Custom Alpine Shell

# Abstract

The Custom Alpine Shell is a shell written in C, which allows for easier access of the filesystem.

# Introduction

For the project we have chosen to write a Custom Alpine Shell with emphasis on access of the filesystem in C.

-What is Alpine

-

# Implementation

## Command Parser

As Alpine does not use a Graphical User Interface, all user commands must be accessible by the command line. This means that the Command Line Input must be correctly interpreted and parsed.

-split-make commands

After a line is entered, the input is first split by the whitespaces, and the first argument is converted to the corresponding command, with not defined inputs being converted to the “error” command, which then displays an error to the user. The rest of the input is saved as nodes inside the Linked List.

-execute commands

After the input is converted into the Linked List, it is executed by finding the matching command case. These command cases first evaluate, whether the length of the Linked List is correct, win that case the execution of the command starts, otherwise an error is shown to the user and the execution stops.

-free memory

## Directory Management

For the directory management it was important to keep the ease of use in mind. Thus, the files and directories inside of the current working directory are saved inside Linked Lists. When using commands, instead of having to specify the filename or directory name, it is accessible by using the unique integer identifier of the corresponding Linked List.

These identifiers are displayed on the screen alongside their corresponding file/directory. Whenever a new directory is accessed, a new corresponding Linked List is generated, and the old one deleted.

## Command Implementation

### Move

When the move command is called without further arguments, the files inside the current directory are displayed to the user, alongside the identifiers of these files. The following user input is then evaluated, and the complete file path is saved internally. Then all the subdirectories are displayed with their identifiers, with which the user can navigate through the directories. Once the final directory is chosen, the new path is saved internally, and the file is moved using these saved parameters.

### Copy

Conceptually it works the same as move, except that instead of being able to directly move a file using a system call, first a new file is created at the destination, into which the content of the first file is written into.

### Rename

### Go

By calling go, all the directories inside the current directory are displayed. When a number is typed, the current working directory is changed to the corresponding directory.

### Run

### Delete

### Help

## (Log)

# Results

# Discussion

## Lessons Learned

## Problems

-Windows-Linux-Alpine-Segmentation errors

-C in general -memory leaks

## Division of Labour

Directories

Parser

Commands

Debugging

Research

Design

Report

# Conclusion

A fully functional Shell for Alpine was created, which allows for easier manipulation of the file system, including renaming, moving copying and deleting of files. Any functionality of the normal shell which are not implemented in the custom shell can be accessed with an integrated system call.