David Giacobbi
CPSC-334: Final Project Reflection

The techniques and tools I used to convert the MyString C++ Library into a CI/CD pipeline involved the following:

1. **GitHub Actions** for workflow to build, test, and package my project on every push.
    a. **YML files** to set up workflows
    b. **Makefile** to organize build, test, and packaging scripts
    c. **Shell Scripting** to run commands to execute each job
    d. **Debian Packaging** to determine where files should be added on a new machine for my project to work as well as what commands should be run upon "dpkg"
2. **Google Cloud Platform** to set up my self-hosted runner on a virtual machine to run my workflow; also set up a VM to deploy my package on as well
3. **Docker** to containerize my workflow jobs while they are being run

The key steps I had to take to build the Debian package of my project were as follows:

1. **Build:** When creating a C++ library for installation, I learned about **.so files** which are a type of file that can be compiled with g++ specifically for use as a library in different projects.

2. **Test:** Similar to my systemd assignment from this semester, I continued to use **googletest** to run tests on my various string functions that I created. It was important to keep in mind which libraries were required for this to run in a new container during my workflow.

3. **Package:** Once the proper elements were built and tested, I simply had to move the configuration files into the **bin/DEBIAN** directory during my package creation. The most important one was the **postinst** script which needed to update the dynamic linker cache so that my library would be recognizable when a user on the new machine used it in their own C++ programming. This was done after the **.so file** for my library was moved into the **/usr/local/lib** directory and the corresponding **.h file** was moved into **/usr/local/include**.

The largest challenge of this new project was trying to understand how a package would make sense for a project that was different than a system service. I did not want to simply move a compiled C++ file into a new machine that could only print "hello world". Rather, I wanted my project to have a little more practical use. I wanted to learn how libraries can be added to a new machine for development use, which involved me learning more about **.so files** which help shrink the size of **.cpp library files**. This installation process is something I have not worked with before so it took some time to understand what was necessary for my project. Additionally, I continued to encounter package management issues in my containers which required more tests before getting it right. However, doing this again a second time made it pretty easy to debug.

The more I work with these DevOps methodologies; it has become more apparent how simple it can be to spend an extra 5-10 minutes to set it up in a project for the future. I have always struggled with keeping up with my testing and ensuring my project continues to work before pushing it to GitHub. However, larger projects such as Senior Design have created headaches for my team as we lacked DevOps organization and ended up spending sprints trying to debug problems, we weren’t aware of when pushing new code or trying to merge a new branch. Practicing these methodologies more will help streamline project work even though it feels like there is more work up front to set it up because it will help me remain confident in my working product as it grows and expands in features.

This class has helped me to gain a lot of confidence and understanding in the information technology sector of Computer Science. I have always felt like I lacked understanding of how my own machine works, often just believing commands work because they work rather than questioning how. I am leaving this course with a heightened curiosity of how things work within my OS as well as a newfound appreciation for the simplicity of Linux. In the course of a couple of months, I have found myself quicker to orient myself around the command line rather than some of the GUIs I used to have to rely on. Most importantly, I have more confidence and trust to manipulate parts of my system without the fear that I am going to destroy my computer. Thinking about the readings we did at the beginning of the semester, I realize the power of simple tools and methodologies as well as how important it is to take the time to not just use them but understand them.