Sravya Kuchipudi
Professor Kaisler, Hasanov
CSCI 6917
29 June 2023

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

The next part of the project was adding and improving different features in the graphical user interface GUI since there was testing capabilities to actually assess that the additions were working. There are a number of features that were important to add but some of the main ones were the separate output window, templates for commands, and additional functionalities.

One significant addition to the code was the introduction of a separate output window. Previously, the output was displayed directly in the main GUI window. However, displaying the output in a separate window offered a few advantages since it allows for a cleaner separation between input and output making it easier for users to focus on their commands and view the resulting output without any visual clutter. It provides more space for displaying the output compared to the limited area within the main GUI window, Also, it enables the user to interact with the main GUI while viewing the output simultaneously.

This output window was implemented by creating a new Toplevel window. Within this window, a Text widget was added to display the command output. The widget was also configured to be read-only which ensures that the output cannot be modified by the user either accidentally or otherwise. One small error that occurred when initially coding this was that each command execution led to a new output window being created. So, to ensure that each time a new command is run, its output is appended to the existing output window, the output_text widget was made a global variable so that it could be accessed and modified from the run_button_click function. Within the run_button_click function, the output window was configured to allow modifications, and the output of the command was appended to the output_text widget. After appending the output, the widget is set back to read-only mode to prevent any user modifications.

The command’s dictionary in the code was updated to include additional information for each command. Each command now contains a description and a template. The description provides a brief explanation of the command's functionality, while the template displays the command syntax, indicating how the command should be used with its respective arguments. These additions enhance the usability of the GUI by providing users with clear information on each command's purpose and usage.

The update of the command’s dictionary also led to the addition of a command information window. This window provides users with descriptions and templates for each available command, helping them understand the purpose and usage of each command reducing the likelihood of errors and facilitating their interaction with the GUI. While it isn’t overly informative, it serves as a quick reference and guide, improving the usability and effectiveness of the application. When the "Command Info" button is clicked, a new Toplevel window is created. Within this window, a Text widget is added that holds information about each command, including the command description and template for use. The information is retrieved from the commands dictionary in the code, which stores the details for each command.

Overall, these decisions were made with the aim of enhancing the overall usability, functionality, and user experience of the Unix Commands GUI. By considering user needs, interface design principles, and the specific requirements of a command-line interface in a graphical environment, these choices contribute to a more intuitive, efficient, and informative application. The addition of an output window allows for better separation of input and output, the command information window provides users with

valuable details about each command, and the updated functionality ensures that the output is appended to the existing output window for each new command execution. These improvements make the GUI more user-friendly and informative, facilitating the execution of Unix commands in a graphical interface.