COMP1047 – Computer Networks – Lab 1.1

Install Ubuntu on VirtualBox

School of Computer Science
University of Nottingham Ningbo China

Introduction

Linux is an industry-standard operating system, and **Ubuntu** is a popular distribution of Linux. In order to run Ubuntu as the guest operating system in a virtual machine on your host machine (such as your Windows or Mac laptop), you can first install VirtualBox and then install Ubuntu within VirtualBox. Then whenever you start VirtualBox, you can select Ubuntu (or any other guest operating system that you've installed) to run in the virtual machine managed by VirtualBox.

If you are on **Windows 10**, a better solution for getting Ubuntu is to install it directly using the Windows Subsystem for Linux feature. For instructions, see https://docs.microsoft.com/en-us/windows/wsl/install-win10.

Hardware-assisted virtualization

To support a virtual machine, your laptop's Intel CPU chip must enable virtualization technology (VT-x). It is enabled by default on some laptop brands, but disabled by default on others. To see whether or not it's enabled on your Windows laptop, follow the instructions at https://www.intel.com/content/www/us/en/support/articles/000005486/processors.html

If VT-x is disabled, you must enable it by changing an option in your laptop's. See the instructions at https://www.sysprobs.com/disable-enable-virtualization-technology-bios

How to access the BIOS depends on the laptop brand: https://www.lifewire.com/bios-setup-utilityaccess-keys-for-popular-computer-systems-2624463

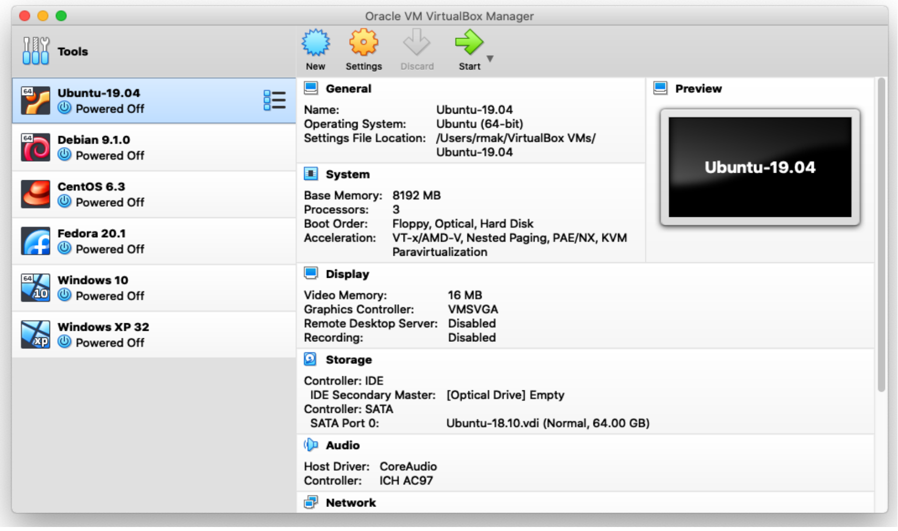
Download the Ubuntu installation disk

Go to https://www.ubuntu.com/download/desktop and download the latest version of Ubuntu, 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.

Install and configure VirtualBox

VirtualBox is a virtual machine manager under which you will run Ubuntu. Windows 10 or macOS will be the **host operating system** for the virtual machine, and Ubuntu will be the **guest operating system** running in the virtual machine.

Download the 6th version of VirtualBox from https://www.virtualbox.org/. Install and run it. It should appear as in Figure 1, except that if this is your first time running it, there won't be any other guest operating systems already installed.



Ubuntu-19.04 Powered Off
Debian 9.1.0 Powered Off
CentOS 6.3 Powered Off
Fedora 20.1 Powered Off
Windows 10 Powered Off
Windows XP 32 Powered Off

General
Name: Ubuntu-19.04
Operating System: Ubuntu (64-bit)
Settings File Location: /Users/rmak/VirtualBox VMs/Ubuntu-19.04

System
Base Memory: 8192 MB
Processors: 3
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage
Controller: IDE
IDE Secondary Master: [Optical Drive] Empty
Controller: SATA
SATA Port 0: Ubuntu-18.10.vdi (Normal, 64.00 GB)

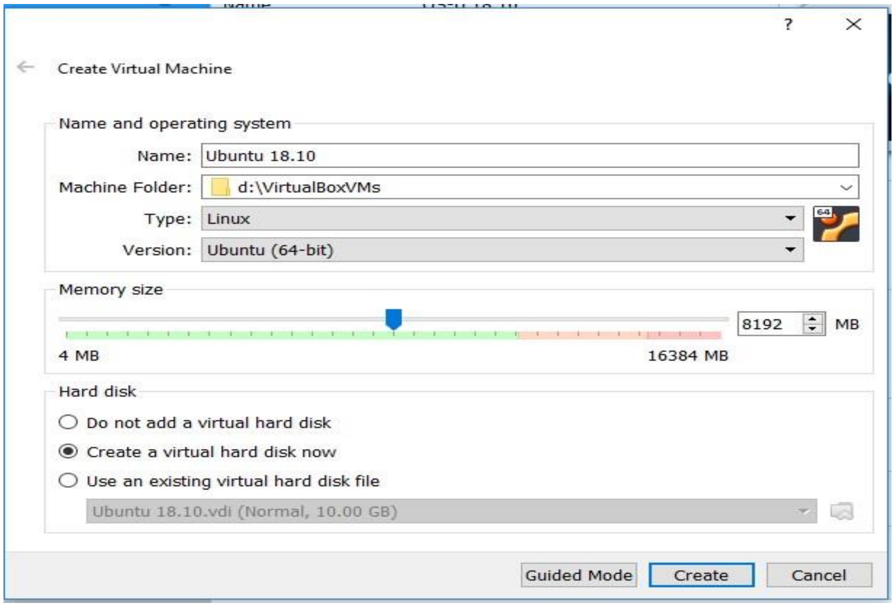
Audio
Host Driver: CoreAudio
Controller: ICH AC97

Network

Preview
Ubuntu-19.04

Figure 1. The VirtualBox main screen (running on MacOS X).

Click the *New* button and fill in the **Create Virtual Machine** form (Figure 2). Pick an appropriate name. Choose a machine folder on your laptop that will contain the virtual machine image. The type should be *Linux*, and the version should be *Ubuntu (64 bit).* Set a memory size for the virtual machine, but don't give it more than half of the memory of your host laptop. Select the *Create a virtual hard disk now* radio button. Click the *Create* button.



Create Virtual Machine

Name and operating system
Name: Ubuntu 18.10
Machine Folder: d:\VirtualBoxVMs
Type: Linux
Version: Ubuntu (64-bit)

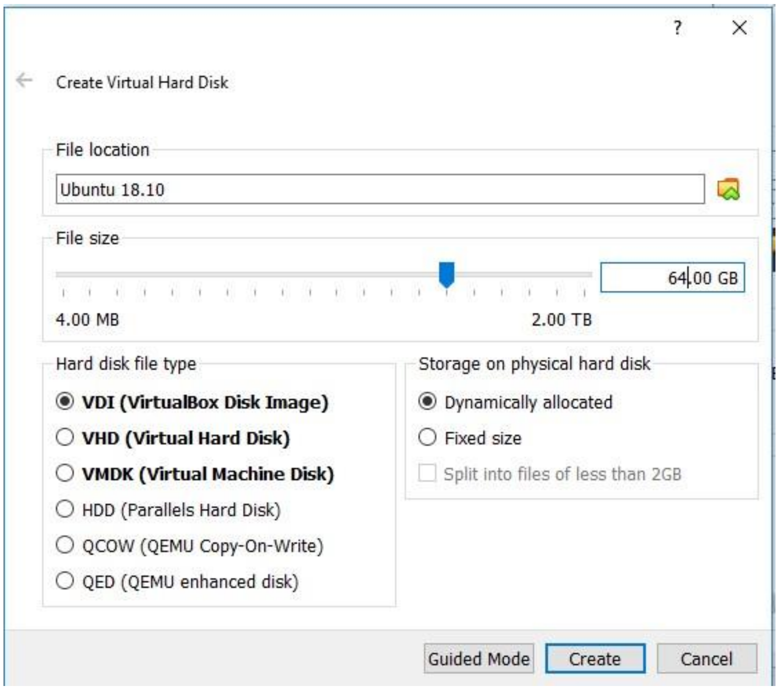
Memory size
8192 MB
4 MB
16384 MB

Hard disk
○ Do not add a virtual hard disk
● Create a virtual hard disk now
○ Use an existing virtual hard disk file
Ubuntu 18.10.vdi (Normal, 10.00 GB)

Figure 2. The Create Virtual Machine form.

On the **Create Virtual Hard Disk** form (Figure 3), specify the maximum size to which Ubuntu's virtual hard disk can grow. Select the *VDI (VirtualBox Disk Image)* and *the Dynamically allocated* radio buttons. Click the *Create* button. This creates the new virtual machine named, for example, Ubuntu 14.04.16 (Figure 4).

Note: For an installation of Ubuntu, specify at least **64 GB** for the size of the virtual hard disk.



Create Virtual Hard Disk

File location

Ubuntu 18.10

File size

64.00 GB

4.00 MB

2.00 TB

Hard disk file type

● VDI (VirtualBox Disk Image)

○ VHD (Virtual Hard Disk)

○ VMDK (Virtual Machine Disk)

○ HDD (Parallels Hard Disk)

○ QCOW (QEMU Copy-On-Write)

○ QED (QEMU enhanced disk)

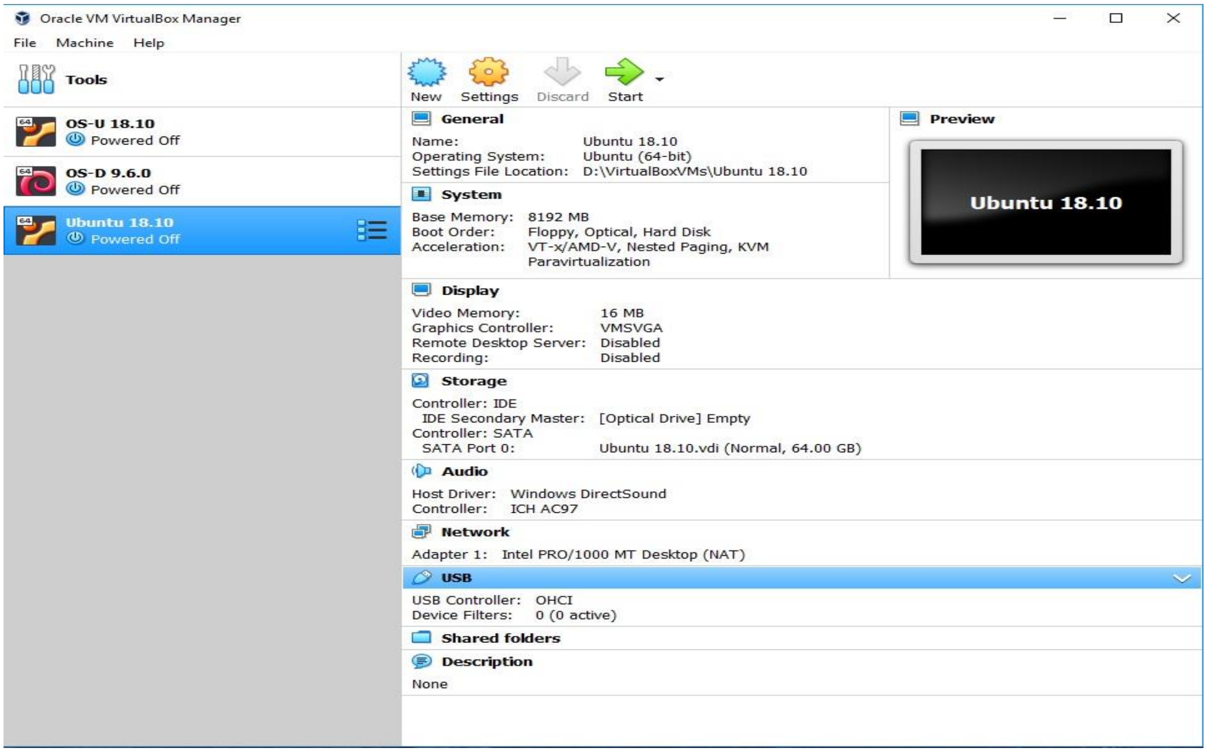
Storage on physical hard disk

● Dynamically allocated

○ Fixed size

□ Split into files of less than 2GB

Figure 3. The Create Virtual Disk form.



OS-U 18.10
Powered Off

OS-D 9.6.0
Powered Off

Ubuntu 18.10
Powered Off

General

```
Name:                    Ubuntu 18.10
Operating System:        Ubuntu (64-bit)
Settings File Location:  D:\VirtualBoxVMs\Ubuntu 18.10
```

System

```
Base Memory:   8192 MB
Boot Order:    Floppy, Optical, Hard Disk
Acceleration:  VT-x/AMD-V, Nested Paging, KVM
               Paravirtualization
```

Display

```
Video Memory:            16 MB
Graphics Controller:     VMSVGA
Remote Desktop Server:   Disabled
Recording:               Disabled
```

Storage

```
Controller: IDE
  IDE Secondary Master:  [Optical Drive] Empty
Controller: SATA
  SATA Port 0:           Ubuntu 18.10.vdi (Normal, 64.00 GB)
```

Audio

```
Host Driver:  Windows DirectSound
Controller:   ICH AC97
```

Network

```
Adapter 1:  Intel PRO/1000 MT Desktop (NAT)
```

USB

```
USB Controller:  OHCI
Device Filters:  0 (0 active)
```

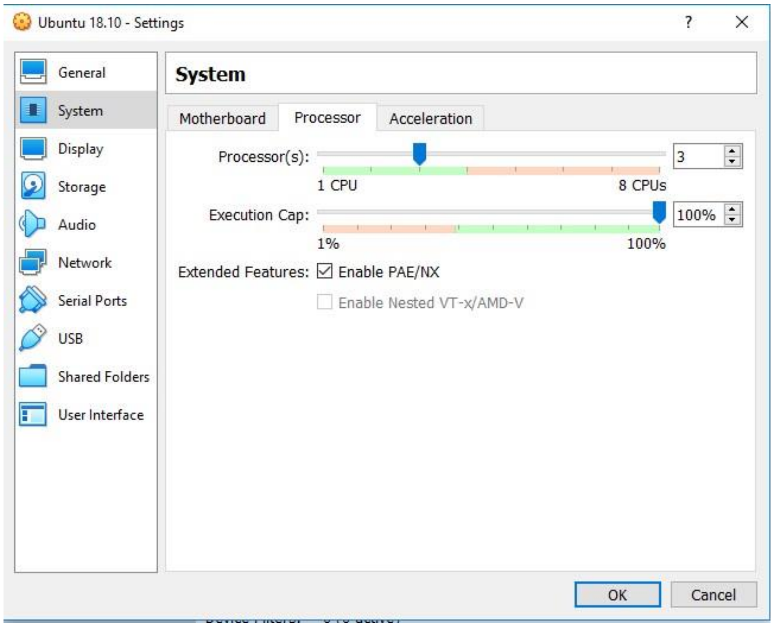
Shared folders

Description

None

Figure 4. The virtual machine named Ubuntu 18.10 is created.

Select the name of the newly created virtual machine. lick the *Settings* icon at the top. Then click on *System* in the left panel. In the **System Settings** form, select the *Motherboard* tab. Specify the amount of base memory you want to devote the virtual machine, but not more than half of the physical memory in your host machine. Select the *Processor* tab (Figure 5) and specify the number of CPUs you want to devote to the virtual machine, but not more than half the number of CPUs in your host machine. Click the *OK* button.



Processor(s): 3

Execution Cap: 100%

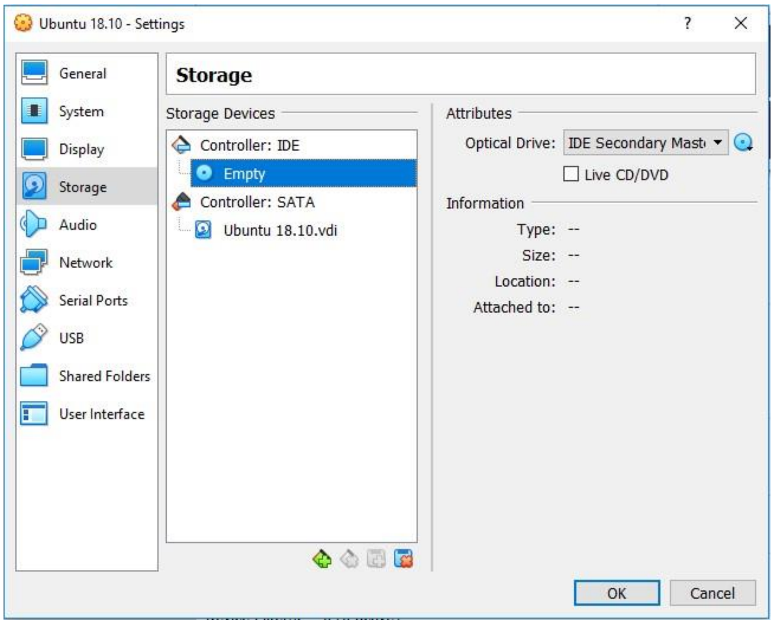
Extended Features: ☑ Enable PAE/NX

☐ Enable Nested VT-x/AMD-V

Figure 5. The System Settings form.

Note: For an installation of Ubuntu, specify at least **2 CPUs** and **4 GB of memory**.

Click *Storage* in the left panel. The **Storage Settings** form (Figure 6) 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*.



Controller: IDE
- Empty

Controller: SATA
- Ubuntu 18.10.vdi

Optical Drive: IDE Secondary Mast

☐ Live CD/DVD

Type: --

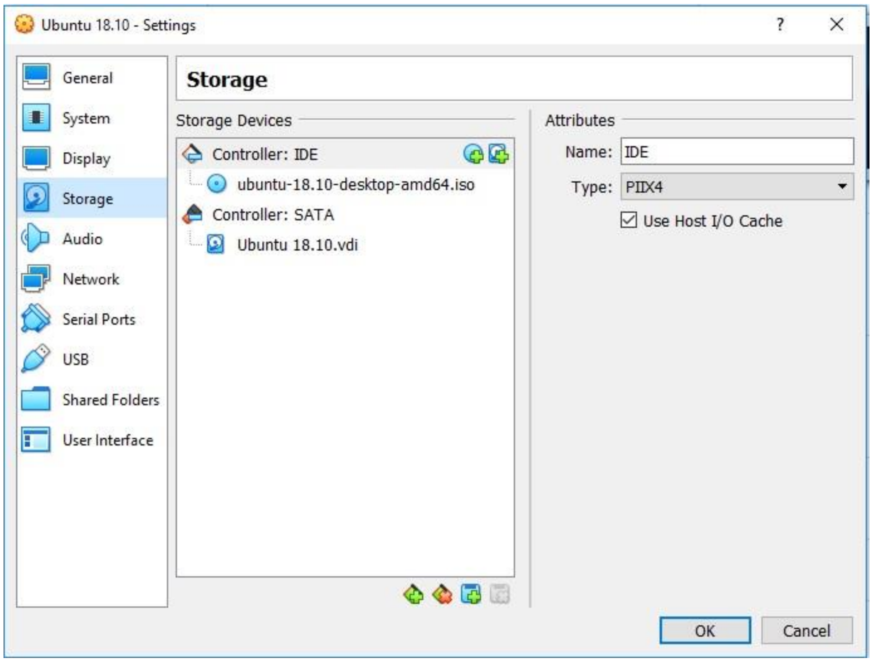
Size: --

Location: --

Attached to: --

Figure 6. The initial Storage Settings form.

You want to install Ubuntu on the virtual machine, so you must "insert" the .iso installation disk image file that you downloaded earlier into the virtual CD ROM drive. Look under *Attributes* and click on the image of the disk to the right of the dropdown menu. Select the .iso file to insert into the drive. You should now see the .iso file name under *Controller: IDE* (Figure 7). Click the *OK* button.



Storage

Storage Devices

Controller: IDE
ubuntu-18.10-desktop-amd64.iso
Controller: SATA
Ubuntu 18.10.vdi

Attributes

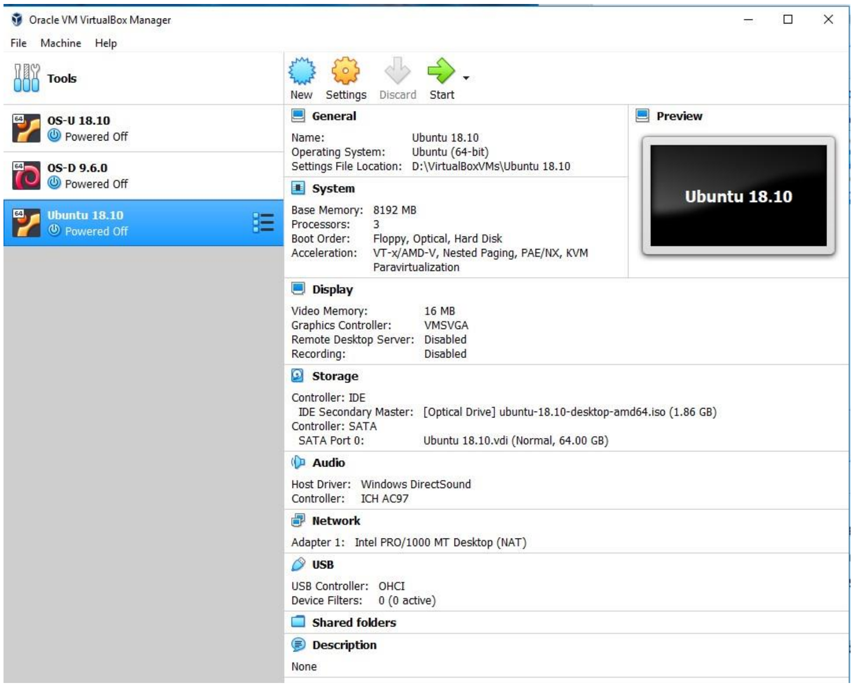
Name: IDE
Type: PIIX4
☑ Use Host I/O Cache

Figure 7. The installation disk image inserted into the virtual CD ROM drive.

Install Ubuntu (ubuntu-14.04.6-desktop-amd64.iso)

Now you are ready to start the virtual machine (Figure 8) in order to install Ubuntu. To start the Ubuntu virtual machine, highlight it and click the *Start* button at the top.

The virtual machine will boot off the installation disk and start the installation process (Figure 9). Because the installation process downloads files from the Internet, you will need a good connection.



OS-U 18.10
Powered Off

OS-D 9.6.0
Powered Off

Ubuntu 18.10
Powered Off

General

Name: Ubuntu 18.10
Operating System: Ubuntu (64-bit)
Settings File Location: D:\VirtualBoxVMs\Ubuntu 18.10

System

Base Memory: 8192 MB
Processors: 3
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] ubuntu-18.10-desktop-amd64.iso (1.86 GB)
Controller: SATA
SATA Port 0: Ubuntu 18.10.vdi (Normal, 64.00 GB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

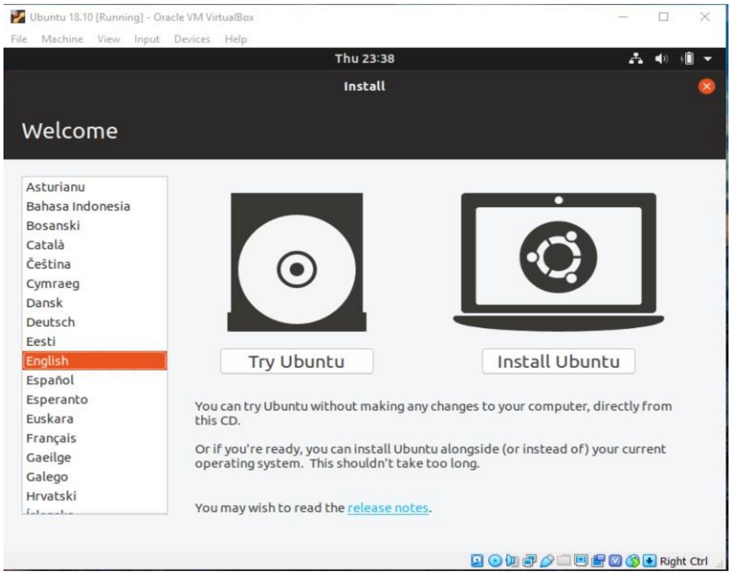
Shared folders

Description

None

Figure 8. Ready to start the Ubuntu virtual machine.

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.



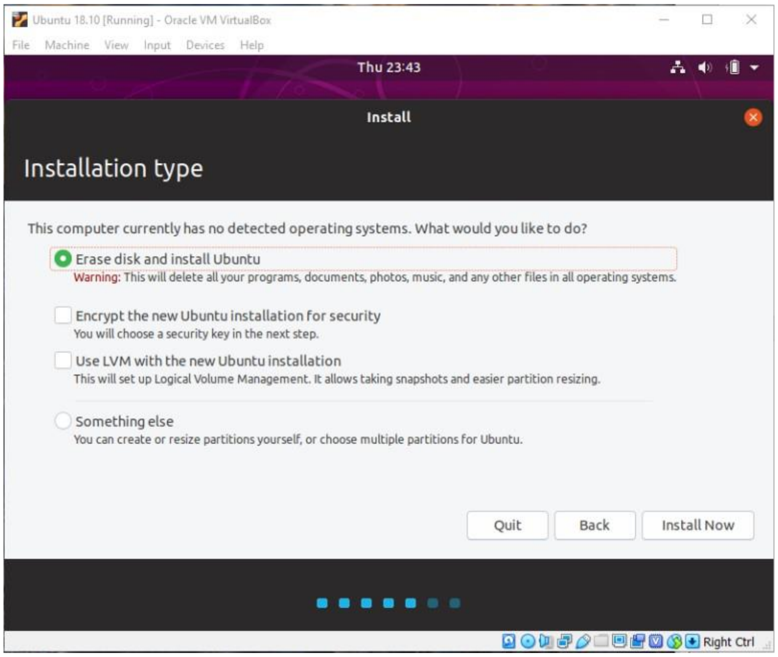
You can try Ubuntu without making any changes to your computer, directly from this CD.

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.

Figure 9. Starting the installation process.

On the **Installation type** form, accept the choice *Erase disk and install Ubuntu* (Figure 10). 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.


Ubuntu 18.10 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Thu 23:43
Install
Installation type
Quit
Back
Install Now
Right Ctrl

This computer currently has no detected operating systems. What would you like to do?

● Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.

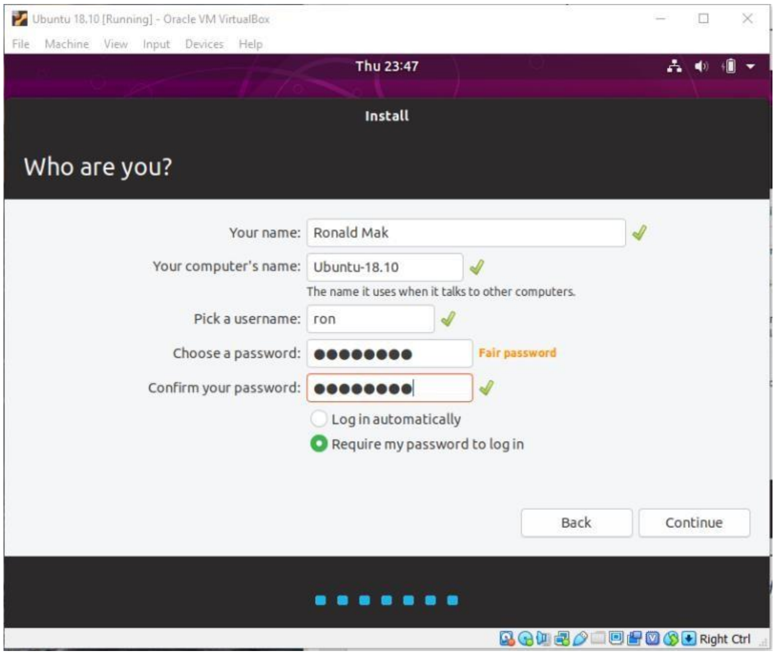
□ Encrypt the new Ubuntu installation for security
You will choose a security key in the next step.

□ Use LVM with the new Ubuntu installation
This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.

○ Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Figure 10. Erase the virtual disk and install Ubuntu.

On the **Who are you?** form (Figure 11), enter appropriate names. The username will also become the name of your home directory in /home. The passwords will become your login passwords.



Who are you?

Your name: Ronald Mak ✓

Your computer's name: Ubuntu-18.10 ✓

The name it uses when it talks to other computers.

Pick a username: ron ✓

Choose a password: ●●●●●●●● Fair password

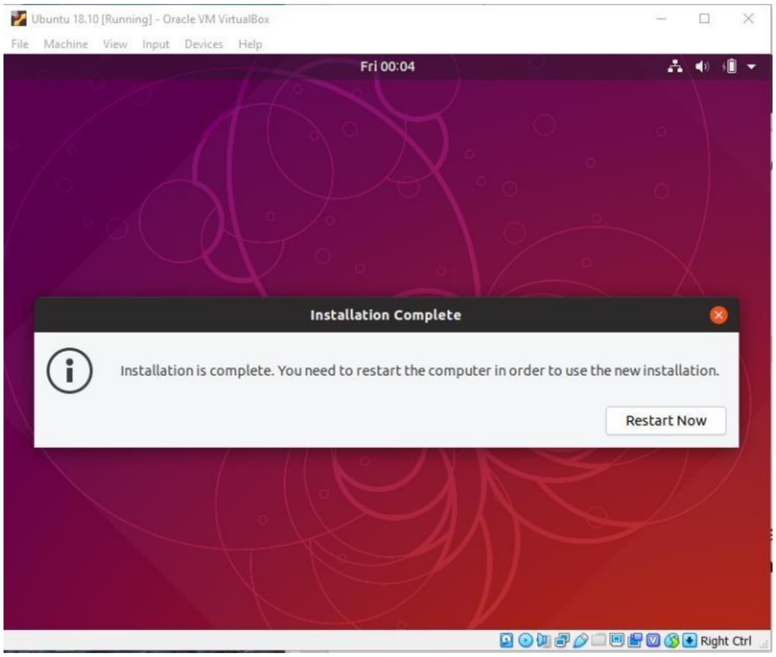
Confirm your password: ●●●●●●●● ✓

○ Log in automatically

● Require my password to log in

Figure 11. 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 12). Press the *Restart Now* button.

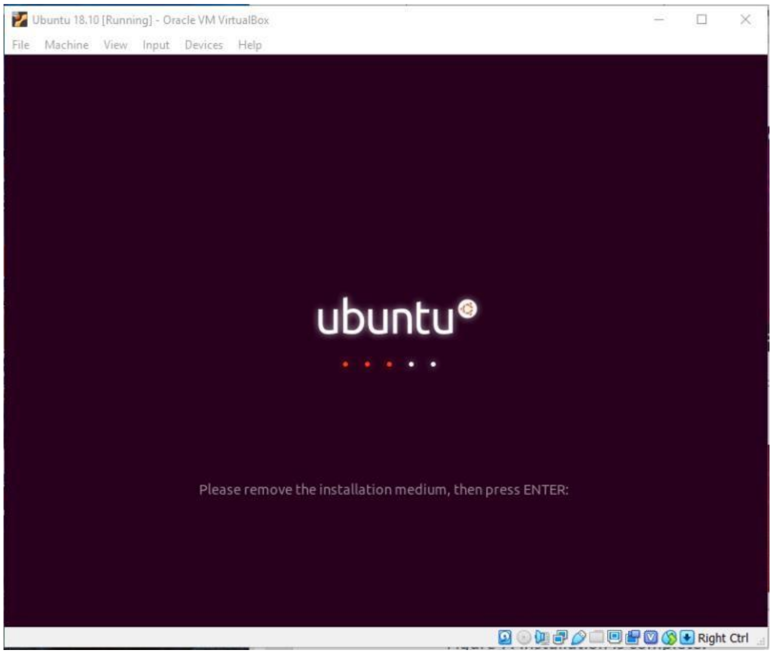


Installation is complete. You need to restart the computer in order to use the new installation.

Figure 12. Installation is complete.

Ubuntu will restart and ask you to remove the installation disk from the virtual CD ROM drive (Figure 13).

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.


Ubuntu 18.10 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ubuntu
Right Ctrl

Please remove the installation medium, then press ENTER:

Figure 13. After the initial system restart.

After the VirtualBox splash screen and some system startup messages, you will be asked to log in (Figure 14).

Click on your name and enter your password.

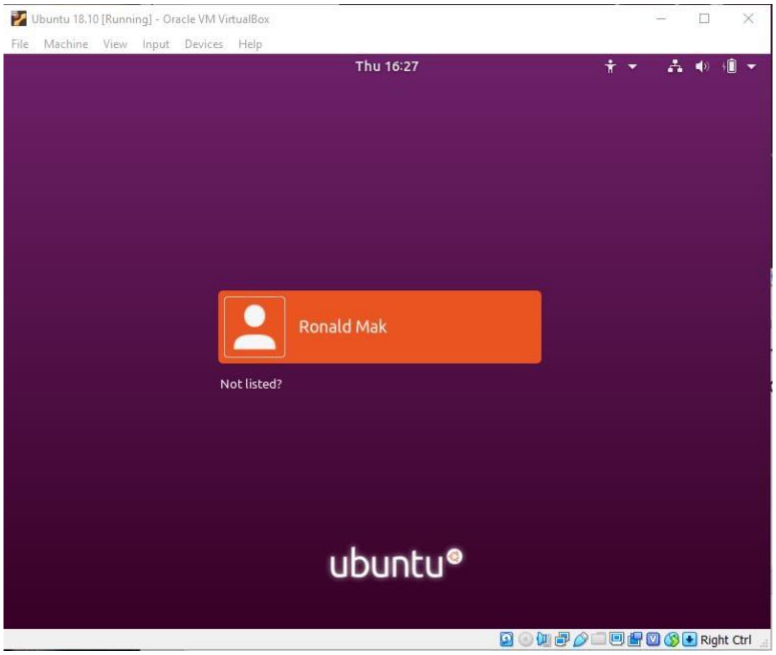

Ubuntu 18.10 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Thu 16:27
Ronald Mak
Not listed?
ubuntu
Right Ctrl

Figure 14. The login screen.

You have successfully installed Ubuntu as a virtual machine (Figure 15)!

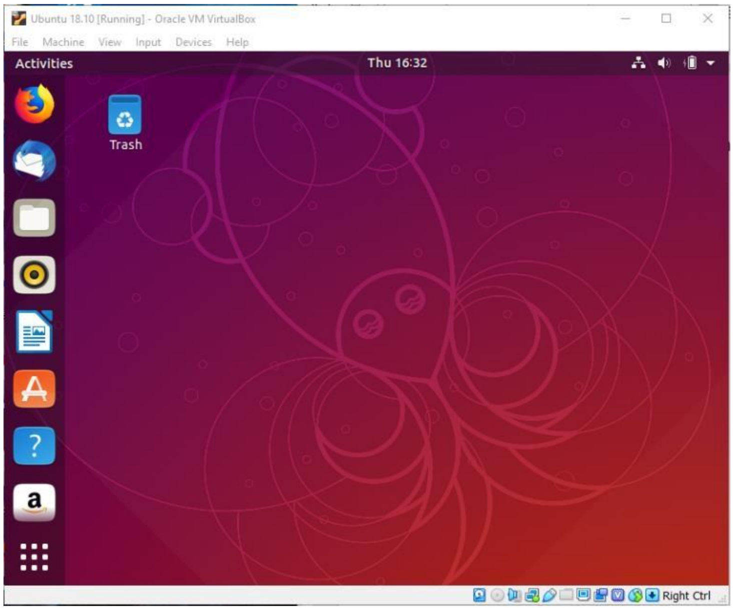
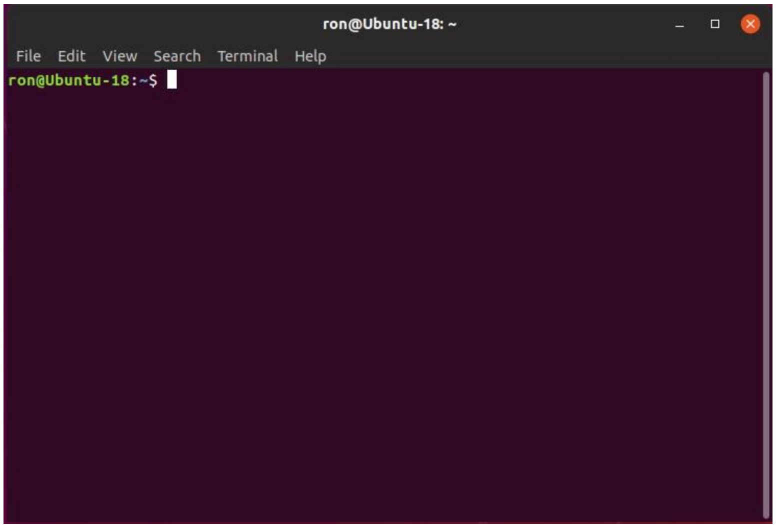

Ubuntu 18.10 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Thu 16:32
Trash
Right Ctrl

Figure 15. 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. Rightclick 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 16).



```
ron@Ubuntu-18:~$ 
```

Figure 16. An Ubuntu terminal window.

Congratulations Ubuntu Installation is Complete

COMP1047 – Computer Networks – Lab 1.2

Configure Ubuntu

School of Computer Science
University of Nottingham Ningbo China

Introduction

This tutorial assumes you've installed Ubuntu, either directly on Windows 10 as Windows Subsystem for Linux, or on a virtual machine hosted by VirtualBox running on MacOS X or Windows 10. See the tutorial "Install Ubuntu on Windows 10 and on Virtual Box": **available on moodle**

Install developer tools

An Ubuntu distribution does not include developer tools such as the GNU **gcc** and **g++** compilers and GNU **make**, so you must manually install them.

Open a terminal window in Ubuntu and enter the following commands:

```
sudo apt update sudo apt upgrade
 sudo apt install build-essential
```

(Use **apt-get** for older versions of Ubuntu before 16.04.) See https://help.ubuntu.com/community/InstallingCompilers

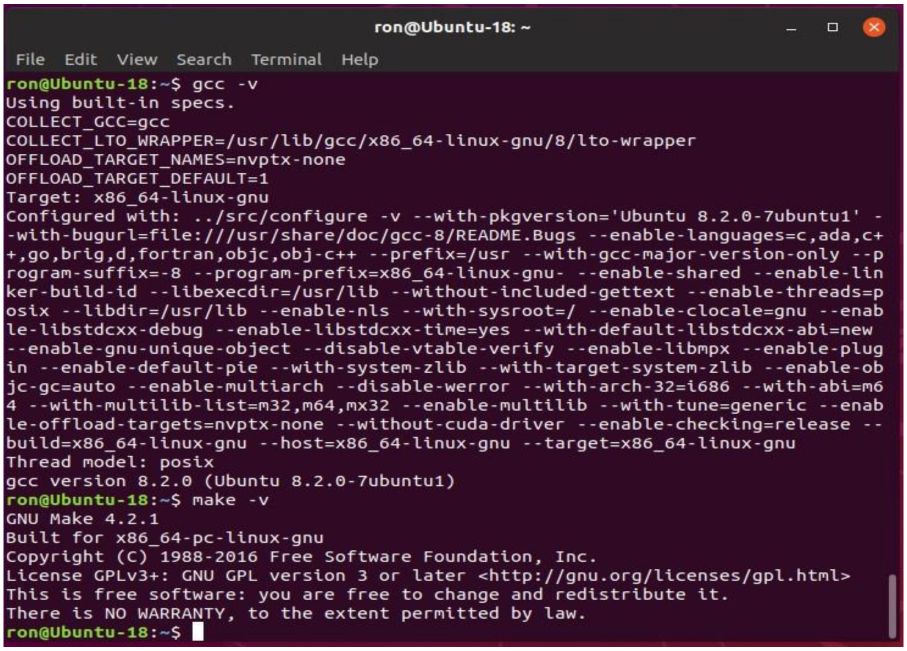
Note: If you receive the error message "Temporary failure resolving us.archive.ubuntu.com", the fix is to enter the following command:

```
sudo dhclient -v -4
```

and then try again. See https://askubuntu.com/questions/884604/temporary-failure-resolving-usarchive-ubuntu-com-live-usb-rescue.

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 1):

- **gcc -v   g++ -v**
- **make -v**



```
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:~$ make -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:~$
```

Figure 1. Verify that gcc, g++, and make were properly installed.

Useful Ubuntu utility programs

The GUI-based file browser named **Nautilus** allows you to view and manipulate files in the Ubuntu file system. If it is not already installed, enter the following command into the Ubuntu terminal window:

sudo apt install nautilus

To run Nautilus, open another Ubuntu terminal window and enter the command **nautilus**

(You want to run this in another terminal window because you want Nautilus to run as a separate process. Also, Nautilus outputs warning messages that you can usually ignore but are distracting if they appear in your main terminal window.)

The text editor named **gedit**, provides a GUI to create and edit text files. If it is not already installed, enter the following command into the Ubuntu terminal window:

- **sudo apt install gedit**

To run Gnu editor (**gedit**) and edit an existing text file, enter the **gedit** command followed by the name of the file, or just **gedit** if you want to create a new text file. For example:

- **gedit   .bashrc**

COMP1047 – Computer Networks – Lab 1.3

School of Computer Science
University of Nottingham Ningbo China

Installing NS-2.35 on Ubuntu 14.04

This post will tell you how to install NS-2.35 in Ubuntu 14.04 64-bit operating system.

NS-2 is a discrete event simulator targeted at networking research. NS-2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks.

Step 1 : Download NS-2.35 from here or here or from the web using any search engine.

Step 2 : Copy the downloaded file from the Download folder to your home directory /home/username folder (your username)

Step 3 : Open your Terminal and execute these commands one by one. If you have already issued these commands in Ubuntu configuration phase then skip it.

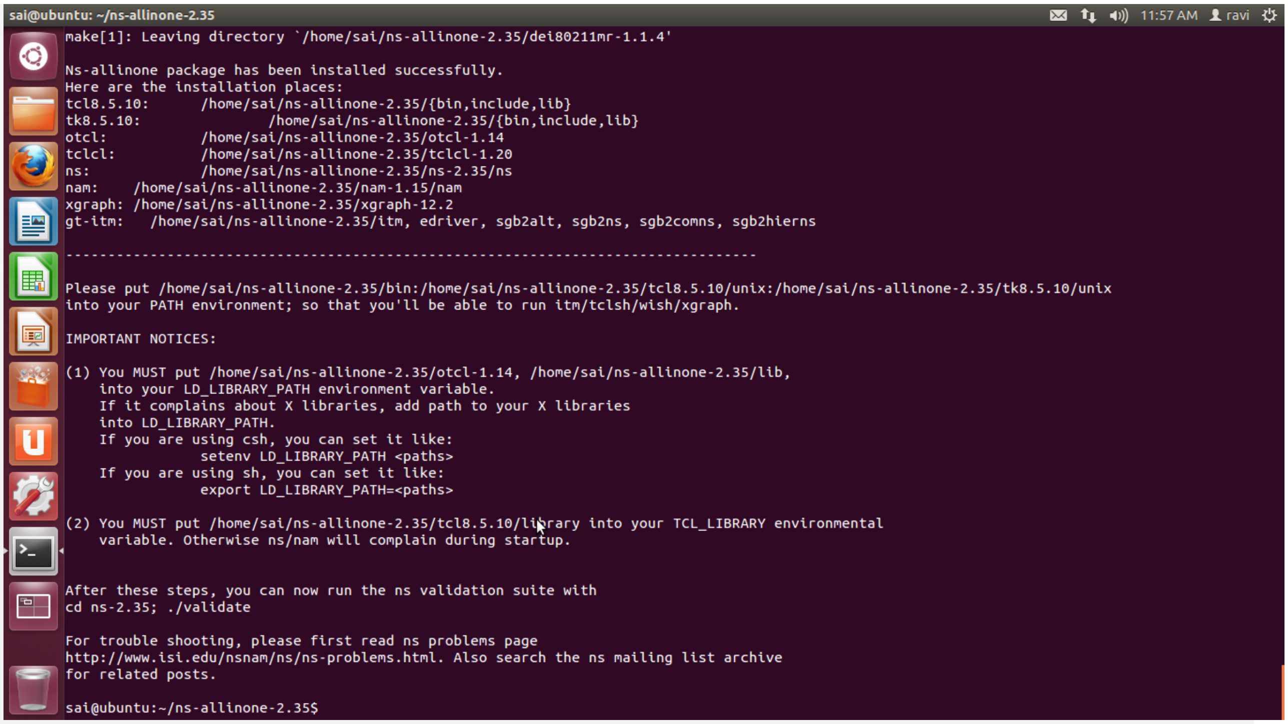
```
sudo apt-get update

sudo apt-get install build-essential automake autoconf libxmu-dev
```

Step 4 : untar ns-allinone-2.35.tar.gz using the command below

```
tar zxvf ns-allinone-2.35.tar.gz
```

Step 5 : Go inside ns-allinone-2.35 by using the command `cd ns-allinone-2.35` from your terminal and enter the command `./install`

Your installation will start and it will take some time for complete installation



```
make[1]: Leaving directory `/home/sai/ns-allinone-2.35/dei80211mr-1.1.4'

Ns-allinone package has been installed successfully.
Here are the installation places:
tcl8.5.10:      /home/sai/ns-allinone-2.35/{bin,include,lib}
tk8.5.10:               /home/sai/ns-allinone-2.35/{bin,include,lib}
otcl:           /home/sai/ns-allinone-2.35/otcl-1.14
tclcl:          /home/sai/ns-allinone-2.35/tclcl-1.20
ns:             /home/sai/ns-allinone-2.35/ns-2.35/ns
nam:    /home/sai/ns-allinone-2.35/nam-1.15/nam
xgraph: /home/sai/ns-allinone-2.35/xgraph-12.2
gt-itm:   /home/sai/ns-allinone-2.35/itm, edriver, sgb2alt, sgb2ns, sgb2comns, sgb2hierns

----------------------------------------------------------------------------------

Please put /home/sai/ns-allinone-2.35/bin:/home/sai/ns-allinone-2.35/tcl8.5.10/unix:/home/sai/ns-allinone-2.35/tk8.5.10/unix
into your PATH environment; so that you'll be able to run itm/tclsh/wish/xgraph.

IMPORTANT NOTICES:

(1) You MUST put /home/sai/ns-allinone-2.35/otcl-1.14, /home/sai/ns-allinone-2.35/lib,
    into your LD_LIBRARY_PATH environment variable.
    If it complains about X libraries, add path to your X libraries
    into LD_LIBRARY_PATH.
    If you are using csh, you can set it like:
                setenv LD_LIBRARY_PATH <paths>
    If you are using sh, you can set it like:
                export LD_LIBRARY_PATH=<paths>

(2) You MUST put /home/sai/ns-allinone-2.35/tcl8.5.10/library into your TCL_LIBRARY environmental
    variable. Otherwise ns/nam will complain during startup.


After these steps, you can now run the ns validation suite with
cd ns-2.35; ./validate

For trouble shooting, please first read ns problems page
http://www.isi.edu/nsnam/ns/ns-problems.html. Also search the ns mailing list archive
for related posts.

sai@ubuntu:~/ns-allinone-2.35$
```

Once the installation is over, you need to set the path for NS-2.35

Step 6 : The installation is over, now you need to check whether NS2 is working or not. In order to check type the following commands in your terminal

Enter `ns` in your terminal, if `%` symbol comes after you press enter, your installation is successful.

Enter `nam` in your terminal, if a Nam window pops out, Nam is installed successfully.

Step 7: Go inside ns-allinone-2.35/ns-2.35 by using the command `cd ns-allinone-2.35/ns-2.35` from your terminal and enter the command `./validate`

This command will validate your `ns` installation.

Congratulations NS-2 Installation is Complete

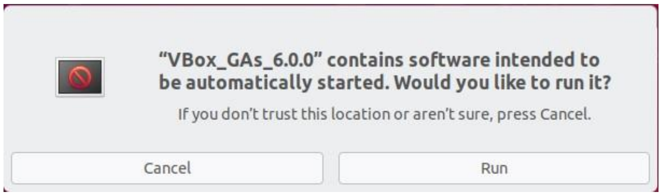
Install guest additions (Not part of Lab activity)

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 17).

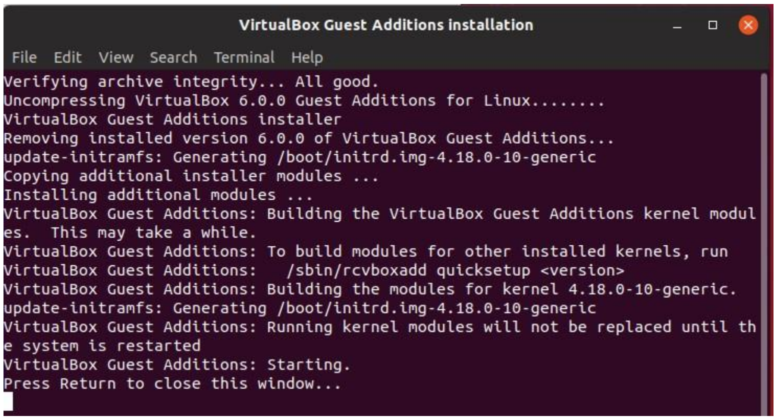


"VBox_GAs_6.0.0" contains software intended to be automatically started. Would you like to run it?

If you don't trust this location or aren't sure, press Cancel.

Figure 17. Click the Run button to install the guest additions.

Click the *Run* button and the software will install (Figure 18).



VirtualBox Guest Additions installation
File Edit View Search Terminal Help

```
Verifying archive integrity... All good.
Uncompressing VirtualBox 6.0.0 Guest Additions for Linux........
VirtualBox Guest Additions installer
Removing installed version 6.0.0 of VirtualBox Guest Additions...
update-initramfs: Generating /boot/initrd.img-4.18.0-10-generic
Copying additional installer modules ...
Installing additional modules ...
VirtualBox Guest Additions: Building the VirtualBox Guest Additions kernel modul
es.  This may take a while.
VirtualBox Guest Additions: To build modules for other installed kernels, run
VirtualBox Guest Additions:   /sbin/rcvboxadd quicksetup <version>
VirtualBox Guest Additions: Building the modules for kernel 4.18.0-10-generic.
update-initramfs: Generating /boot/initrd.img-4.18.0-10-generic
VirtualBox Guest Additions: Running kernel modules will not be replaced until th
e system is restarted
VirtualBox Guest Additions: Starting.
Press Return to close this window...
```

Figure 18. 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 19). Click the *Run Software* button in the upper right.

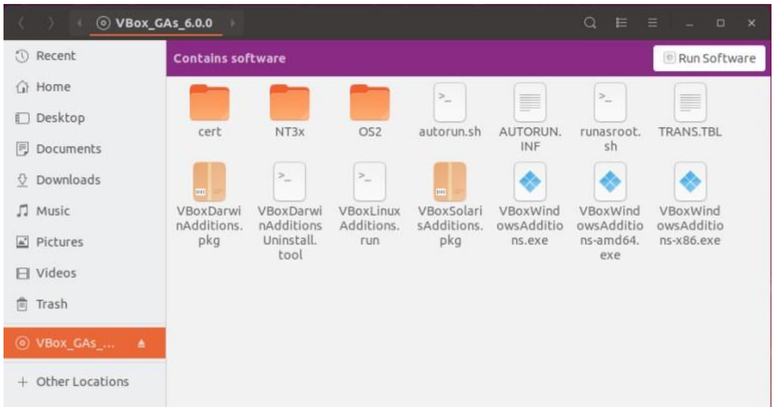
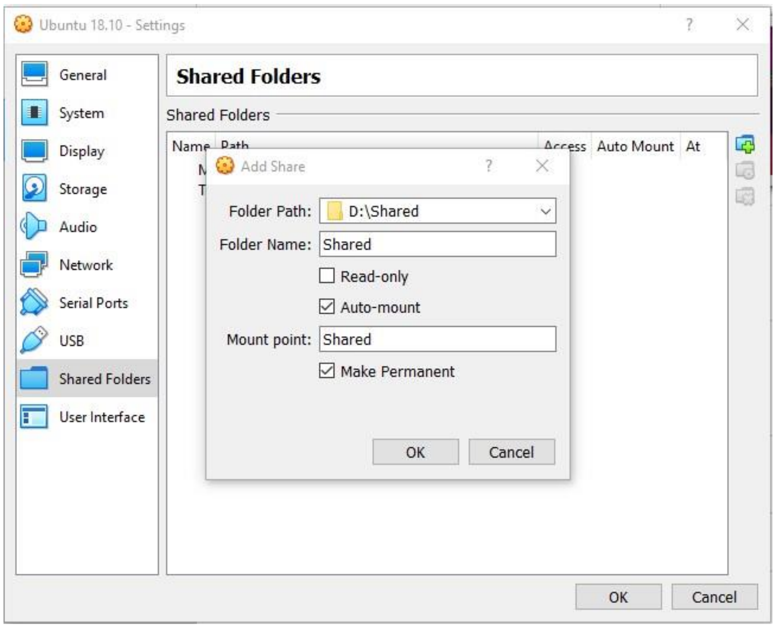


Figure 19. Contents of the guest additions virtual disk.

Create and mount the shared folder

A shared folder is a directory on your host operating system (i.e., Windows 10 or macOS) 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.

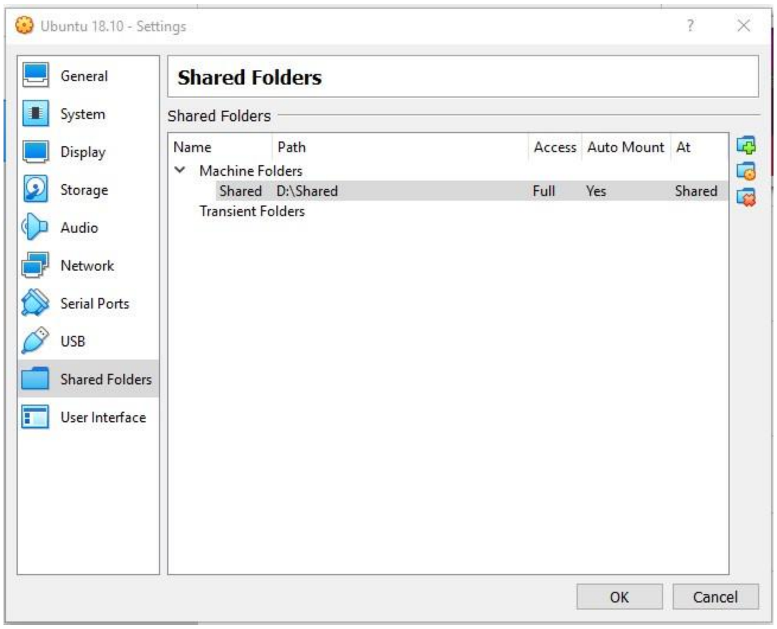
Create a directory in your host operating system that you wish to be shared, such as `D:\Shared` in Windows or `~/Shared` in macOS. In the VirtualBox main screen, select the Ubuntu guest operating system name and click the *Settings* icon at the top. Click on *Shared Folders* in the left panel. In the **Shared Folders** form (Figure 20), 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.



Folder Path: D:\Shared
Folder Name: Shared
☐ Read-only
☑ Auto-mount
Mount point: Shared
☑ Make Permanent

Figure 20. Shared folder dialog box entries.

Click the *OK* button to see your choices (Figure 21).



Shared Folders

Shared Folders

Name	Path	Access	Auto Mount	At
Machine Folders				
Shared	D:\Shared	Full	Yes	Shared
Transient Folders				

Figure 21. 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., **`unnc`**).

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. 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 (e.g., **`D:\Shared`** in Windows or **`~/Shared`** in macOS) is shared with folder **`Shared`** in your Ubuntu home directory.

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

```
ls shared
BigPi.cpp  BigPi.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. Be especially careful typing the **echo** command, particularly the double **>>** symbol. If you accidently 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**. Since both you and **sf_Shared** are in the group **vboxsf**, you are able to add, remove, or edit files in that directory.