

Alias and unalias commands (command_alias.cpp)

Command_alias.cpp implements two internal commands: alias and unalias, which have similar functionalities as the Unix-like commands to add and remove custom shortcuts for long commands. It consists of two functions: 'internal_alias' and 'internal_unalias' to handle adding and removing shortcuts, respectively. Both functions utilize string manipulation techniques to extract relevant information from the user input and interact with the aliasMap data structure

Alias has the following functionalities in our program:

-h: Display help message for the alias command.

-p: Display all existing aliases.

-a: Remove all aliases.

alias_name='command': Define an alias where alias_name is the name of the alias, and command is the associated command

Unalias has the following functionalities:

-h: Display a simple help message for the unalias command.

alias_name: The name of the alias to remove.

Help command (command_help.cpp)

The Help command in command_help.cpp provides users with assistance and information about various internal commands supported by the program. It displays a general help message when invoked without any arguments. This message provides an overview of the available internal commands and how to access their respective help messages. In addition, users can request help for specific commands by providing the command name as an argument to the Help command. For each supported command (e.g., cd, alias, unalias, history, set), the Help command provides detailed information on its usage, options, and functionalities.

History command (command_history.cpp)

The History command in command_history.cpp allows users to manage and view the command history of the program. It provides functionalities to display, clear, delete specific entries, and retrieve past commands from the command history. Additionally, users can reference the most recent command or a command by its index using special notations. It has the following functionalities:

-h: Display a simple help message for the history command.

-c: Clear the command history by deleting all entries.

-n NUM: Display the last NUM commands in history.

-d offset Delete the history entry at position OFFSET. Negative offsets count back from the end of the history list.

!n Retrieve the nth command from the history list.

!! Retrieve the most recent command from the history list.

Set command (command_set.cpp)

The Set command in `command_set.cpp` offers users a versatile tool for configuring shell options and environment variables within the program. It leverages the functionality of the `execvp` system call to execute shell commands, enabling users to modify various settings and behaviors dynamically. It serves just like Unix's `set` command, and the following are a few of its functionalities:

Help Message Display (-h):

Users can access a detailed help message for the Set command by specifying the `-h` option.

Configuration of Shell Options:

The Set command enables users to configure shell options by providing specific command-line arguments. For example, users can set options such as `-o`, `-e`, `-u`, among others, to modify shell behavior and control error handling, environment variable expansion, and other shell functionalities.