|  |  |  |  |
| --- | --- | --- | --- |
|  | 3/7/2015 |  | |
|  | |  | |
| Project 1 Report  *Quite a Shell (quash) implementation* | | | |
|  |  | |  |
|  |  | | Tyler Steiner, Michael Esry |

Project 1 Report

Quite a Shell (quash) implementation

# Introduction

## Problem at hand

The problem at hand that we were given was to implement a shell program similar to bash. A shell program performs basic interaction with a user, including directory navigation and program launching (both in the foreground and as a background process), as well as other basic interactions. The implementation must be done in the language C.

## Proposed Solution

The proposed solution will be a single program that will act as a shell program implementing the requirements as given.

# Implementation

## Discussion

The project was successfully implemented using a single .c file. As per the requirement, quash successfully implements its own set, cd, jobs, quit, and exit functionality. Each of the built-in functions is further described in the README file for quash and all of the functions are listed out if a user types “help”.

Our group very heavily utilized github to split up the work and share the resulting code. While lacking a formal separation of the tasks needing to be implemented, our group used heavy communication, in association with github, to split up, work on, and finish the project.

Each part of the program produced its own set of challenges. For example, an overarching problem of getting and dealing with the string specifying the input command was something that had to be dealt with early on. This was further complicated by a very limited knowledge and experience level with the C language.

## Current Limitations

# Testing

There were two main phases for testing this project. The first testing stage was a normal type of testing that is done while coding. The main difference being that it was done by both members of the group. When one member of the group would successfully implement a feature, they then reported that to the other person who, after pulling the code down from github, would test out that feature to see as well.

The second phase of testing was much more involved. It was done at the end of the coding process. At that point, all of the features and requirements of quash were tested one by one, using all possible permutations of a command that could be required.

# Concluding Remarks

# Appendix

## README

## Code