Alan Macklin

Computer Science Bachelor of Science

Professor Julie Henderson

December 19, 2024

Problem Statement:

When anyone wants to learn how to use Linux, there is very little anyone can do to learn the basics of CLI because none of the resources make much sense initially. Specifically, as a student, I have also found the “introduction” information about Linux in the Charleston Southern University (CSU) curriculum somewhat lacking in terms of raw material. It does not help that it is possible to take Survey of Scripting Languages, which seems to be the primary Linux introduction class, after classes that require you to use Linux VMs such as applied networking. In most classes that require programming, especially classes that use C/C++, it is suggested that you use Linux because you have access to tools like “makefile.” In each of these classes, you are required to either know how to get all the tools you need or just copy and paste the correct commands into your terminal. Despite the fact that manual or man pages are a pretty core part of the Linux experience, they were only briefly explained one time throughout my entire undergraduate studies.

One of the primary spots Linux finds its presence in the curriculum is in NETLabs. As a Linux user, these NETLabs committed numerous mistakes in each lab that I could nitpick. As an example, in the processes NETLab from Operating Systems, many of the commands were explained like this: “Use the command XXX, because it does XXX,” or “Now add X operator/flag to command XXX because it allows for functionality XXX.” Since all that was required of me was a screenshot, I felt that these NETLabs had the potential to be a place for real learning but missed explaining the important parts of each tool or even how to find information in a man page. It felt like wasted potential to only require screenshots.

Using guides for Linux seems like a good place to learn the ins and outs of Linux, but often these guides assume you know more than you do. So when you look up guides, you end up copying and pasting commands you barely understand until you get something that works. Both of the options of making walk-throughs and guides end up being really bad for newer users.

Project Description:

This project is modeled after what I would consider to be an “ideal” NETLab for learning. To briefly summarize what a NETLab looks and feels like, it is a step-by-step walkthrough of how to do a certain task or use a certain command. At certain steps, you take a screenshot of the result of a command and save it in a document. When you get to the end, you turn in proof that you have typed out all the commands. I want to turn this walkthrough style into a guide, or in a sense, I want to make a guided walkthrough of Linux. By combining the strengths of guides with a walkthrough, the weaknesses of both should be addressed.

This project is targeted at people with minimal to no experience in the Command Line Interface (CLI). The guided walkthrough hereby referred to simply as the guide, is to be designed to cover all the commands that I have used extensively over my Linux experience as well as important commands to be able to identify. This will include some basic commands from binutils, grep, ping, and package managers, among other things. The guide will go through some commands, briefly explaining how to use each one. In every couple of sections, there will be an exercise that requires you to interface with a virtual machine or computer. There will also be sections that require reading man pages to find specific parts. The rm command comes to mind for having a couple of particularly important flags to understand. As such, after having a detailed explanation of the command and the flags, I would then have them go to the man page to see what the man page says about -rf. The last section should have a way to test how well the users can figure out which command to use and when. They should also have to read the man page to figure out what different flags do and where to interface with the commands in different aspects.

Proposed Languages:

Bash, Vim script

Proposed Packages:

binutils, tree, bash, netutils, xcfe4, cowsay, fortune, cmatrix, lynx, neofetch, sl, vim, nano

Additional Software:

Debian.iso

Personal Motivation:

I am interested in convincing all my friends that Mac OS and Windows are terrible operating systems run by bad companies. Linux, on the other hand, is open-source and free. This makes it much safer overall because trusted distributions are not going to collect and sell your data like Windows or Mac can. Even if you are worried about a Linux distro stealing your information, they are open source, so you can check it yourself.

Schedule:

9/11 - Starting Point

9/25 - Practical tools

10/10 - Freeform

10/16 - Proposal and Deployment

10/30 - First round of editing and skilled Linux user review

11/20 - Second round of editing, testing complete data collected

11/27 - Processed data and presentation complete