Homework 1
Welcome, Setup, and Some Light Reading

EECS 201 Fall 2020

Submission Instructions

Answer the bolded text that begins with a "**Q**". This assignment is an "online assignment" on Gradescope, where you fill out your answers directly.

Preface

Homework in this class can sometimes be a little underspecified. You are expected to Google, to try things, and to fail from time to time. Making mistakes is highly encouraged, it's how you learn. We have many office hours if you find yourself getting stuck, but we will always start with the questions, "What have you tried so far?" and "Why do you think that didn't work?"

1 Set Up an Ubuntu Virtual Machine

While I mentioned in the first lecture that I'm fine with more or less any *nix system, I will require you to keep an Ubuntu virtual machine on hand. Not every *nix system is built the same way, so not *everything* will be 100% compatible. Ubuntu 20.04 will serve as the golden standard of behavior for the assignments in this class. While most things should work the same way on mac OS and Ubuntu, sometimes when POSIX (the standard for *nix systems) does not specify a particular behavior for particular tools, the behavior of the tools may differ.

One of the reasons is for this is that mac OS uses the BSD versions of classic *nix tools while Linux usually uses the GNU versions. BSD and GNU may have their own extended behavior in their implementations of classic *nix tools.

What a virtual machine does is provide you an emulation of a computer system. In the context for this class, the virtual machine is an emulation of a traditional desktop/laptop computer. This allows us to install "guest" operating systems on them, mess with them, and wreck them without affecting the original "host" operating system. By having a virtual machine in this class, you can freely explore the *nix environment with no danger to your "host" system.

1. Download a copy of the **Desktop** version of **Ubuntu 20.04**. This will come in the form of an *.iso* disk image file. Traditionally, you'd install an operating system with a CD or DVD; these files represent the data on those.

2. Download and install the latest version of VirtualBox. While I say this, feel free to use another virtualization software that you prefer like VMWare Player for the rest of the class, but don't expect official support on it from the instructional staff. Just bear with me on this assignment and use VirtualBox. When you're done with the assignment, feel free to uninstall it and go back to whatever you were using before. *If you are on Windows and using WSL2 or some other Hyper-V application, the latest version of VirtualBox should work alongside it now. However, if you do run into issues, first, try going into your VM's System settings > Acceleration > Paravirtualization Interface, and set it to Hyper-V. If it still doesn't work, try taking a look into disabling Hyper-V when using VirtualBox. Here's a workaround to turn off/on Hyper-V on StackOverflow, using Command Prompt in Administrator mode. Make sure to restart your computer for this to take effect.*

3. Set up a new virtual machine.

   (a) Note that the drop-down menus for operating systems don't install the operating system; they just set up the virtual machine with some default settings and give it a cool icon. Select the "Ubuntu (64-bit)" drop-down item.

   (b) For the most part the defaults are fine. Note that the Ubuntu download page also provides some nice recommended system specs.

   (c) Ubuntu 20.04 requires at the *minimum* 2 GiB (2048 MiB) of RAM, so set the memory size to at least 2 GiB. If you can spare the RAM on your computer, I'd go with 4 GiB.

   (d) I recommend setting at least 2 CPUs to make things a bit more zippy.

    (e) Note that RAM and CPU count can be configured after setup.

    (f) The default hard drive size of 10 GiB can work, but is a tad small, leaving only about 3 GiB left on a Minimal Installation. Try maybe 20 GiB or even 50 GiB. By default, disk images are *dynamically allocated*, which means the disk image will grow on demand when more space is needed until it hits the limit you have set.

    (g) After setting up go to your VM's Display settings and max out "Video Memory". This will allow for fancier graphics like larger display resolutions to work.

4. Install Ubuntu on your new virtual machine.

    (a) Remember what I said about *.iso* files? Our VM has a virtual CD/DVD drive. Go to your VM's Storage settings and click on the Empty CD under "Controller: IDE". On the right hand side, under "Attributes" click on the CD. This will allow you to navigate and find that Ubuntu *.iso* file.

    (b) Start up your VM!

    (c) I recommend doing a "Minimal Installation" since you probably won't be consuming media on this virtual machine. Feel free to do a normal one if you do want to see what the full "Desktop Linux" experience is like.

    (d) I recommend "Downloading updates while installing".

5. Once Ubuntu is running, install the Guest Additions: check VirtualBox's Devices menu → Insert Guest Additions CD Image, then either let the disk auto-run or run the disk manually when it's recognized by Ubuntu. You might get a message about gcc, make and perl not being installed. This means you need to install those utilities. Bring up a Terminal window (right click on the desktop, look for the application, explore! There's also a keyboard shortcut ;)) and run `$ sudo apt install gcc make perl`. This will invoke the package manager, APT, to retrieve those software packages. You can then go open the disk and click the "Run Software" button to run the installer again.

    Q: What are Guest Additions? What do they do? What changed after you installed them? (If nothing happens, try resizing the window and try maximizing the window, or try full-screening. If something still doesn't happen, turn off the machine and starting it again. If *still* you don't notice anything, turn off the machine and change the "Graphics Controller" option to something else in your VM's Display settings and try again)

    Q: What is the output of `$ lsmod | grep "^vbox" | awk '{print $1}'`?

6. Play around with your new machine! Try installing stuff with APT, like Git: `$ sudo apt install git`. Try writing and running a Hello World program. What about other tools you've used before? If you have any, can you get an old course project running?

    Q: What does `sudo` do?

7. It may be helpful to setup some shared folders so you can share data between the host (your computer) and guest (your VM) operating systems. If you go to your VM's settings, there should be a "Shared Folders" section. Here you can pick out what folder on your host system to share. For a simple setup, you can select your folder, set "Auto-mount" on, and set "Make Permanent" (don't worry, you can turn this off or remove the folder sharing). In your VM, if you left the mount point blank, you can find your shared folder in `/media/sf_<your folder name>`. Try `cd`'ing or `ls`'ing it. You'll find that you don't have permissions! With `$ ls -l` we can check more info about that directory. The problem is that the user owner of the directory is `root` and the group owner is `vboxsf` ("VirtualBox Shared Folders"). This is a simple enough fix: you'll have to add your user to the `vboxsf` user group. I'll leave this as an exercise for you to figure out :) The changes are not immediate. You'll have to restart your VM. Once you're back in, try playing around with your shared folders :) If you're interested, you can find more info about VirtualBox shared folders here.

2 Readings

Each of these are short blog posts: 5-10 minute reads. These were selected to give you a little exposure to some varying perspectives. The authors, Joel in particular, have several other very interesting posts that you may find interesting. After each reading, write a response for the given question.

Biculturalism by Joel Spolsky
http://www.joelonsoftware.com/articles/Biculturalism.html

Q: Has your computing experience thus far aligned more with "Windows culture" or "Unix culture"? What makes you feel that way?

These two articles use the word "research" a lot, but the points made apply well to any work in computer science.

Helping my students overcome command-line bullshittery by Phillip Guo
http://www.pgbovine.net/command-line-bullshittery.htm
and the counter-point
On the value of command-line "bullshittery" by Eytan Adar
https://medium.com/@eytanadar/on-the-value-of-command-line-bullshittery-94dc19ec8c61

Q: What did you take away from these articles?

Q: How long did this assignment take in *minutes*?