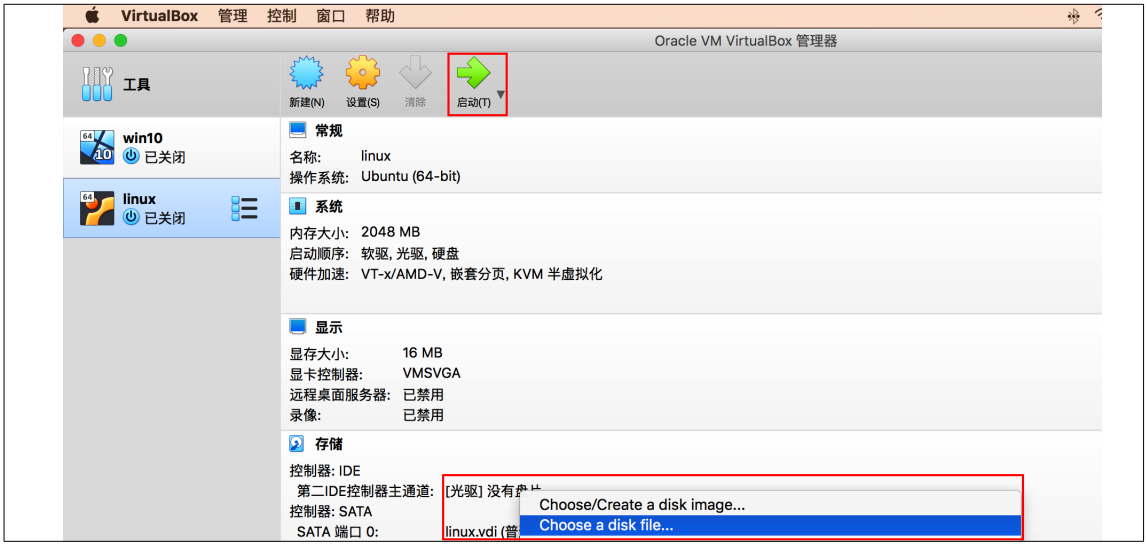

Figure 1.4: `Start` machine from optical drive with ISO image loaded

1.1.3 Install Ubuntu for the Virtual Machine

Now we show the process of installing Ubuntu on the bare machine created above.

i. First, `Start` your created virtual machine after loading the downloaded image, e.g., on my computer it is the `ubuntu-20.04.01-desktop-amd64.iso`, into the optical drive, cf. Fig. 1.4;

ii. When the machine is started, click `Install Ubuntu` and the installation process will begin (cf. Fig. 1.5);

iii. Go ahead with the installation guidance by choosing the default options until you are asked to set your name and password (cf. Fig. 1.6);

iv. The installation process proceeds and when it is completed a reboot of your system is required.

Figure 1.5: Click `Install Ubuntu` to start the installation process

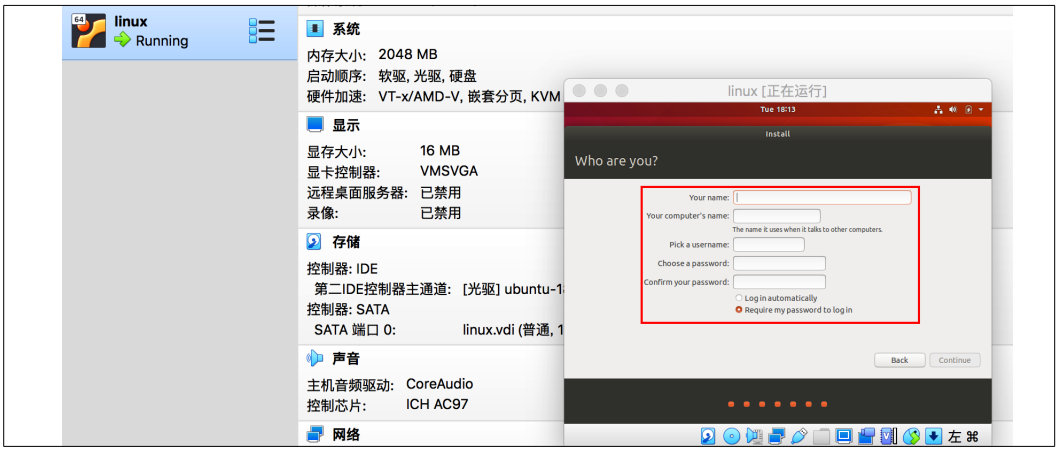

Figure 1.6: Set your `username` and `password` for the installed system

1.1.4 Install C-Programming Packages for Ubuntu

Suppose that you have Internet connection on your host operating system, then the packages supporting C programming in Ubuntu can be installed by the following steps.

i. Right click on the Desktop in Ubuntu, and then click the `Open Terminal` menu, cf. Fig 1.7 (left);

ii. Type `sudo apt-get update` into the window and press the `enter` key, cf. Fig 1.7 (right), the meaning of which will be explained in Section 1.2;

iii. Type `sudo apt-get install build-essential` into the window and press the `enter` key; then when `Do you want to continue?` is prompted, press `y` followed by the `enter` key.

The installation will finish after a while.

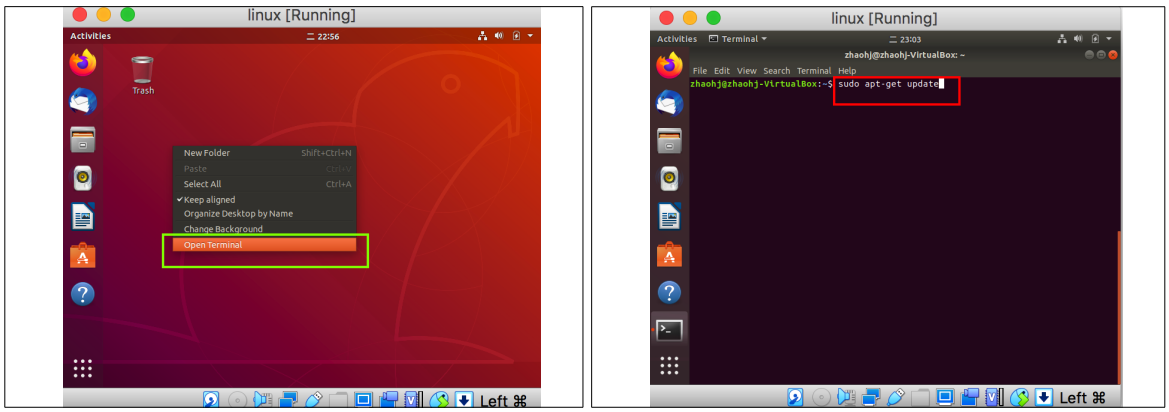

Figure 1.7: Open a terminal and install packages for C programming in Ubuntu

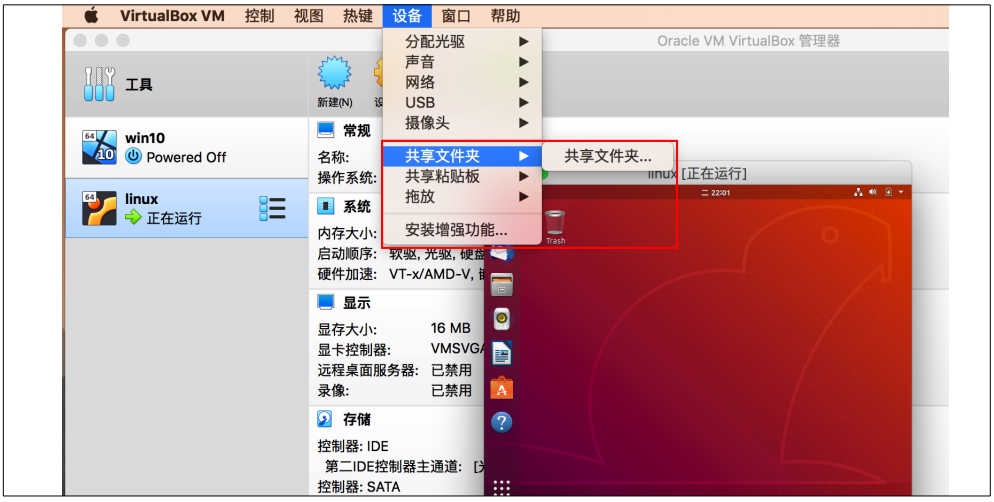

Figure 1.8: Settings facilitate your use of VirtualBox and Ubuntu

1.1.5 Other Settings

There are some additional settings that facilitate your use of Ubuntu in a virtual machine environment, as shown in Fig. 1.8, which can be found under the `Devices` menu of VirtualBox.

- Guest Additions: Guest Additions provides better performance and usability for the guest operating system, e.g., shared folder, shared clipboard, and drag and drop operations will be supported[2]; it can be installed by clicking on the menu at the bottom as shown in Fig. 1.8;
- Shared folder: setting a specific folder of the host operating system (OS), e.g., Windows 10, as a shared folder enables the sharing of files between Ubuntu and the host OS; You may need to add your username to the `vboxsf` group to gain permission to access the shared folder, e.g. on my computer the command `sudo usermod -aG vboxsf os` needs to be executed where `os` is my username;

[2]A system reboot may be required and these functions may not work on all platforms.

- Shared clipboard: setting this option to bidirectional enables the sharing of clipboard;
- Drag and drop: setting this option to bidirectional enables the drag and drop operations between the host and guest OS.

1.2 Basic Commands for Using Linux

I should add the `head`, `tail` command and the `grep` command in revised versions. These two commands are used in companion with `dmesg` and pipe `|`. For example `ls | tail -n 10`

1.3 C Programming in Linux

1.4 The `make` Tool

1.5 Linux Kernel Module

Chapter 2

Processes and Threads

TODO: chapter 2 to be completed!

TODO: C code format

TODO: The following two references are going to be replaced: Surname and Surname, 2017b, Surname and Surname, 2017a

Bibliography

Surname, N. and Surname, N. (2017a). An article title. *The Journal*, pages 0--10.

Surname, N. and Surname, N. (2017b). *A book title*. The Publisher.