

## Techniques:

- Shell scripts – build\_deb.sh created to handle building the Debian package, adapted from systemd homework
- Self-hosted runners – used the Google Compute cloud VM to run workflow jobs
- Makefile – used to shorthand compilation and running, adapted from systemd homework
- GitHub Actions – created a workflow with jobs to run on a self-hosted runner

## Steps taken to build the release package:

The build\_deb.sh script handled the Debian packaging similarly to my build\_deb.sh from the systemd homework over the semester. The script cleans any existing build, then rebuilds and copies the binary into the correct directory. It creates the control file, handles permissions with chmod for all the project files, and builds the Debian package.

## Challenges during the project:

The biggest challenge I faced during the project was trying to create more tests in tests.cpp. I have never written any C++ unit tests before and tests.cpp was created by my group members so I tried to make test functions like JTest for Java but it didn't end up working. Every test I tried adding caused the test job to fail because of the expected output so I just left the original tests as is for the project.

## DevOps methodologies evaluation:

DevOps methodologies will help spot and tackle any testing problems early through the development stage. Automating the testing process would eliminate any testing problems caused by human error and would only be caused by the code being tested or from the workflow being misconfigured. The continuous testing allows frequent stable releases and helps ensure that the final production release is high quality.

## Reflection on how the course influenced my perspective on software development:

The course showed me how valuable testing while developing really is. It changed how I tested code because I was used to only testing by compiling and running programs with no exposure to unit testing. It taught me how development and operations are integral with each other and that they are equally important for any kind of engineering. The tools used in the class like Docker and Kubernetes are some that I will probably use for senior design to streamline the process and so that I don't get gray hairs and feel only regret.