Lab #1: Installing and using Linux in a Virtual Machine

Introduction

In this lab we will set up a virtual machine and install Linux in it.
We will review basic features of the command line using the **`bash`** shell.

Learning Objectives

1. Learn how to create a virtual machine running Linux
2. Review and understand basic features of bash.
3. Learn how to use basic commands that monitor the operating system.

Preparing for the lab

Look up these wikipedia articles on Unix, Linux, Windows, and macOS
Note the differences between Unix, Linux, Windows and macOS.
Read the wikipedia article on Ubuntu, and look over the main Ubuntu website.

Look up, using Wikipedia or a Google search, the meanings of the terms “hard disk partitions”, “virtual memory”, “swap space”, “virtual machine” and “bash shell”. Make sure you have a basic understanding of what these terms mean.

We will use the free virtualization software, VirtualBox, to set up an Ubuntu Linux system within a running Windows or Mac OS system. We will download and install VirtualBox and the Ubuntu distribution of Linux on your laptop or desktop computer.

Lab Activities

Report due 11:59 pm Sept. 10th

Use a laptop or desktop with 30+ GB of free disk space and 4+ GB RAM.

Part I: ***Installing and starting an Ubuntu VM*** (15-30 minutes)***:***

First, install Virtual Box.

VirtualBox is a software package for setting up virtual machines. VBox allows you to run a “guest” OS in a virtual machine that runs on a “host” computer. It is free and available here: https://www.virtualbox.org/wiki/Downloads . If you have a Windows computer, download the one for “Windows hosts”.
Some terms used by VirtualBox:

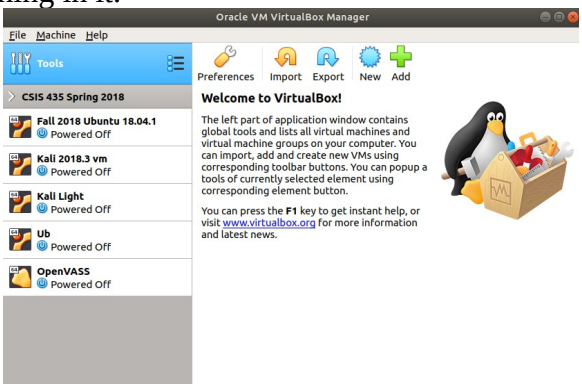
- **Host**: The operating system that will run VirtualBox. If you are using your laptop, your laptop (Windows, Mac OS X or Linux) is the “host”. If you are using the lab computers, the (Windows) workstation is the host.
- **Guest**: The OS that runs inside VBox. In this course, we will use an Ubuntu guest OS.

When using your own computer, **download** the **appropriate** version of **VirtualBox** and **VirtualBox Extensions** for your host system. Run the VBox installer. If you have Hyper-V on your system, you may run into problems when running VBox; disable or remove Hyper-V. If you have problems running VBox despite not having Hyper-V, try the free version of VMware.

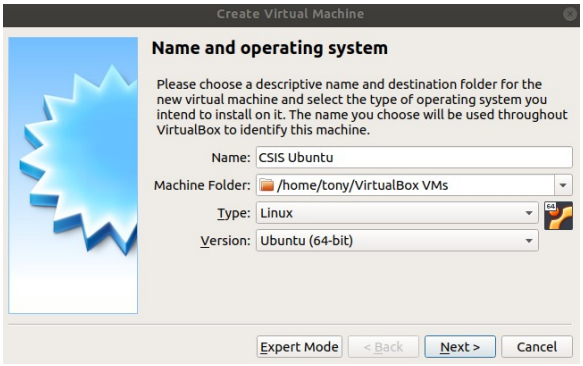
Ubuntu: Download Ubuntu Desktop (20.04 LTS) `.iso` file from https://www.ubuntu.com/download/desktop. This is a 2.6 GB file that we will use through VBox. Remember the location of the downloaded .iso file.

We are now ready to create a virtual machine with Ubuntu running in it.

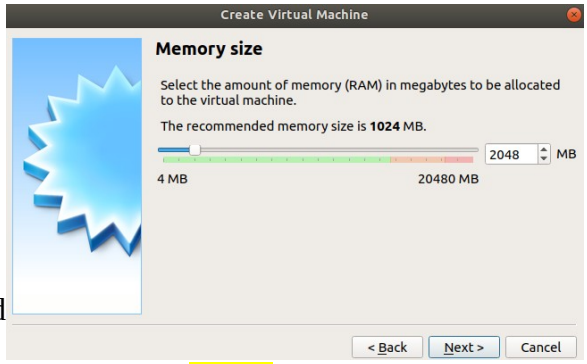
1. Run VBox. You should see the window shown here:



2. Create a new Virtual Machine by clicking on the New button. You will be asked to enter a name and the type of OS. Use the name "CSIS Ubuntu". "Linux" for type and "Ubuntu (64 bit)" for version as shown below

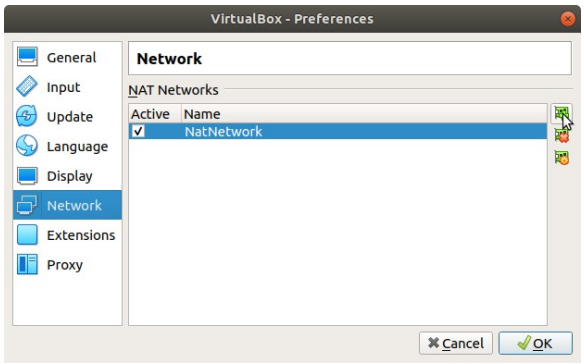


3. Click Next to continue and you will be asked to decide on how much memory to assign to this new VM. The minimum needed is 512 MB. The more (2GB would be great) the better but using too much will slow down your computer. Click Next after you make your choice.

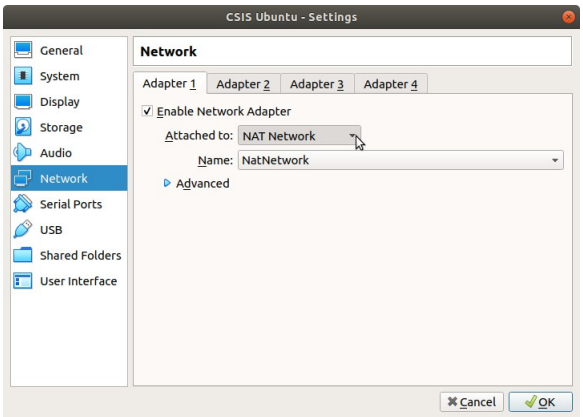


4. Next, set up a virtual hard drive within your current drive: choose the "Create a virtual hard drive now" option and click Create. Choose VDI for the format for the virtual HD and click Next; choose "Dynamically Allocated" to make the virtual HD a dynamic virtual HD rather than a fixed size and click Next. And then we have to choose a name and a size – the default name is fine and it would be a good idea to allow the virtual hard drive to get to a max size of about 20 GB – if you have the space to spare. Click Create. At this point, a VM has been created with a virtual hard drive (a file in your host system).

5. Some VM settings can be changed even when the VM is running. One thing you may want to change now is the Network settings. First, we will set up an internal "virtual network" that VirtualBox VMs can use. Click File → Preferences → Network → Add button → OK. This will add an internal virtual network named "NatNetwork".



Next, we will "attach" the virtual network card of the VM to this new virtual network: click on Settings → Network → choose for the "Attached to" setting.

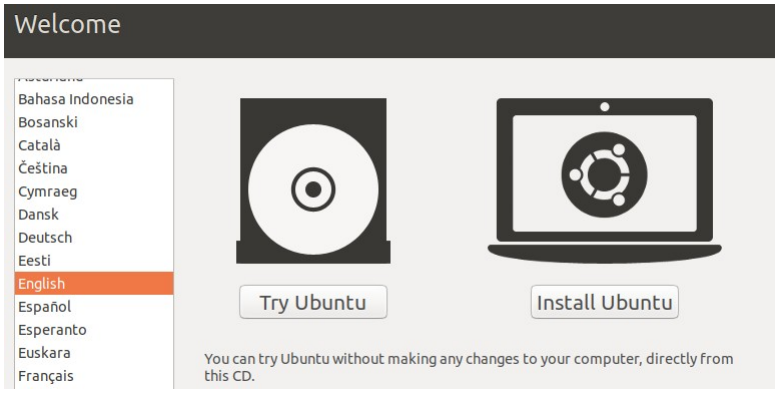


When using your laptop with campus wifi, note that current UWRF network security policies may not give VMs direct access to the campus wifi network. In such situations, the NAT Network setting allows us to access the internet from our VM and it will look to UWRF as if we are accessing the internet from the laptop itself.

If you are using your computer in your own network at home or in a cafe (but not the campus wifi), you can also use the network adapter of your VM in Bridged mode. Select **Bridged Adapter** in your VM's Network Settings and this will make your VM appear as a new computer on your physical network.

6. You will need to know where your downloaded Ubuntu .iso file is as we will use it from within VirtualBox in this step. Click on Settings→Storage and, in the Storage Tree pane, click on the "Empty" disk: In Attributes, click on the disk icon and select "Choose a virtual optical disk file" as shown here. Browse to where you saved the Ubuntu .iso file, select it, and click Open. Make sure to select the "Live CD/DVD" box. Click OK.
   At this point the .iso file should be connected as if it were a bootable DVD. So, click the Start button (with the green arrow). If everything works, a window should appear with Ubuntu booting up.

7. Eventually, you should see the "Welcome" screen for the Install process. Select "Install Ubuntu". This will install Ubuntu on the 20 GB virtual drive we set aside earlier. The install will happen in a number of steps.



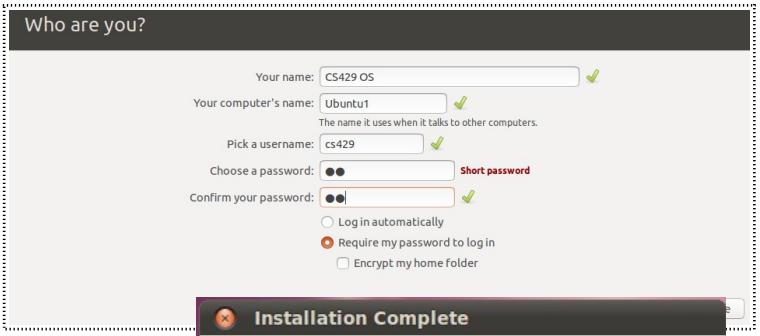
   For the keyboard layout, choose English (US) and click Continue.

   In the "Updates" window, choose "Normal installation", "Download updates while installing" and "Install this third-party software" and click Continue.

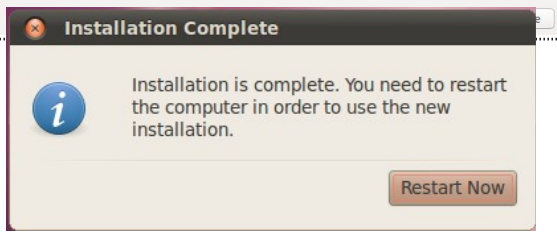
   In the "Installation type" window, choose "Erase disk and install Ubuntu" and click Install Now. You may get a warning about writing the changes to disks – click Continue. For time zone, choose Chicago or Central and select Continue.

8. The next step will set up a user account. You should see a screen that asks you to enter a name, password, etc as shown below. Enter **CSIS OS** for Your Name and enter a name for the "computer" - actually the VM. Use the username **csis** and a suitable password.

   Choose "Require my password to log in", click Continue and you should see a screen with a progress bar indicating where you are in the install process. It should take about 10 minutes.



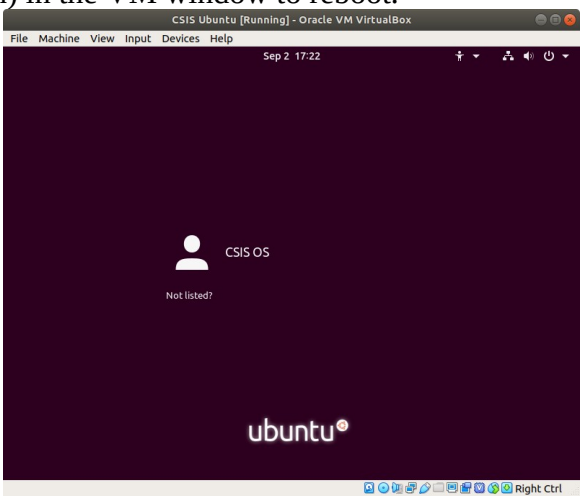
9. When the installation is complete, you should see saying that we need to restart:



   Click the Restart Now button and, if everything works, you will be asked to remove the installation media from the drive and press ENTER. What we want to happen now is that we should restart the system and have it boot from the "virtual hard drive" where we installed Ubuntu.

   The "installation medium" is actually the .iso file and we will want to remove this file from the VM so that the system does not reboot from the .iso "virtual dvd". Remove the .iso file from the VM by clicking Settings → Storage → choose the ubuntu "dvd" under the IDE controller →

click on the DVD and switch it to your Host DVD drive if you have one or else choose Remove Drive and choose OK. Then press ENTER (or Return) in the VM window to reboot.



At this point, if everything worked, Congratulations! You have successfully installed a Linux operating system in a virtual machine!

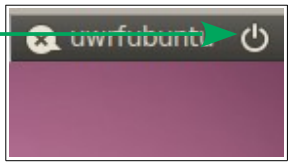
If the installation was successful, you should see a login window with "CSIS OS" as a login option. Go ahead and log in with the password you set up earlier.

Choose Devices → Insert Guest Additions CD Image and when asked whether you want to run the program, choose yes. This will start a "Terminal" window with outputs for the Guest Additions installation and will take a few minutes but eventually it will print a message saying that we should press Return. Do so and after this we should be able to resize the VM window so that it is larger.

Find the System Monitor application (click Activities and enter "System") and run it. If System Monitor is not found, we can install it: start a Terminal window (click Activities to find it) and install System Monitor using the commands `sudo apt update; sudo apt install gnome-system-monitor`

Use the command: `gnome-system-monitor` and if necessary, use `sudo snap` to install any required packages. Note the details of what System Monitor tells you about the Resources – how many CPUs, how much memory – and File Systems – how many devices and their types and sizes – for this installation – include this in your report.

To quit the Linux VM, locate the "on/off button" in the upper-right corner of the screen, clicking on it should give you a menu of choices: choose Shutdown … .



One of the aims of this lab is to ensure that you have the experience of installing Linux, especially if you have never done this kind of thing before. You want to be confident enough to be able to repeat this installation procedure either on another computer or a virtual machine.
Try out the various applications that are available to you – note that the networking may not be configured and so a web browser may not show you any useful information.

Problems running the Linux VM using VirtualBox? Try the free version of VMware. Other notes:

1. When you have all the settings right, the VM should be as responsive (fast) as a "real" system. If it isn't, you may have to change some of the settings. Try shutting down the system and playing with the settings.
2. You may be asked to update the system – go ahead and do it.
3. If your VM "display" remains small, install the "**Guest Additions CD image** ..." (in the Devices menu). You should be able to resize your window after this. You may need to install a C compiler before Guest Additions can install; use these commands: `sudo apt update ; sudo apt install build-essential`
   Then re-try installing Guest Additions
4. If you are planning on using USB 2 devices, download and install the Oracle VM VirtualBox Extension Pack from the VirtualBox download page.
5. Antivirus software on Windows systems have blocked some VMs from starting. Take a look at how people are fixing this on the VirtualBox forums.
6. Ethan tried the above steps out and said it worked for him. His suggests:
   i. If the VM does not boot, change the number of CPUs for the VM to be at least 2
   ii. If the VM is unstable, set Video Memory of the VM to be 128 MB
   iii. Enable the Shared Clipboard to allow copy/paste between the VM and the host.

Part II Using the `bash` shell (30 minutes)

Open a Terminal Window: Search the "dash" icon in the upper left corner for "terminal" and click on the Terminal icon – this should bring up a terminal window.

The terminal window has a `bash` shell in which you can try out Unix shell commands. Enter the following commands:

- **sh**<*Enter*>
- **ps**<*Enter*>
- **bash**<*Enter*>
- **ps**<*Enter*>
- **pwd**

  What can you say about the output of the above 5 commands?
- **ls**
- **ls -l**
- **ls -al**

  What is the difference between the above 3 commands?
- **cat > test.out**
  abcdef <*Enter*>
  <*Ctrl*>***d***
- **ls -l**

  What did the cat command do?
- **cp test.out test2.out**
- **ls -l**

  What did the cp command do?
- **rm test.out**
- **ls -l**

  What did the rm command do?

- **LS**
- **Ls**
- **Pwd**
- **PWD**

  What happened? Does this tell you anything about Unix commands?

- **mkdir testdir**
- **ls -l**
- **cd testdir**
- **pwd**

  What did the mkdir and cd commands do?

- **mkdir testd2**
- **ls -l**
- **mv testd2 test1d**

  What did the mv command do? Why?

- **rmdir testd2**
- **ls -l**
- **rmdir test1d**
- **ls -l**

  What did the rmdir commands do?

- **date**
- **cal**
- **man cal**

*The above command should show some information about the cal command. Use **Space** to page down. Enter a '**q**' to quit the man command.*

- **cal -3**
- **cal 3**

  Can you figure out how to print out the calendar for the entire year? Include this in your lab report.

- **who**
- **whoami**
- **id**

  Include the outputs of the above 3 commands in your lab report.
- **top**

  You can quit the top command by entering a "q".

At this point, the installation (part 1) should be done. Finish up the install – step 7 of part 1.

Part III Navigating the file system using the `bash` shell (20 minutes)

1. Log in as `csis`, open a new Terminal window and look up the **`man`** page for the **`script`** command by entering:
   - `man script`

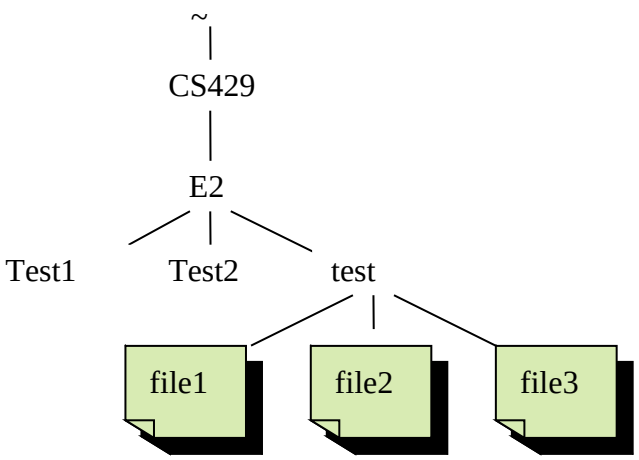
   Use the space bar to page through the output and enter a 'q' to get out of the man manual page. What does the script command do? Next, try the following commands:
   - **mkdir CS429**
   - **cd CS429**
   - **mkdir E2**
   - **cd E2**
   - **pwd**

     What does the output of the `pwd` *tell you about the commands you entered before it?*
   - **ls -l**
   - **ls –l ..**
   - **script test.out**
   - **pwd**
   - **cd ~**
   - **pwd**
   - **whoami**
   - **id**
   - **who**
   - **date**
   - **ls -l**
   - **ls –l ..**
   - **ls -l ../..**
   - **exit**

     Make sure you can understand the result of the above sequence of commands. The script command generates an output file that is called a "typescript" file.
   - **pwd**
   - **ls –l**
   - **cat test.out**

One way to transfer files from the VM is by enabling USB access to your VM – look in Settings to figure out how to do this. Copy the contents of the **`test.out`** file to a USB flash drive using the graphical file manager called Nautilus: click on Places in the top panel and then Home Folder. Include the contents of this file in your report. Use the `Courier` font for the outputs of commands and make sure that this is readable and well formatted.

2. Now that you know how to use "**`script`**", set up a single typescript file that shows how you performed the following tasks:
   i. Enter the command:
   ```
   script part2
   ```
   ii. Create the directory scheme shown below in your "home" directory (~):



   Use the cat command to create the files: file1 and file2. To create a file using the cat command, first enter:
   ```
   cat > file1
   ```
   - then enter some text that you want to put into the file, and when you are done,
   - hit return after the last line and then
   - *<Control>d* to stop the cat program. Use the **`ls -l`** command to check if this worked.
   - You can create a "copy" of a file using the **`cp`** (copy) command:
   ```
   cp file1 file2
   cp file2 file3
   ```
   iii. Starting at your `CS429` directory, **recursively** list all files and directories
   iv. Remove the files: `file1` and `file3` and the directory `Test2`. Go back to your `CS429` directory and **recursively** list all files and directories in it.
   v. Stop the **`script`** command using the **`exit`** command and use the **`cat`** command to list the contents of the file that **`script`** generated. Save the output of the **`script`** command for your lab report. Use the Courier font for the outputs of commands and make sure that this is readable and well formatted.

3. cd to your home directory. If you look at the output of the '`ls -l`' command, a typical text file may be listed as:
   ```
   -rw-r--r--  1 csis csis  15 2020-08-23  file1
   ```
   Reading left to right, the first '–' indicates that this is a regular text file. If you have a directory in your current directory, it will appear something like:
   ```
   drwxr-xr-x  2 csis csis 120 2020-08-23  test
   ```
   The first '`d`' indicates that this is a directory file, which is usually called a directory.

Although you may be used to referring only to things you can, for instance, print, as files, Linux and Unix treats many things as files including: directories, keyboards, mice, monitors, printers, network connections, etc. These "devices" will have different letters to signify that they are not regular files.
Change to the **`/dev`** directory and list all the files there. What kind of "files" does the **`/dev`** directory have?

4. Change back to your home directory and use the '**`ls -al`**' command to list all the files there. Then use the '**`file`**' and '**`stat`**' commands to get more information about the files `.bashrc` and `.profile` and the directory `Documents`. Include this information in your report.

5. Make sure that the *`~/CS429/E2/test/file2`* file exists and that you are in your home directory. Then execute the following commands:

```
ln ~/CS429/E2/test/file2 c1
ln -s ~/CS429/E2/test/file2 c2
```

What did the above commands do?
Use the '**`ls -l`**', file, and **`stat`** commands to look for as any differences between `c1` and `c2` as you can find; in your report, write down all the important differences you observed.

One place where we can check the meanings of the different options for commands is: explainshell.com – enter your command in the site and it should give you a short explanation of commands and also of each of the options you enter. Try the "`ln -s`" or "`tar -xvfz`" commands in this site.

6. Keep the files, directories, etc. that you created above for the next lab. Shutdown the VM – it may take up a good part of your system's memory!

Part IV Monitoring the Linux system and installing programs from the command line (60 minutes)

Netflix engineers use Linux VMs and to quickly assess the performance of a VM, they use a list of commands as described in http://techblog.netflix.com/2015/11/linux-performance-analysis-in-60s.html

Note that this blog uses a fixed width font (like `Courier`) to make it clear which words are commands so that we don't read them as ordinary text. We will emulate this style in our reports and use for all commands and command outputs.

1. Log in as **`csis`**, open a new Terminal window and look up the **`man`** pages for the following commands and try running each one. If they are not available on your VM, you will have to install it as described below:
   - `ps`
   - `pstree`
   - `top`
   - `htop` – see below for notes on installing this
   - `vmstat`
   - `df`
   - `du`
   - `iotop`
   - `iostat`
   - `lsof`
   - `netstat`
   - `vnstat`
   - `nethogs`
   - `monit`

2. For example, we can view the man page for **`ps`** command by entering:
   ```
   man ps
   ```
   Use the space bar to page through the man page and use the "q" key to quit the man command.

3. Installing programs: To install a program or a command we can use the **`apt`** command. First, update the list of packages for your system using the command:
   ```
   sudo apt update
   ```
   The `sudo` command runs the apt command as the super-user or administrator. Use your **`csis`** password when it asks you to enter a password. This command may result in a lot of text scrolling past – you do not have to read all of this. Once this is done, install a program using the command (the following command will install htop):
   ```
   sudo apt install htop
   ```
   If you are asked whether you are sure you want to proceed, enter the "y" key for "Yes". To install the `iotop` command, use the command:
   ```
   sudo apt install iotop
   ```
   and so on. If a command replies that you need root privileges, prepend the `sudo` command, for example:
   ```
   sudo iotop
   ```
   This runs the iotop command as the super user "root"

4. For each of the commands from `ps` to `monit` listed above, use your own words without copying and pasting from the man pages or elsewhere to describe briefly (25 words or less) in your report what each command does, indicate whether you have used this command in the past and how familiar you are with it.

Lab 1 Report

*Turn in (using the Canvas dropbox) a list of the steps you used for the Linux (Ubuntu) installation and **all answers** to all questions after the Prelab. This report will probably be about 10 or more pages long.*