Design Document: Simple Shell

CSE 134: Embedded Operating Systems
Assignment 1

Mitchell Elliott
CruzID: miclelli

April 12, 2021

1. Goals

The goal of this programming assignment is to implement a simple shell that parses commands and executes them using systems calls. The shell will run on FreeBSD on a Raspberry Pi.

2. Design

- Print a prompt (shell>) to the user
- Read a line of input from the user
- Process the input using getargs()
- Determine which function(s) needs to be called and if I/O needs to be redirected, pipes need to be used or if multiple programs need to be run separately
- Fork the program using fork() and execute the commands using the execvp() system call
- Repeat the above steps until the user types exit

3. Details

- Single programs
    - Use fork() and execve()
- Input and output redirection
    - Input will be redirected when the user types (<) and output will be redirected when the user types (>)
    - Use open(), dup2() and close()
    - If the output redirection symbol is >&, both standard output and standard error are redirected

  - A single > (or >&) replaces the named file with the output
  - If there is a double >> (or >>&), the output is appended to the file
- Multiple programs not connected
  - If you have multiple programs on a single line with semicolons (;) between them, they will run in series, one after the other
  - A program must exit before the next one on the line can start
- Multiple programs connected by pipes
  - Use pipe()
  - If the pipe symbol is |&, both standard output and standard error are redirected
- Internal shell commands
  - cd: Change the current working directory, using chdir()
  - exit: Exit the shell using exit()