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 can be accessed by using the unique integer identifier of the corresponding Linked List.

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

## Command Implementation

### Move

### Copy

### Rename

### Go

### Run

### Delete

## (Log)

# Results

# Discussion

## General/What we’ve learned

## Problems

-Windows-Linux-Alpine

-C in general -memory leaks

## Division of Labour

Directories

Parser

Commands

Debugging

Research

Design

# Conclusion