Sravya Kuchipudi
Professor Kaisler, Hasanov
CSCI 6917
22 June 2023

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

One of the main aspects of this project is connecting a graphical user interface (GUI) to the Mac Terminal to allow for a seamless integration of user interactions and command execution. In the last week I worked on integrating a few commands into the GUI and then pairing the GUI to the Terminal.

Using help from the articles and readings provided by Dr. Kaisler as well as my own knowledge and research six different Unix commands were chosen to initially work with on the GUI. The first choice was ls which is used to list the files and directories in the current working directory. This was a necessary inclusion since it would help facilitate testing and checking the other commands. Then commands that would help manipulate directories were chosen. The cd command is used to navigate the file system by changing the current working directory in the Unix shell by taking the desired directory as an argument. The mkdir command is used to create directories or folders in the file system by taking one or more directory names as arguments and creates them in the current working directory. Finally, some commands that can be used on files or directories were chosen. The cp command is used to copy files and directories from one location to another by taking two arguments: the source item name and the destination name . The mv command is used to move or rename files and directories by taking two arguments: the source item name and either the new destination or item name. Lastly, the rm command is used to remove or delete files and directories by taking either the file name or -r directory name as the argument.

The next step was to integrate the GUI with the Terminal so the functionality could be tested. For this the subprocess module was chosen to provide a straightforward method to execute commands in the Terminal from the GUI application. By using the subprocess.run() function, input from the GUI can be passed as command-line arguments to the Terminal. Basically, this was done by creating a run_command function that takes a command as input, runs it in the Terminal using the subprocess.check_output method, and returns the output.

While it was not too difficult to code this, during the testing phase it was noticed that the commands were running as individual shells at each execution. Basically, if the cd command was run to change the directory and then the ls command was run to check for proper execution, the shell would still show the original working directory and not the selected directory. From here some research and experimentation was needed to figure out how the execution could be done properly. Instead of running each command as a separate process, a single shell session was tested to maintain the state of the current working directory using the initialize_shell() function, which creates a subprocess running the Bash shell. This approach kept causing the GUI to not respond and attempts at troubleshooting it were causing other errors so some more reading was done to find a different way which led to the os module.

The os module is a built-in python module that helps interact with the operating system. The os module was used in conjunction with the subprocess module to fix the issue with changing directories. When the command is cd, the os.chdir function is used instead of subprocess.run because changing the directory of the current process is not possible using subprocess.run alone. The os.chdir function allows the current working directory to be changed within the current Python process.

So far, the GUI has been created and linked to the MacOS Terminal such that a handful of commands can be successfully executed. From here it is likely that a good number of commands will be added in the next few weeks. Also, the appearance of the user interface is rather simplistic and blocky so

some changes may be made with that, either in enhancing using the current package or shifting with a different package to make for a sleeker look.