George Calvert
Dr. Crandall
Final Project Reflection
4-30-25

For this project I aimed to convert my mypl project from OPL that created my own programming language from the ground up through python, to a CI/CD Debian package that creates a terminal tool for all users. To achieve this, I utilized a Makefile that allowed build automation that runs scripts to build a tmp directory to package, build the Debian file, lint the Debian file, and finally test the mypl.py script through pytest. To build this Debian package, first you must run ‘make build’ to prep the directories and files to package, then ‘make build-deb’ must be run to actually package the files/directories created in the last step, then finally ‘dpkg -i mypl.deb’ must be run to install the necessary files into the right locations for the mypl.py executable to be seen by all users to run mypl files developed by said users. To test this process, I utilized a self-hosted runner that ran a github workflow to build, test, build-deb, lint-deb, and finally checked to see if the runner can utilize the tool. Additionally, I added another workflow that is run when a tag is pushed that builds the Debian file and then saves that artifact on the repository. Some challenges that I encountered when converting this old project included path management, which included ensuring that all users were able to utilize this tool with no problems, and then dependencies, like which python interpreter pytest versions. Other than those small issues, I didn’t encounter anything that truly set me back in my development of this project. Overall Devops philosophies can benefit future projects by providing automation, for example through the makefile and workflows, and then since the build process is structured it would enable me to easily reproduce and create consistent package creation. More generally, I believe this course has greatly impacted my knowledge towards software development. This has been evident in projects, like this one, which have allowed me to get hands on experience with packaging and its management, thus allowing a greater understanding of the Linux architecture. To add on, this course has caused me to shift focus from just writing and testing my own code to delivering functioning software in a packaged, tested, and documented manor. Additionally, the emphasis on the importance of reproducible builds and deployments has shifted my thought process when developing and turning in assignments. Before this would be a pump and dump kind of project where if it worked and passed tests then that means it works and is submitted, now this has turned into a more well-rounded approach where, added onto the past requirements, this needs to be reproducible and easily followed by any developer. With this it has also highlighted the value of automation through the lifecycle of development. For instance, instead of running commands through repeatedly typing this will now be turned into automation either through scripting or automation tools like workflows.