Sravya Kuchipudi

Professor Kaisler, Hasanov

CSCI 6917

May 30, 2023

Project Proposal: Building a GUI for Unix Commands Execution on MacOS

The goal of this project is to provide a straightforward graphical user interface (GUI) that enables users to choose Unix commands from menus, enter the required parameters, run the commands, and show the results. This GUI will offer a simple and user-friendly alternative for MacOS's command prompt for executing Unix commands. The project will also use PySimpleGUI which is a Python GUI that transforms other GUI frameworks into a more straightforward interface.

Currently, executing Unix commands on MacOS involves the user utilizing the command prompt or Terminal application to input their commands. While this method is powerful and widely used, it requires users to have knowledge of command syntax and can be intimidating for users who lack experience with it. It also is a purely text-based approach meaning it lacks a visual interface, which makes it challenging for users to discover and select commands that they may want to use. To improve the current approach, the project will develop a graphical user interface (GUI) that makes it easier to run Unix commands on MacOS. Users using the GUI will see a menu-based interface that allows for simple browsing and selection from a list of accessible commands. Also, the GUI will prompt users to input any arguments needed for the selected command, guaranteeing proper execution. Our solution will be more approachable and user-friendly by offering a visual interface, even for people with no or little command-line expertise.

The work done in this project will benefit a wide range of users, including both new and experienced users who want to leverage the ability of Unix commands on MacOS. New users will find it easier to explore and use Unix commands through the intuitive GUI, while experienced users may appreciate the convenience and efficiency of a graphical interface for executing commonly used commands. Additionally, individuals transitioning from other operating systems to MacOS will find our GUI particularly helpful in adapting to the Unix command environment while adjusting to the new system.

While many errors that will occur in the project will be difficult to predict before beginning the hands-on work, there are still a few potential risks associated with this project that can be considered ahead of time. Due to my varying levels of experience with different parts of this project I first need to consider that troubleshooting errors may take longer, and therefore I will have to vigilant in my research to balance my knowledge. More specifically in terms of implementation, the security of the approach is important since there needs to be awareness of not introducing any security vulnerabilities when executing commands with the GUI. Also, if any errors do occur when the user inputs their command, there needs to be a set way to handle errors with clear error messages to the user.

The out-of-pocket cost for this project is low since the hardware and software (MacBook Pro laptop with MacOS software) are already acquired and the current plan is to use open-source packages as needed. The main reason there would be any other costs would be if any reference material is needed to be purchased for better understanding. This project should be able to be completed in 11 weeks as outlined in the tentative weekly schedule. The plan is to use the first few weeks to research GUI frameworks, design the overall layout and structure of the GUI, and implement the menu and ability to take input. Then in the next few weeks, there will be the work done on the functionality of the GUI and command execution. Finally, the last few weeks will be used to improve on the overall aesthetics and usability as well as including additional commands/functions to the GUI so that it is ready for the final demonstration.

By the midterm check in, the plan is to have a functional GUI prototype that supports a subset of commonly used Unix commands on MacOS. The prototype should include the menu-based command selection, argument prompts, command execution, and output display functionality. Basic error handling and user feedback mechanisms should also be implemented. This will allow a simple demonstration of the project to show the GUI and how a few commands can run. For the final demonstration, there will be a presentation of a polished and fully functional GUI that supports a range of Unix commands commonly used on MacOS. The application should have an easy user experience, providing intuitive command selection, argument input, and output display. The demonstration will show how well the GUI handles both valid and invalid inputs and how the interface is formatted for ease of use.