MINI-SHELL OS PROJECT

Group Members:

* Samama Farooq - 373810
* Sohaib Ahsan - 383611
* Allah Rakha – 379284

Section: BSCS -11B

Submitted To: Mam Farzana Jabeen

Date: 23-5-2023

1. Introduction

The Mini-Shell project aimed to extend the functionality of a basic shell by implementing additional commands. The important areas we aimed at in our project were related to networking, pipelining, and file handling. This report provides a comprehensive overview of the project, highlighting the approaches, commands implemented, their explanations, and the overall conclusion.

2. Project Objectives

The primary objectives of the project were as follows:

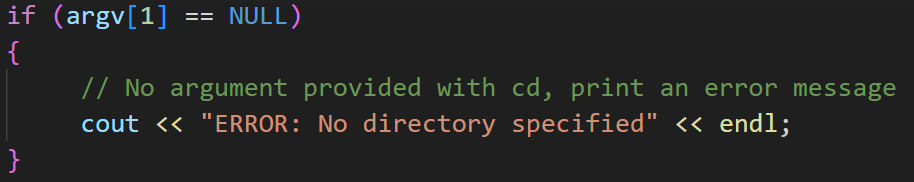
* Enhance the existing shell by adding new commands.
* Implement commands related to networking for tasks such as ping.
* Enable file handling capabilities like copying, appending, and removing files.
* Integrate pipelining functionality to allow chaining of commands.
* Implement our own defined commands to improve user experience.

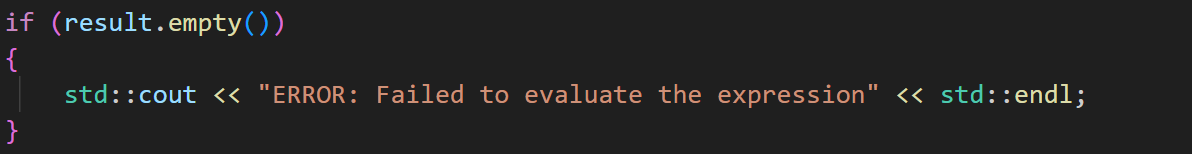
3. Project Repository

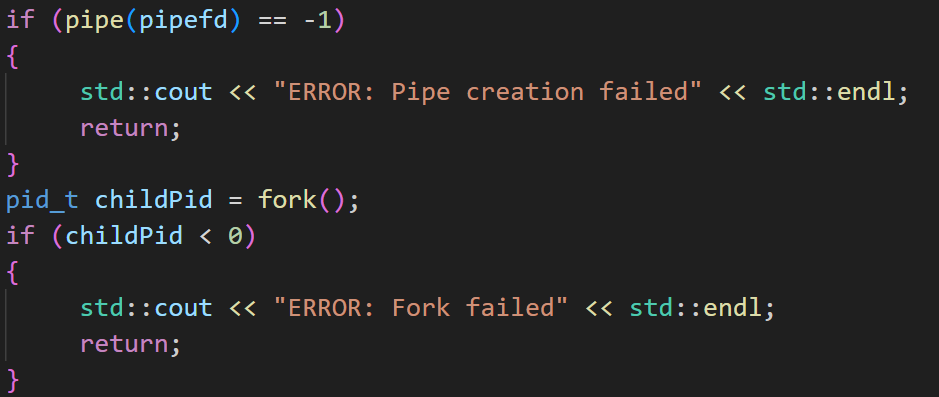
We used the second repository for our semester project. The purpose for using the second repository was although many functions were already implemented, we customized them using pipelining and other methods. Also, we created some new commands to allow better experience to the users.

4. Error Handling

The shell implemented error handling mechanisms to handle various scenarios, such as incorrect command syntax, invalid file paths, network connectivity issues, and command-specific errors. Error messages were displayed to the user to provide feedback and assist in troubleshooting.

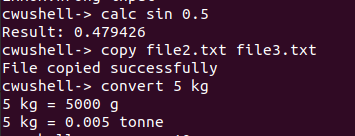


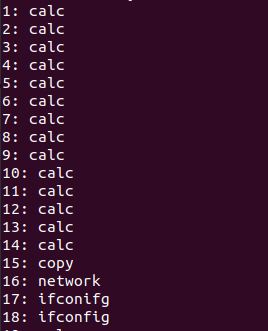




5. History Log

The shell maintained a history log of executed commands, allowing users to review and repeat previous commands easily. The history log provided a convenient way to track and refer back to past commands. Additionally, we also have a history command to display all the commands previously executed.



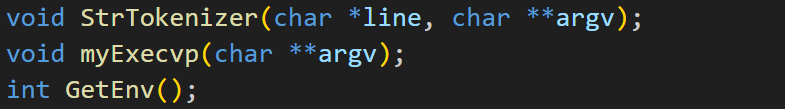


6. Project Approach

To achieve the project objectives, the following approaches were adopted:

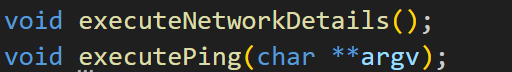
6.1 Shell Development

The initial focus was on developing the basic shell infrastructure, including user input parsing, command execution, and command history tracking. This formed the foundation for implementing the additional commands.



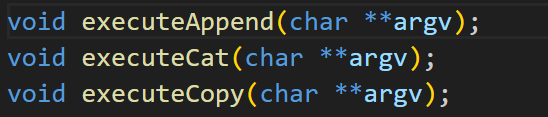
6.2 Networking Commands

Commands related to networking, such as 'ping', were implemented using system calls and appropriate libraries. The 'ping' command allowed users to test network connectivity by sending ICMP echo requests to a specified destination.



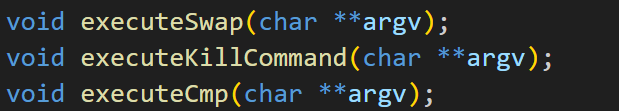
6.3 File Handling Commands

Commands like 'cat', 'copy', 'append', and 'remove' were implemented to provide file handling capabilities. These commands utilized file system APIs to perform tasks such as reading file contents, copying files, appending data to files, and deleting files.



6.4 Pipelining Commands

Pipelining was integrated to enable the chaining of commands. This feature allowed users to pass the output of one command as input to another, enhancing the versatility and power of the shell. The implementation involved managing inter-process communication and synchronization.



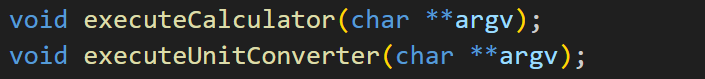
6.5 Web Browsing Command

The 'open' command was implemented to open a given URL on a web browser. This involved utilizing system commands or libraries to launch the default browser with the specified URL as a parameter.



6.6 Mathematical Commands

The ‘calculator’ and the ‘unit converter’ command uses basic arithmetic operations to produce the desired outputs for the users.



6.7 Random Library Commands

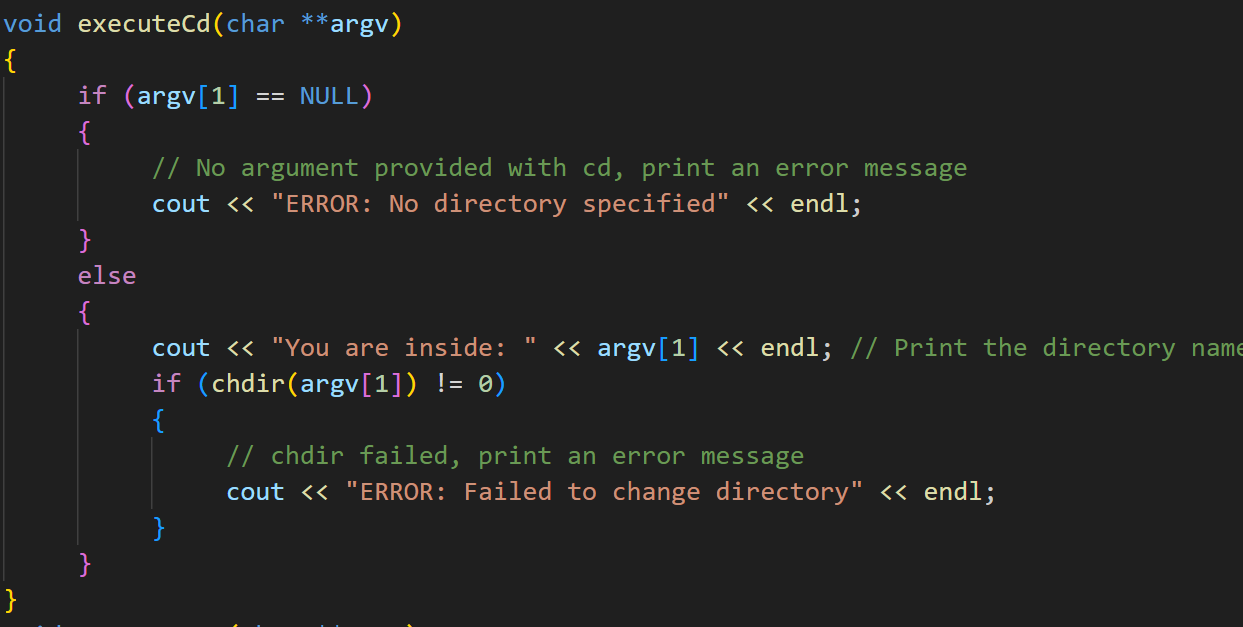
The ‘random password generator’ command uses the random library to generate random numbers, letters and symbols for the specified length of the password.



7. Implemented Commands

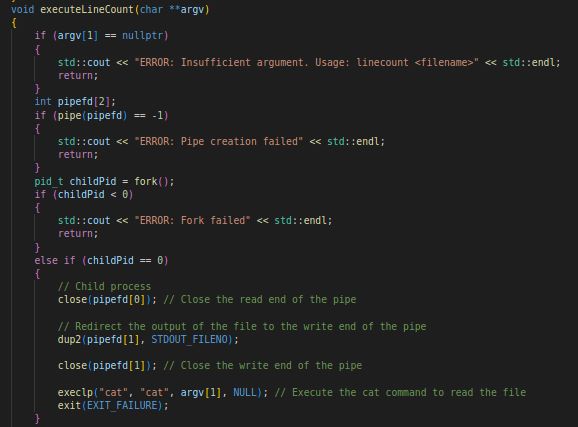
7.1 cd

The 'cd' command allows users to change the current working directory. It enables navigation within the file system.



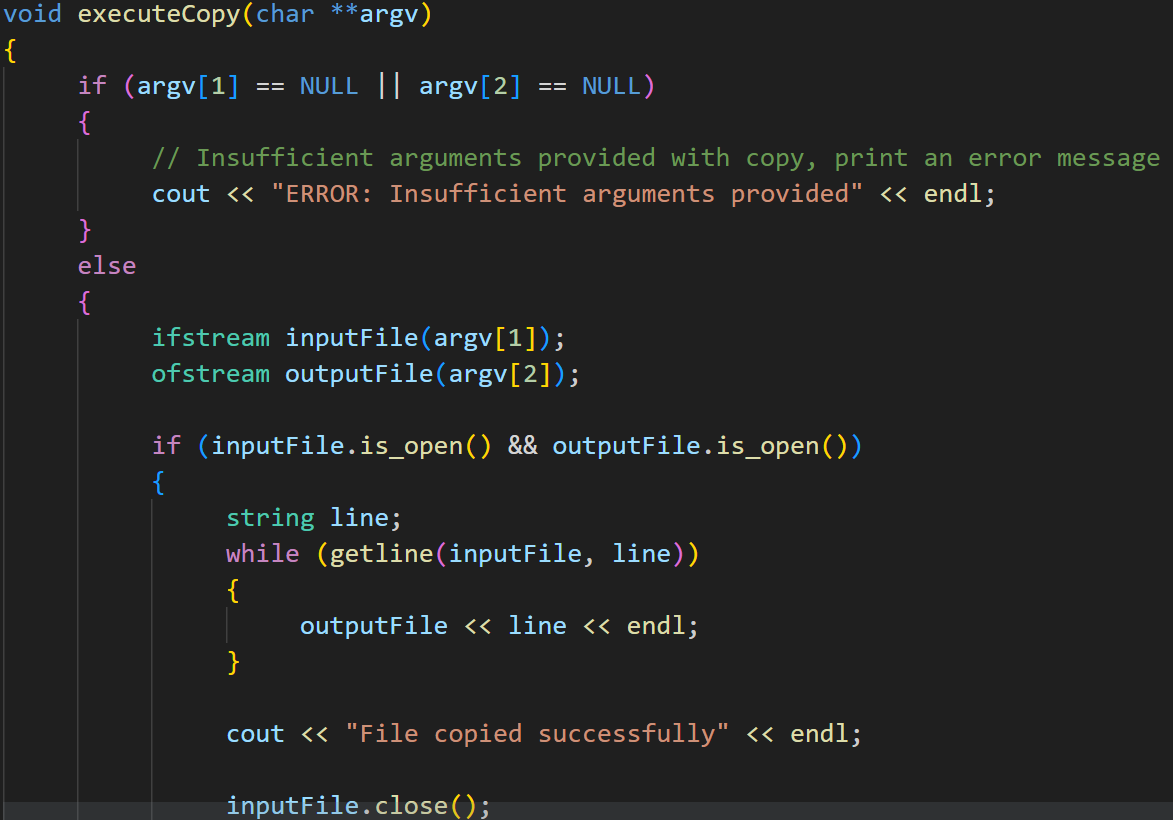
7.2 line count

The 'line count’ commands displays the numbers of lines present in the file specified.



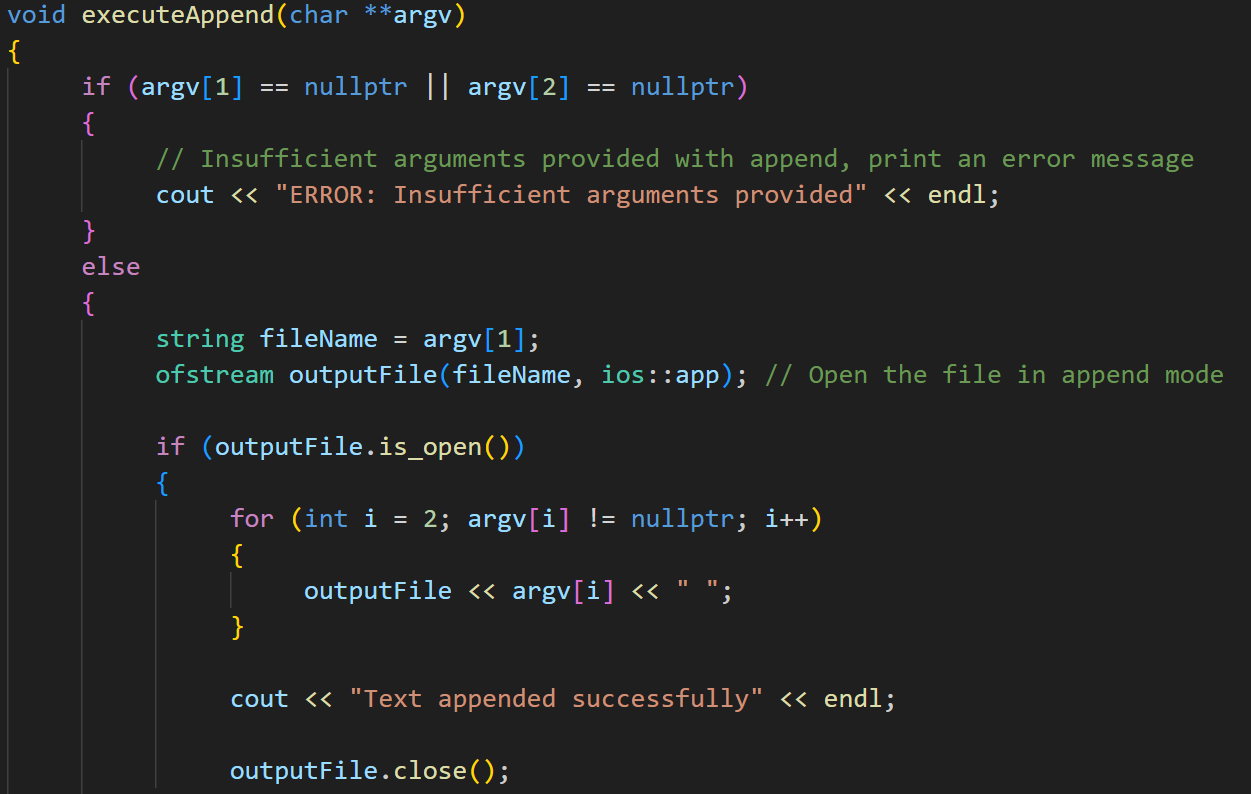
7.3 copy

The 'copy' command copies a file from a source location to a destination location. It creates a duplicate file with the same content.

****

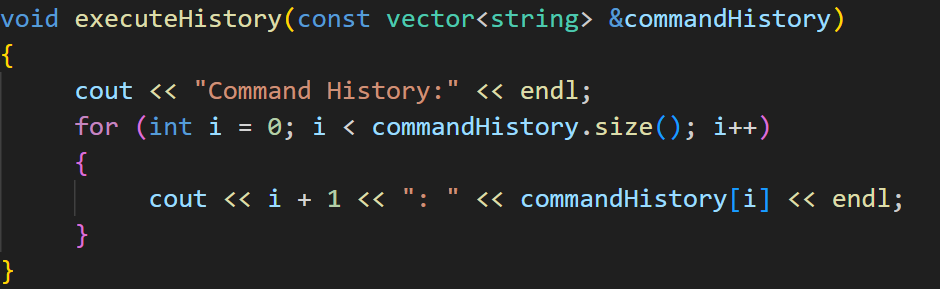
7.4 append

The 'append' command appends the contents of one file to another. It takes two file arguments and adds the content of the source file to the end of the destination file.



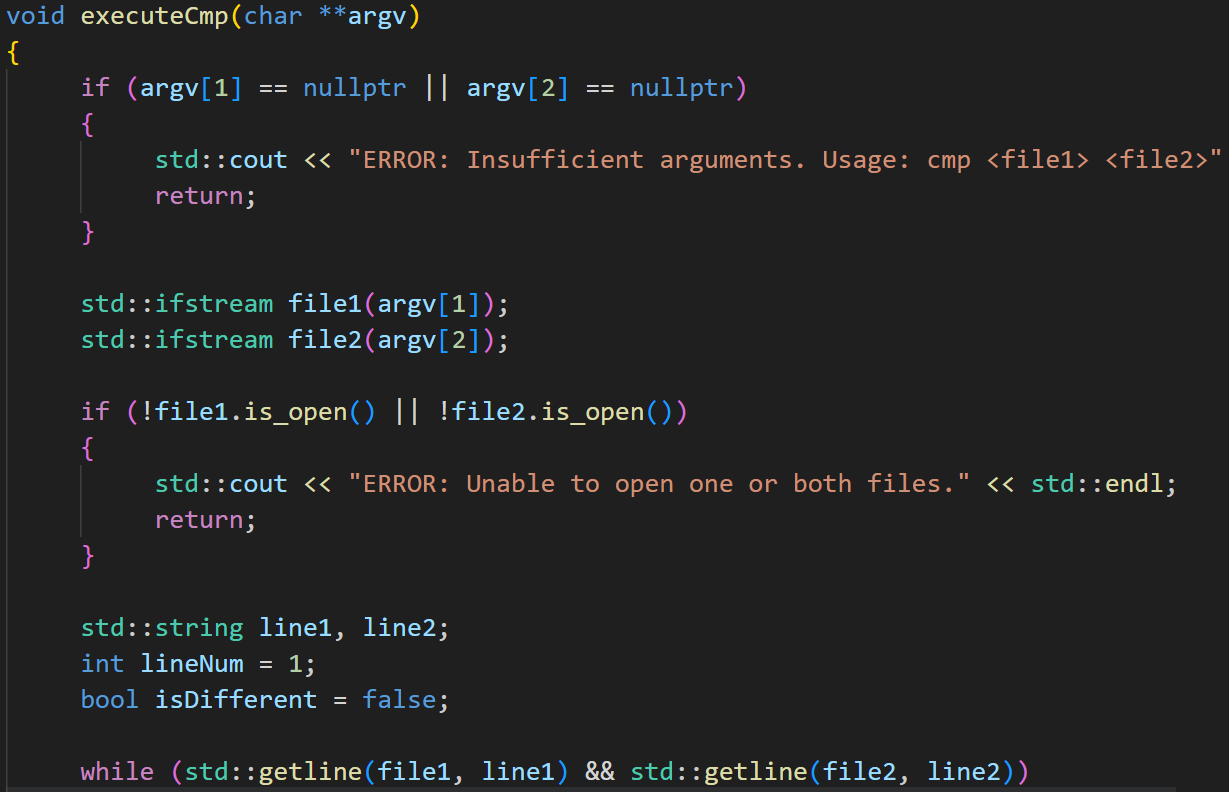
7.5 history

The 'history' command displays the command history of the shell. It provides a list of previously executed commands for reference.



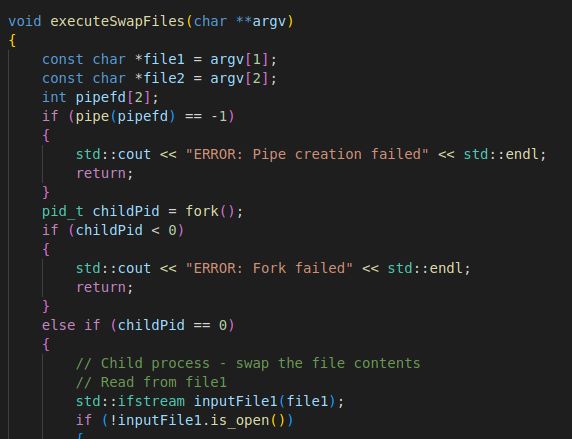
7.6 cmp

The 'cmp' command compares two files byte by byte and displays the first byte at which they differ. It helps to identify differences between two files.



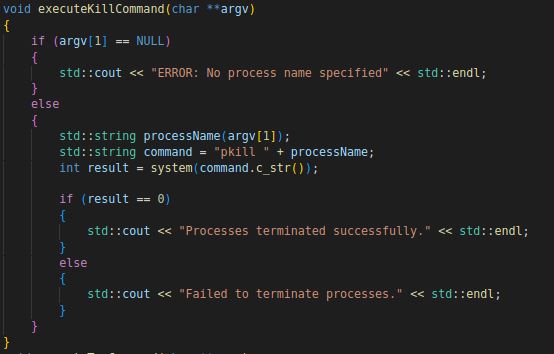
7.7 swap

The 'swap' command swaps the contents of two files with each other.



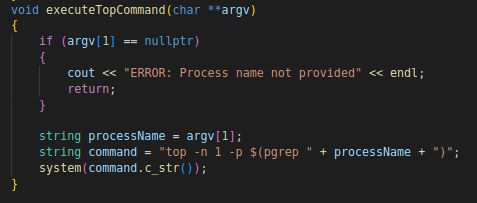
7.8 kill

The 'kill' command terminates a running process by sending a specified signal. It allows users to end processes gracefully or forcefully.



7.9 Top

The 'top' command displays real-time information about running processes and system resource usage. It provides insights into process activity and resource consumption.



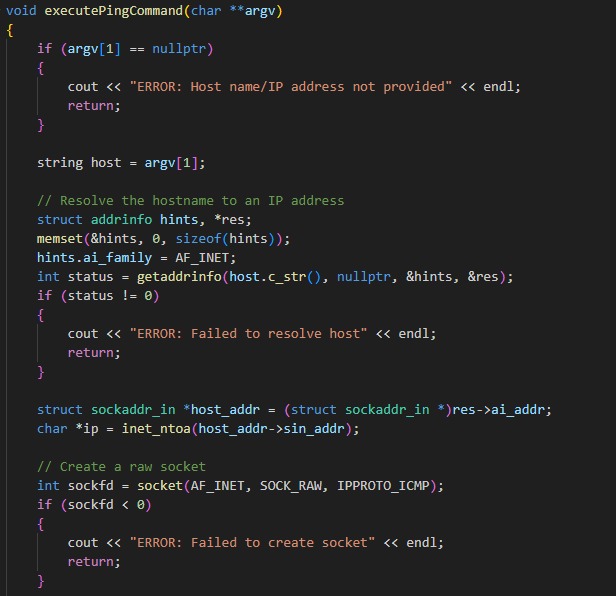
7.10 open

The 'open' command opens a given URL on a web browser. It launches the default browser with the specified URL as the target.



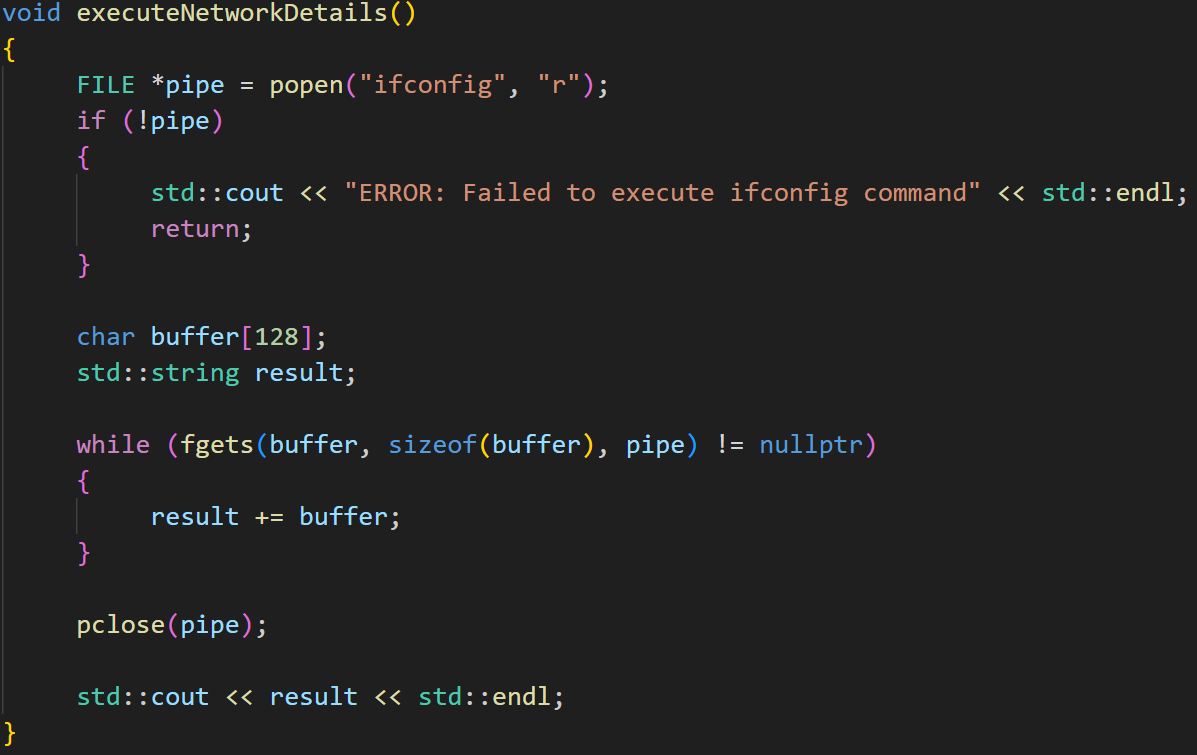
7.11 ping

The 'ping' command allows users to test network connectivity by sending ICMP echo requests to a specified destination. It measures the round-trip time for packets and provides information on packet loss. The command syntax typically includes the destination IP address or hostname.



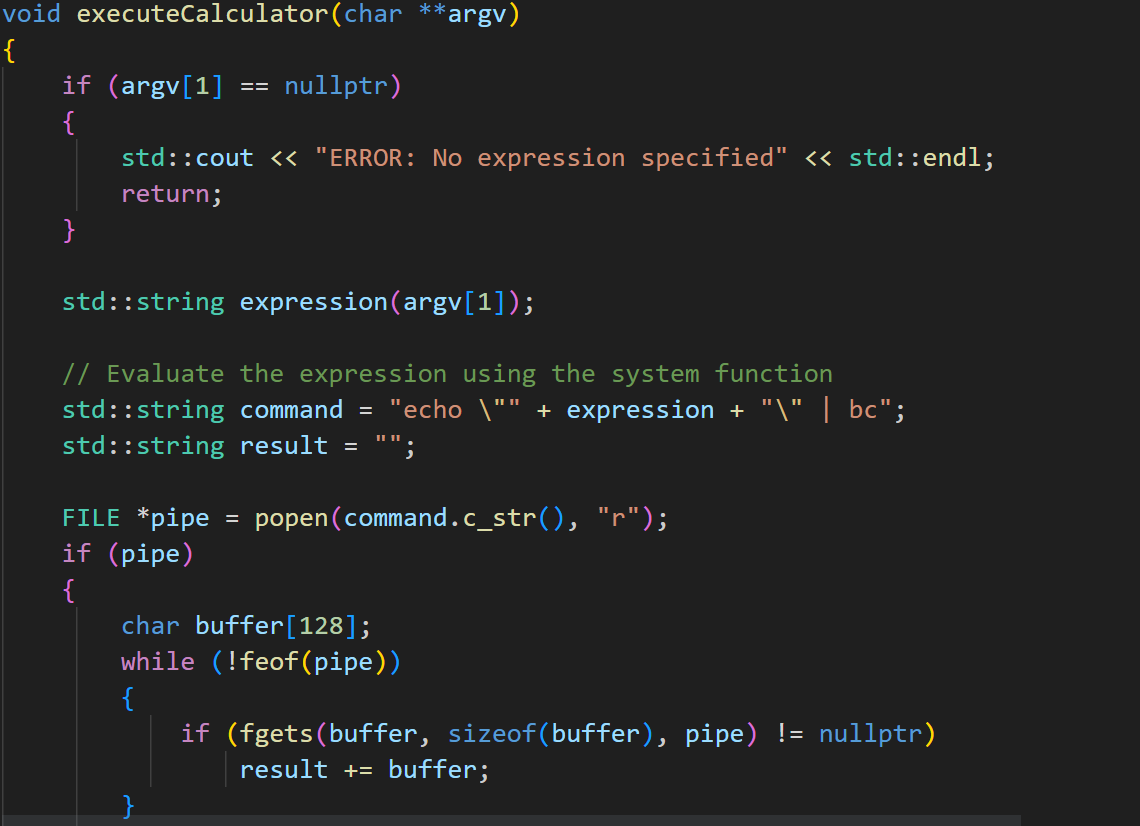
7.12 Network

The 'Network' command displays the network interface configuration information for the system. It provides details about the network interfaces, including IP addresses, MAC addresses, subnet masks, and other network-related parameters. This command helps in diagnosing and troubleshooting network-related issues.



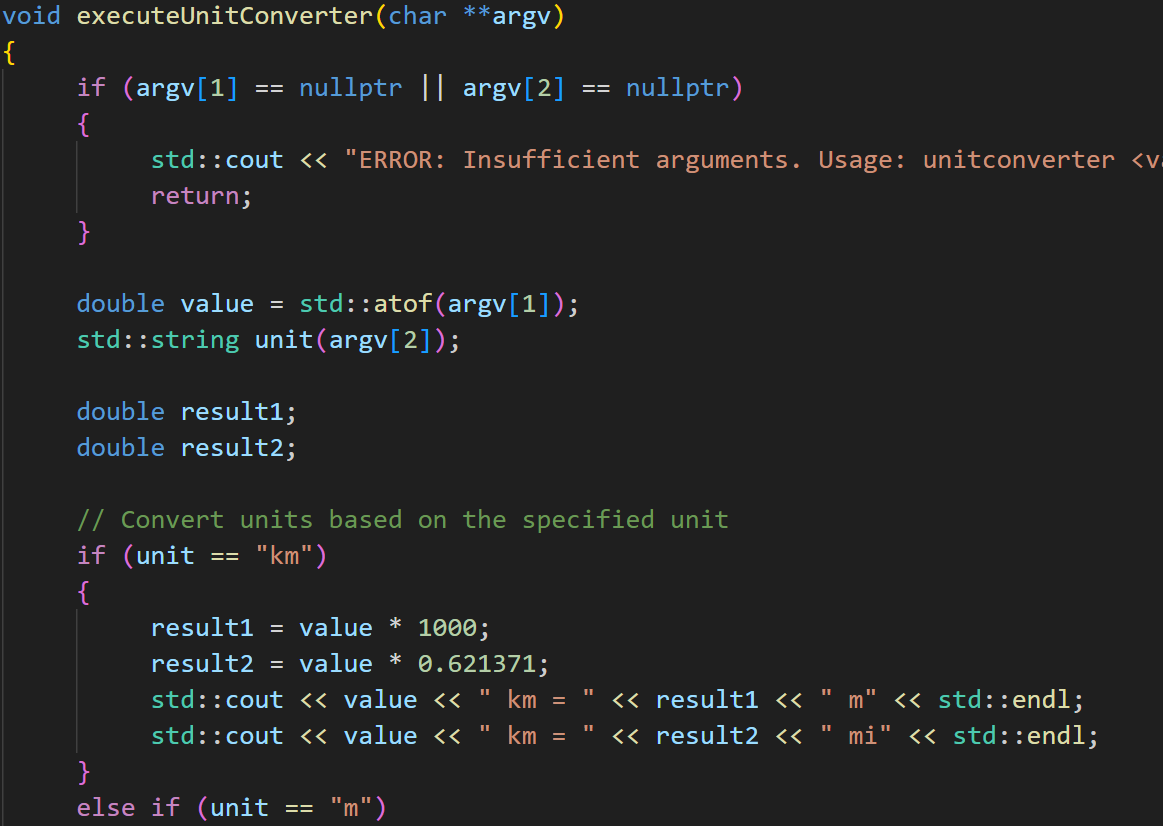
7.13 Calculator

The 'calculator' command provides a simple calculator functionality within the shell. It allows users to perform basic arithmetic operations such as addition, subtraction, multiplication, and division. Users can input mathematical expressions, and the calculator will evaluate and display the result.



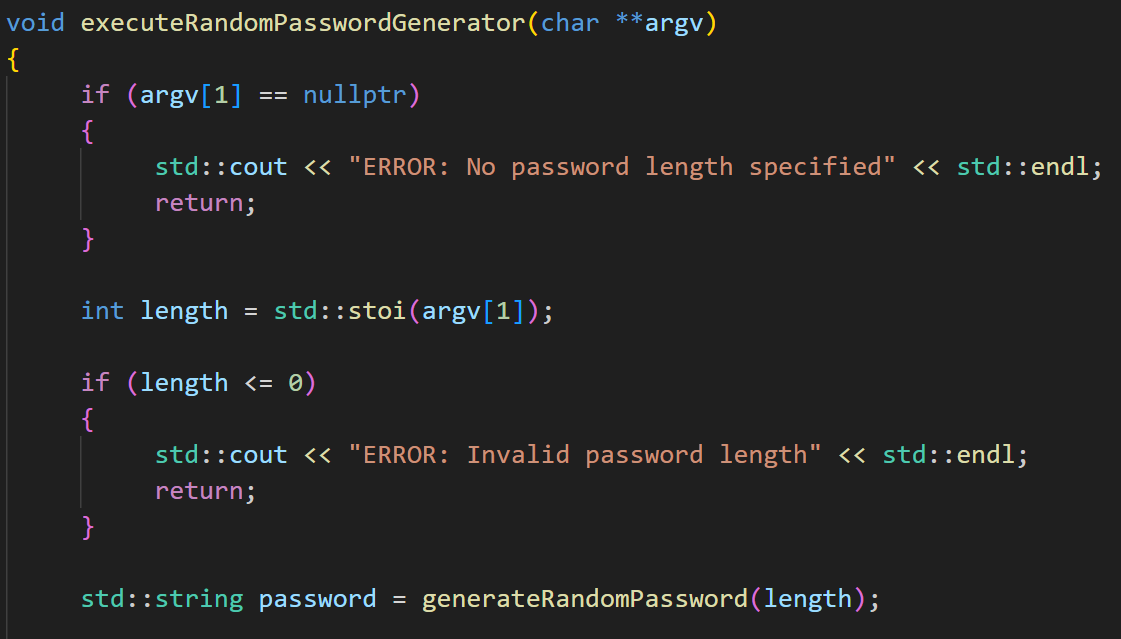
7.14 Unit Converter

The 'unit converter' command allows users to convert between different units of measurement. It supports conversions for various categories such as length, weight. Users can specify the amount and converted units would be displayed onto the shell.



7.15 Random Password Generator

The 'random password generator' command generates a secure and random password. It allows users to specify the length of the password. The command generates a password meeting the specified criteria and displays it for use.



8. Conclusion

The Mini-Shell project successfully extended the functionality of the shell by implementing additional commands related to networking, pipelining, and file handling. The new commands, including 'ping', 'ifconfig', and others, enhanced the versatility and usability of the shell. The project achieved its objectives and provided users with a more feature-rich and powerful shell environment for their tasks. Through the implementation of various approaches and commands, the project showcased the potential of expanding a basic shell into a more comprehensive tool.

9. References

Google drive link of video:

https://drive.google.com/drive/folders/1ia7XqNaJf7kuR3w8dMoO63Zqd9r5CokD?usp=sharing