

# Install and Configure Ubuntu on a VirtualBox Virtual Machine

Ronald Mak

Department of Computer Engineering

Department of Computer Science

January 11, 2019

## Introduction

Because the class will use Linux software tools and utilities, we will install and configure Ubuntu as the guest operating running under VirtualBox. Ubuntu is a popular version of the Linux operating system. See <https://www.ubuntu.com/>. You should have already installed VirtualBox and created a virtual machine for Ubuntu (Figure 1). If not, read “[Install and Configure VirtualBox on Windows](#)”.

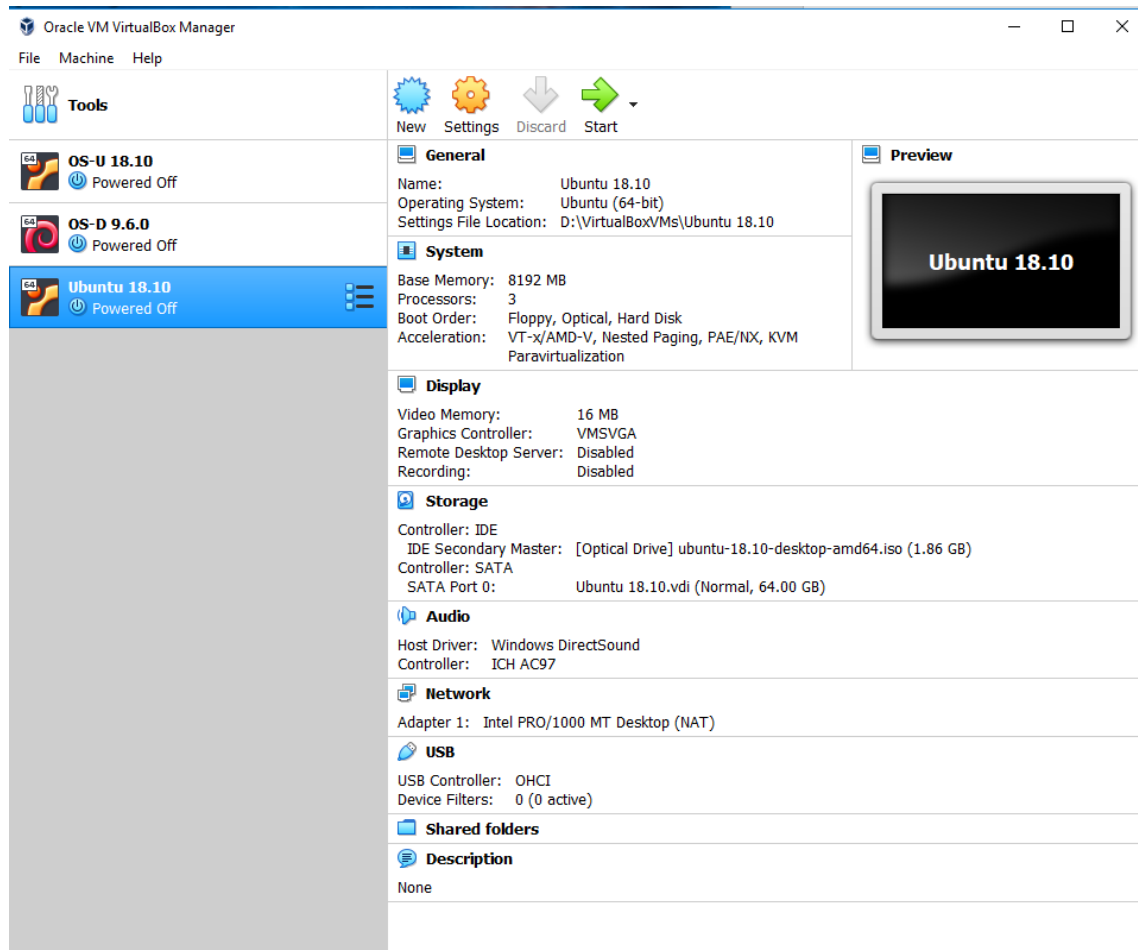


Figure 1. VirtualBox with an Ubuntu 18.10 virtual machine.

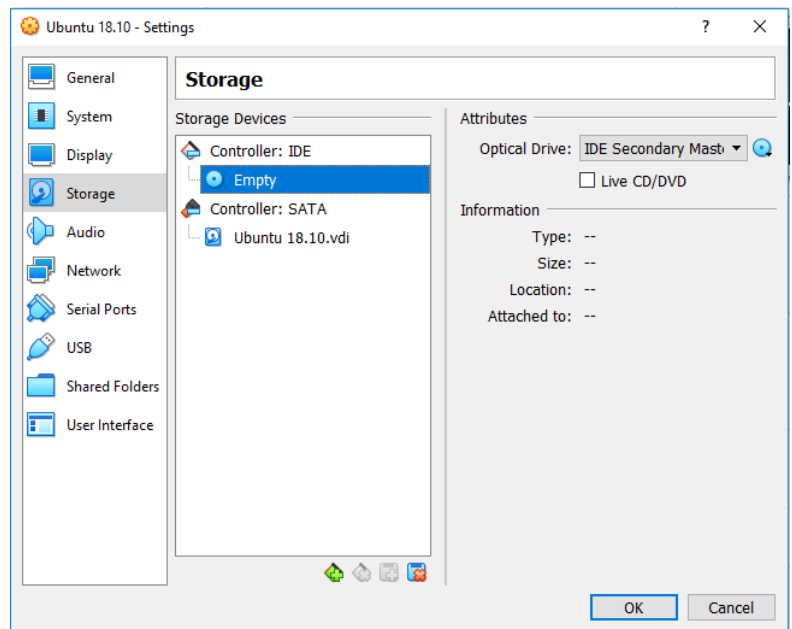
## Download the Ubuntu installation disk

Go to <https://www.ubuntu.com/download/desktop> and download Ubuntu 18.10, which is a 64-bit operating system. You will get a .iso file which is an image of the installation optical disk (i.e., a CD ROM). Remember where you stored the file.

## Insert the installation disk

We want to install Ubuntu on the virtual machine, so we must “insert” the .iso installation disk image file that we downloaded earlier into the virtual CD ROM drive. In the VirtualBox main screen, select the name of the Ubuntu virtual machine and click the *Settings* icon at the top. Click *Storage* in the left panel. The **Storage Settings** form (Figure 2) shows the virtual CD ROM drive, which is initially empty, and the virtual hard drive, which is the .vdi virtual disk image that VirtualBox created. Select *Empty* under *Controller: IDE*.

Figure 2. The initial Storage Settings form.



Look under *Attributes* and click on the image of the disk to the right of the drop-down menu. Select the .iso file to insert into the drive. You should now see the .iso file name under *Controller: IDE* (Figure 3). Click the OK button.

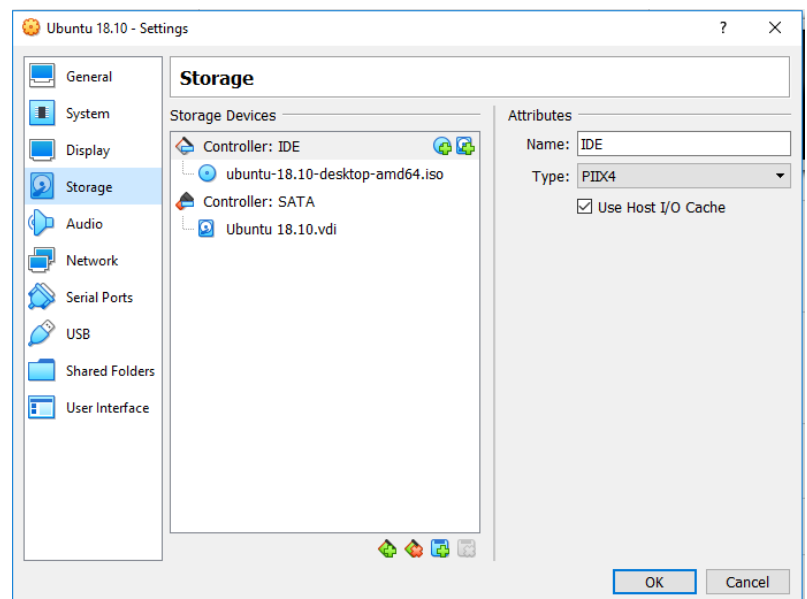


Figure 3. The installation disk image inserted into the virtual CD ROM drive.

## Install Ubuntu

On the main VirtualBox screen, select the Ubuntu 18.10 name and click the *Start* icon at the top to start the virtual machine. We had inserted the Ubuntu installation disk into the virtual CD ROM drive, so the virtual machine will boot off that disk and start the installation process (Figure 4). Because the installation process downloads files from the Internet, you will need a good connection.

Choose your language and click *Install Ubuntu* to start the installation process. At various times, you will be asked to make a choice. Except otherwise directed, you should accept the default choices.

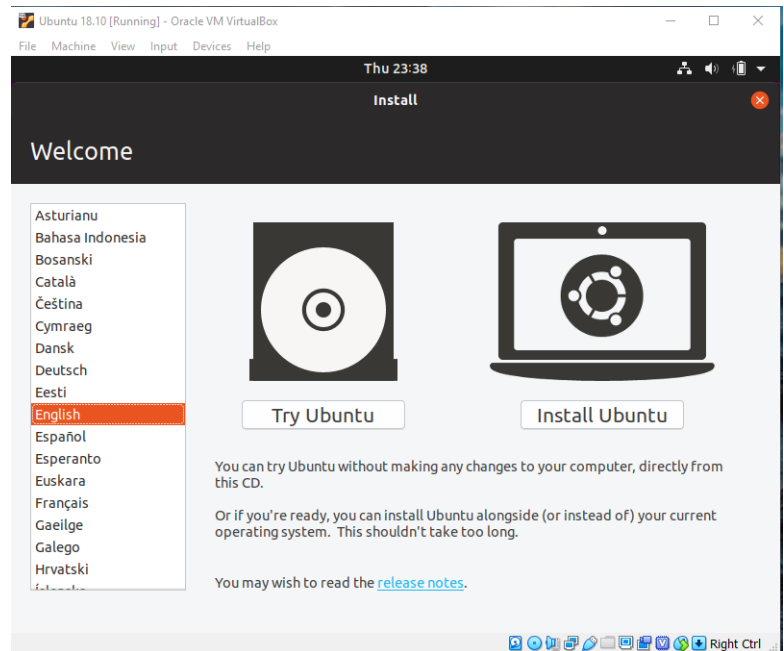


Figure 4. Starting the installation process.

On the **Installation type** form, accept the choice *Erase disk and install Ubuntu* (Figure 5). The disk that it will erase is the virtual disk, not your host laptop's physical disk! Click the *Install Now* button and the subsequent *Continue* button.

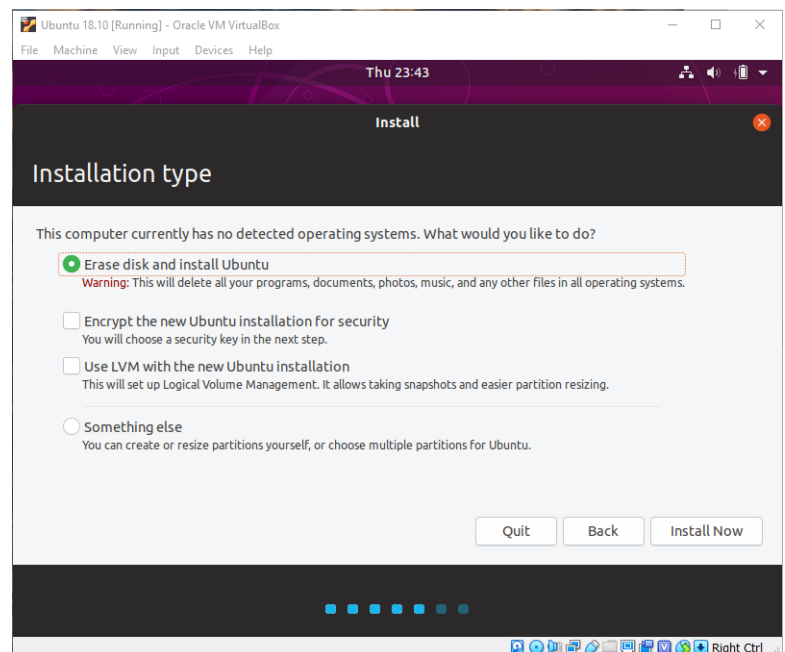


Figure 5. Erase the virtual disk and install Ubuntu.

## R. Mak, Install and Configure Ubuntu on a VirtualBox Virtual Machine

On the **Who are you?** form (Figure 6), enter appropriate names. The username will also become the name of your home directory in /home. The passwords will become your login passwords.

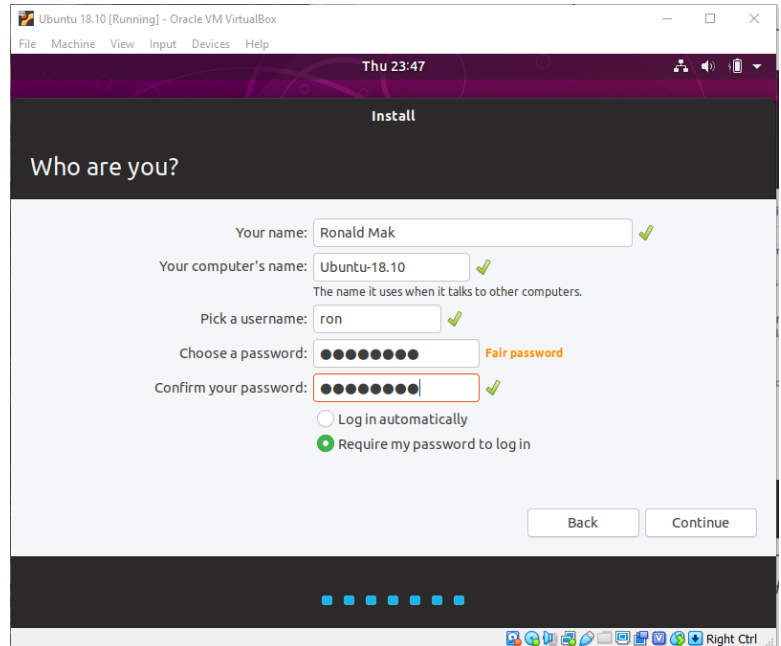


Figure 6. Choose names and passwords.

The installation process will take a while, especially if you have a slow Internet connection. You can watch files download and install, or you can go have a cup of coffee. When it's finally done, it will ask you to restart (Figure 7). Press the *Restart Now* button.

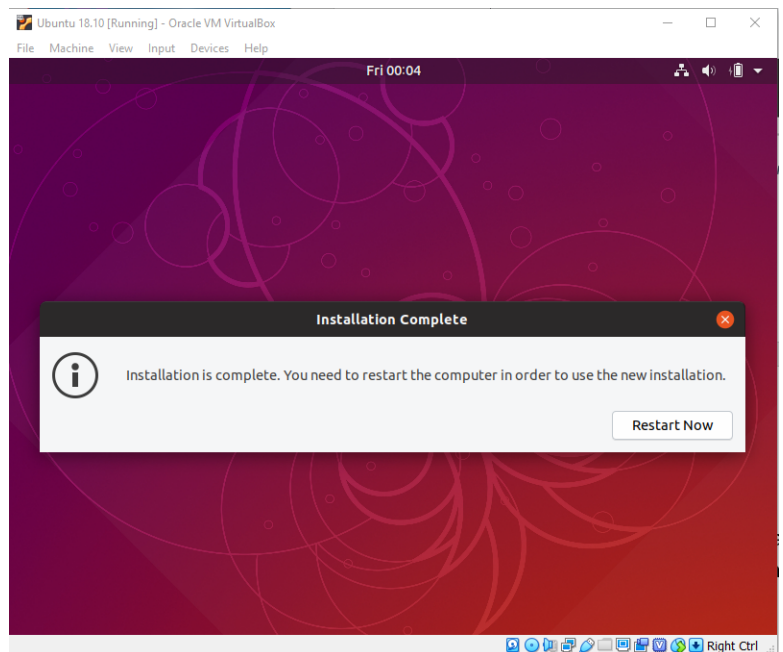


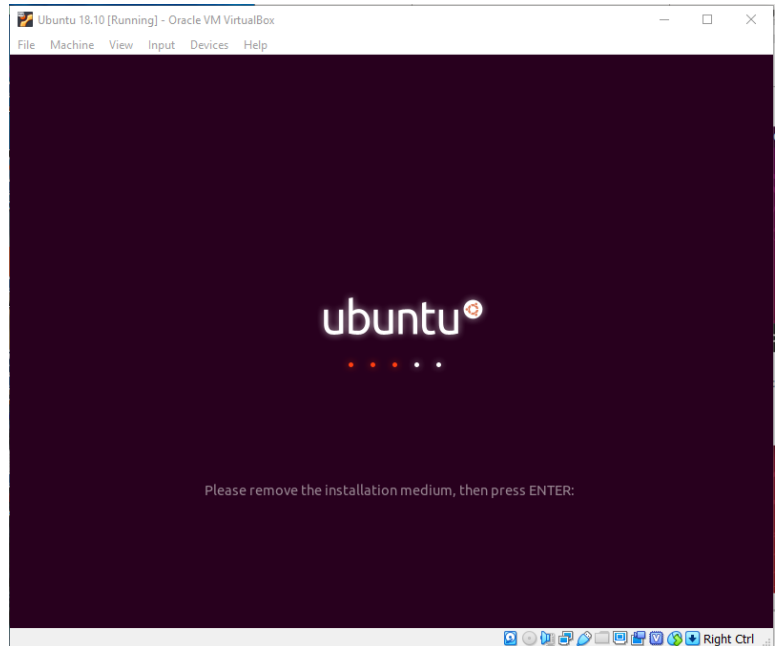
Figure 7. Installation is complete.

## R. Mak, Install and Configure Ubuntu on a VirtualBox Virtual Machine

Ubuntu will restart and ask you to remove the installation disk from the virtual CD ROM drive (Figure 8).

Right-click on the image of the CD ROM at the bottom of the screen and select *Remove disk from virtual drive*. You may need to press the right control key on your keyboard if the virtual machine has “captured” your mouse. After removing the disk, click in the Ubuntu window and press the enter key.

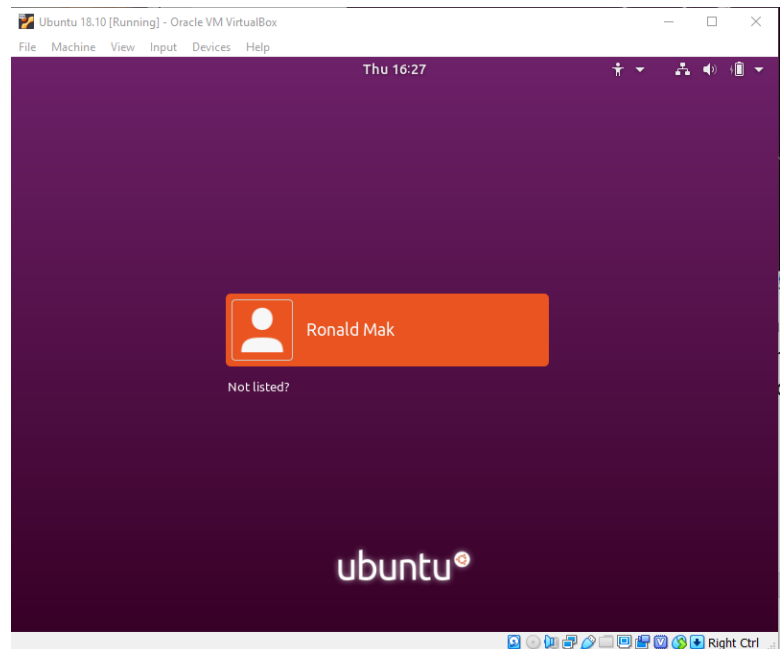
Figure 8. After the initial system restart.



After the VirtualBox splash screen and some system startup messages, you will be asked to log in (Figure 9).

Click on your name and enter your password.

Figure 9. The login screen.



You have successfully installed Ubuntu as a virtual machine (Figure 10)!

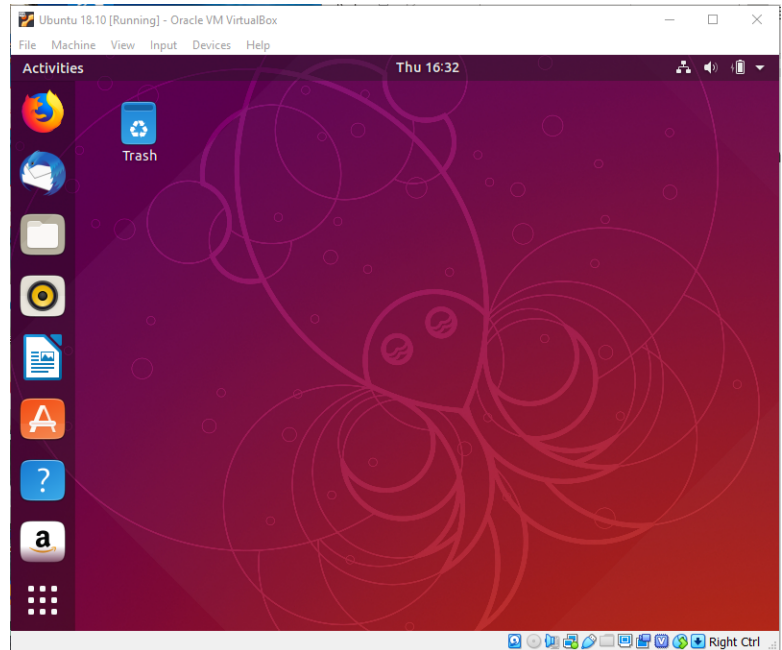
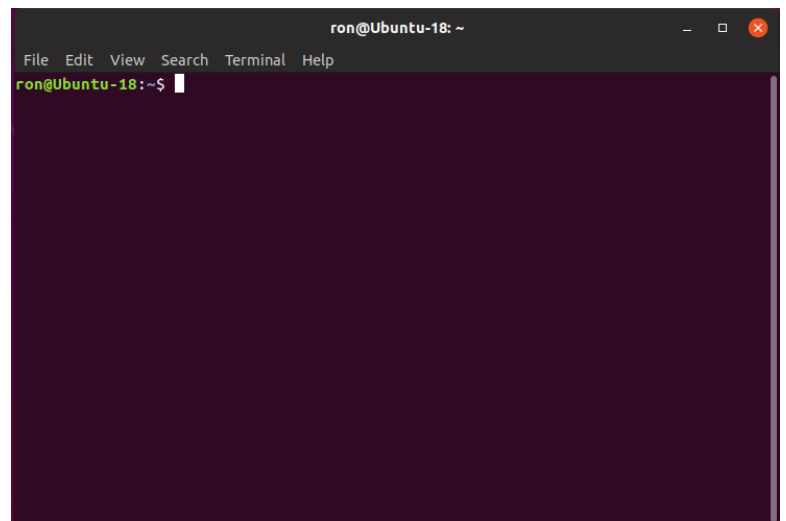


Figure 10. Ubuntu is successfully installed!

## Open an Ubuntu terminal window

Open an Ubuntu terminal window in order to enter and execute commands on the command line. Click the Show Applications icon (the matrix of nine dots) at the bottom left of the desktop to display application icons. In the search box at the top, type "terminal". The terminal icon appears. Right-click the icon and select *Add to Favorites* to add a terminal icon to the left panel so that you can easily open a terminal window in the future. Double-click the terminal icon to open a terminal window (Figure 11).

Figure 11. An Ubuntu terminal window.



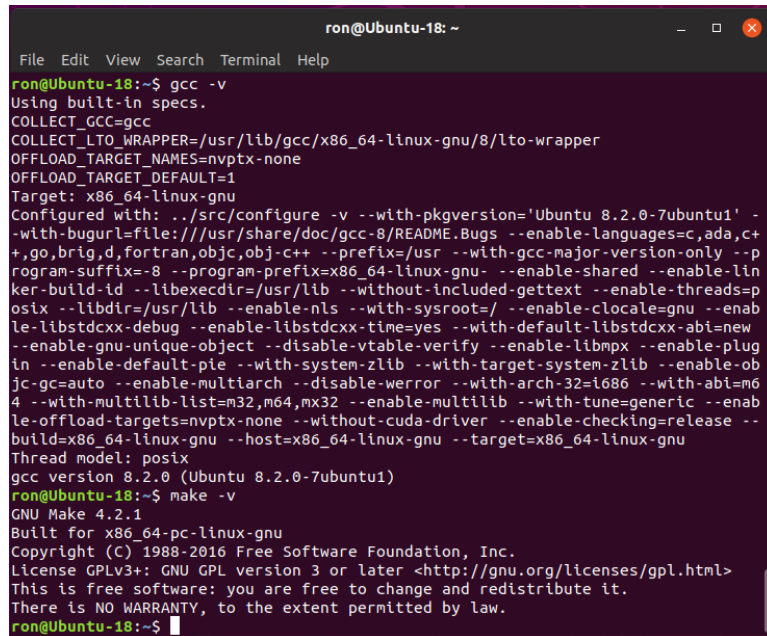
## Install developer tools

The Ubuntu distribution does not include developer tools such as the GNU gcc and g++ compilers and GNU make. See <https://help.ubuntu.com/community/InstallingCompilers> and follow its instructions.

After installing the developer tools, you should be able to verify that gcc, g++, and make were properly installed by entering the following commands in the Ubuntu terminal (Figure 12):

```
gcc -v
g++ -v
make -v
```

Figure 12. Verify that gcc, g++, and make were properly installed.



```
ron@Ubuntu-18: ~
File Edit View Search Terminal Help
ron@Ubuntu-18:~$ gcc -v
Using built-in specs.
COLLECT_GCC=gcc
COLLECT_LTO_WRAPPER=/usr/lib/gcc/x86_64-linux-gnu/8/lto-wrapper
OFFLOAD_TARGET_NAMES=nvptx-none
OFFLOAD_TARGET_DEFAULT=1
Target: x86_64-linux-gnu
Configured with: ../src/configure -v --with-pkgversion='Ubuntu 8.2.0-7ubuntu1' -
--with-bugurl=file:///usr/share/doc/gcc-8/README.Bugs --enable-languages=c,ada,c+
+,go,brig,d,fortran,objc,obj-c++ --prefix=/usr --with-gcc-major-version-only --p
rogram-suffix=-8 --program-prefix=x86_64-linux-gnu- --enable-shared --enable-lin
ker-build-id --libexecdir=/usr/lib --without-included-gettext --enable-threads=p
osix --libdir=/usr/lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enab
le-libstdcxx-debug --enable-libstdcxx-time=yes --with-default-libstdcxx-abi=new
--enable-gnu-unique-object --disable-vtable-verify --enable-libmpx --enable-plug
in --enable-default-pie --with-system-zlib --with-target-system-zlib --enable-ob
jc-gc=auto --enable-multiarch --disable-werror --with-arch=32=i686 --with-abi=m6
4 --with-multilib-list=m32,m64,mx32 --enable-multilib --with-tune=generic --enab
le-offload-targets=nvptx-none --without-cuda-driver --enable-checking=release --
build=x86_64-linux-gnu --host=x86_64-linux-gnu --target=x86_64-linux-gnu
Thread model: posix
gcc version 8.2.0 (Ubuntu 8.2.0-7ubuntu1)
ron@Ubuntu-18:~$ g++ -v
GNU Make 4.2.1
Built for x86_64-pc-linux-gnu
Copyright (C) 1988-2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
ron@Ubuntu-18:~$
```

## Install guest additions

Guest additions are additional software that you install in the guest operating system (i.e., Ubuntu) that provide important useful features, such as shared folders and the ability to cut and paste between the guest and host environments. Guest additions are fully explained in Chapter 4 of the *VirtualBox User Manual*, which is in the doc directory of the VirtualBox installation on your host machine.

In the top menu bar of the virtual machine window (you may need to hold down the right control key on your host keyboard and press C to make the top menu bar appear), drop down the Devices menu and select *Insert Guest Additions CD image ....*

You can also insert the image manually. Look for the virtual disk image file VBoxGuestAdditions.iso in the top folder of your installation. Right-click the image of the CD ROM at the bottom of the virtual machine window (right-click control-C if necessary to see it) and insert the .iso file.

A dialog will pop up asking if you want to run automatically started software (Figure 13). Click the *Run* button and the software will install (Figure 14).

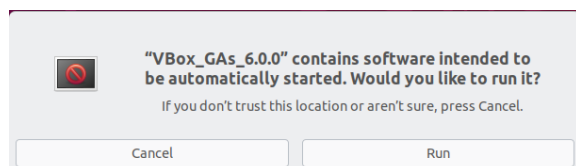


Figure 13. Click the Run button to install the guest additions.

## R. Mak, Install and Configure Ubuntu on a VirtualBox Virtual Machine

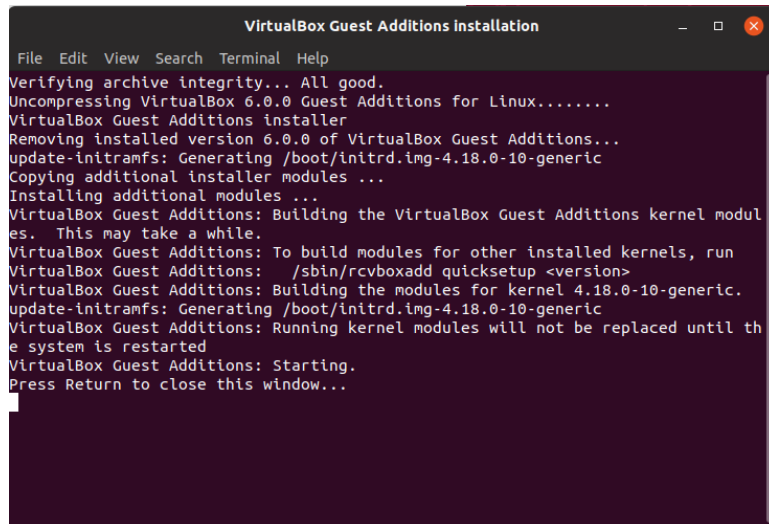


Figure 14. Installing the guest additions.

You can also start the installation script manually. Click the Vbox\_GAS icon in the upper left of the Ubuntu desktop to display the contents of the virtual disk (Figure 15). Click the Run Software button in the upper right.

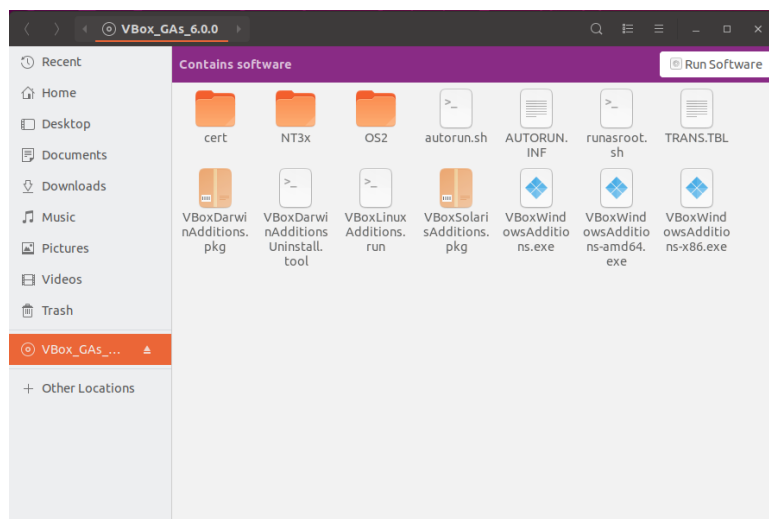


Figure 15. Contents of the guest additions virtual disk.



## Create and mount the shared folder

A shared folder is a directory on your host operating system (i.e., Windows) that is also accessible by the guest operating system (i.e., Ubuntu). A shared folder is the easiest way to transfer files between the host and the guest.

In Windows, create a directory, such as D:\Shared, that you wish to be shared. In the VirtualBox main screen, select the Ubuntu guest operating system name and click the *Settings* icon at the top. Click on *Shared Folders* in the left panel. In the **Shared Folders** form (Figure 16), click the + folder icon on the right. In the dialog box, enter the *Folder Path* of your directory and “Shared” as the *Mount point*. Check the *Auto-mount* and *Make Permanent* boxes. Click the *OK* button to see your choices (Figure 17).

Figure 16. Shared folder dialog box entries.

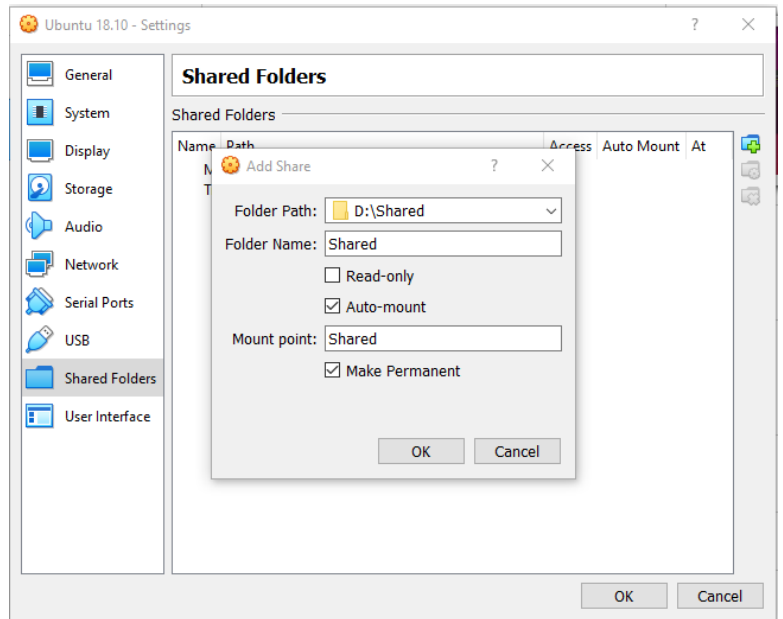
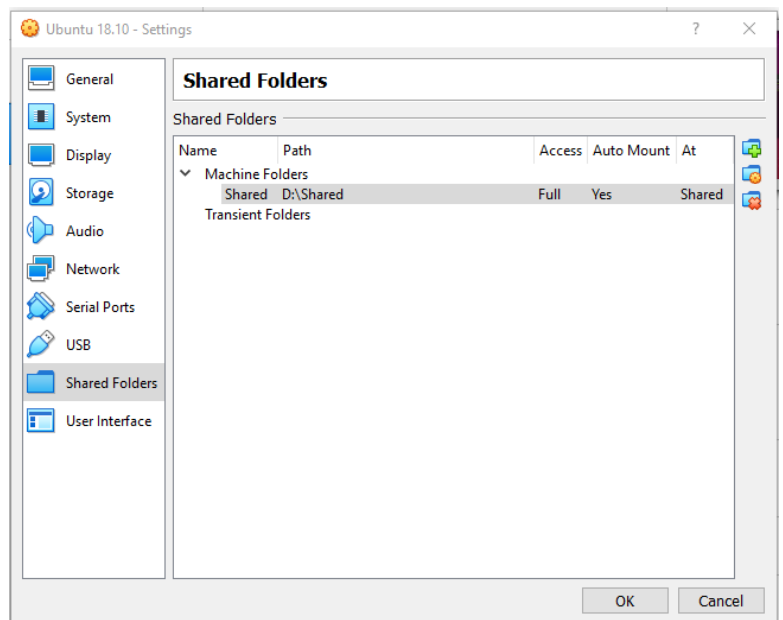


Figure 17. Shared folder specification.



In the Ubuntu operating system, in order to access data in the shared folder, you must be in the vboxsf group. Enter the following command on the command line in the terminal window:

```
sudo usermod -a -G vboxsf username
```

where *username* is your username (e.g., ron).

To mount the shared folder, enter the following commands:

```
mkdir Shared  
sudo mount -t vboxsf -o uid=1000,gid=1000 Shared /home/username/Shared
```

where *username* is your username (e.g., ron). The **mkdir** command creates a folder (directory) named Shared in your home directory. The **sudo mount** command mounts the shared folder to the folder you just created in your home directory. Now the folder you created in Windows (say D:\Shared) is shared with folder Shared in your home directory.

To test the shared folder: Put some files into the shared Windows directory (say, BigPi.cpp and BigPi-Windows10.txt). Then use the **ls** command in the Ubuntu terminal to see the files listed:

```
ls shared  
BigPi.cpp BigPi-Windows10.txt
```

To make this mount permanent so that it will be in effect the next time you start Ubuntu, execute the following four commands:

```
sudo su  
cp /etc/fstab /etc/fstab-save  
echo "Shared /home/username/Shared vboxsf defaults 0 0" >> /etc/fstab  
exit
```

where *username* is your username (e.g., ron). Be especially careful typing the **echo** command, particularly the double >> symbol. If you accidentally trash the system file /etc/fstab, you can recover it from the copy /etc/fstab-save.

The shared folder is also automatically mounted for you in /media/sf\_Shared.