

Homework 1

Welcome, Setup, and Some Light Reading

Due: Saturday, September 17, 10:00PM (Hard Deadline)

Submission Instructions

Submit this assignment on [Gradescope](#). You may find the free online tool [PDFescape](#) helpful to edit and fill out this PDF. You may also print, handwrite, and scan this assignment.

1 Set Up a Ubuntu Virtual Machine

One of the goals of this class is to understand systems work so that you can customize and improve them for yourself. On CAEN, course environments are already set up and everything “just works”. On a brand new Ubuntu install, however, we will have to find, install, set up and manage many tools ourselves.

Recall from lecture that a virtual machine (VM) is a fake computer running as a program. We’ll use a VM in this course as a playground to test things out and work without risking anything on your day-to-day machine. To kick things off, we start by getting a basic environment set up this week.

One final thought: Homework in this class will often 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. Get a copy of the **Desktop** version of **Ubuntu 16.04** (this is a big download, consider doing it on campus).
2. Download and install [VirtualBox](#).
3. Open VirtualBox and create a new virtual machine. Most of the defaults are fine. The default hard drive size of 8 GB is a little small, I recommend going bigger (50 GB or so). By default, disk images are *sparse*, which means it won’t take 50 GB of real disk space to create a fake disk, rather the fake disk will grow on demand as it’s used, so there’s not a lot of harm in choosing a big number.
4. Install Ubuntu on your new virtual machine. I recommend “Downloading updates while installing”.
5. Once Ubuntu is running, install the Guest Additions (try VirtualBox’s Devices menu → Insert Guest Additions CD Image; you’ll need to reboot once this finishes).

Q: What are Guest Additions? What do they do? What changed after you installed them and rebooted your VM?

I didn’t notice anything that happened other than the mouse being easier to move around. The system also seemed to be a little faster than before. As such, I decided to Google an answer! Guest additions are system drivers and applications that were added to our VM Ubuntu to make it run smoother and better. The purpose is to increase the performance of the OS within the virtual machine. There are a variety of things that get installed listed: <https://www.virtualbox.org/manual/ch04.html>.

6. Play around with your new machine! Try writing and running a Hello World program. What about other tools you’ve used before? Can you get an old course project running? How is it different than a CAEN environment?

I don’t find it too different than a CAEN environment! Emacs isn’t initially installed though. Everything seems to work fine!

2 Readings

Each of these are short blog posts, 5-10 minute reads. I selected these to give you a little exposure to some varying perspectives. The authors, Joel in particular, have several other very interesting posts that I highly encourage exploring. 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?

I've had a mixed experience so far. Before I become a developer/started the CS major, I was definitely more aligned with the Windows culture. I preferred having a system where everything just worked. I didn't really care about the ins and outs of development - i.e. I was an end user. However, since I've actually started doing development work, I'm definitely more aligned with unix culture. I'm not an "Aunt Marge" anymore, and I find that I rarely care about building a GUI. Because I enjoy solving algorithmic problems, I prefer just having a program that does what I need it to do. Thus, there really isn't a purpose in building graphical user interfaces. I enjoy using Bash and Sublime, but granted, my opinion is solely formed off my experiences with Unix. I've never actually looked into doing development on Windows, so maybe I have inherent biases toward unix.

GO UNIX!

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?

I found both articles to be interesting reads. Whereas the article by Phillip Guo focused on the value of actual research and trying to get past command-line bullshittery, the article by Eytan Adar countered by saying that part of learning and growing is through struggling with command line. I find myself more in alignment with the points made by Adar - progressing through command line bullshit is not really meant to be procedural. I think it's supposed to be a challenge, and once you make many mistakes, you start understanding where your problems are coming from. In my own life, I've always found that when I actually fail at something, that's when I learn the most. When people tell me how to do something, I rarely retain that knowledge. That being said, command line sucks. I struggle a lot with it, but I see its value, and hopefully I can learn to do more with it by the end of this class!