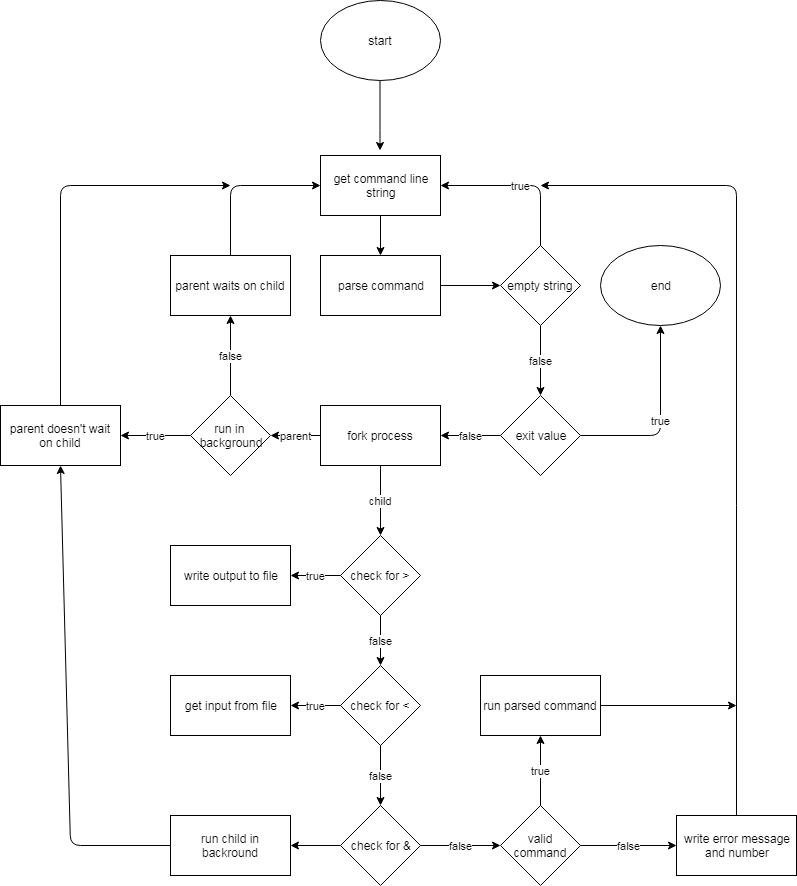
Jeremiah Hsieh

ICSI 500

Project 1 Shell Implementation

**System Documentation**

a.i Data flow diagram

a.ii

There are 3 functions for this program.

a. The main function contains the primary loop to get input from user

b. The parseInput function takes the string from main and splits it into individual words. It also checks for file input/output and background running conditions as well as exit conditions.

C. The execute function forks the process and runs the parsed commands in the child while the parent waits. Based on conditions from parseInput it may delay child runtime or get input/output from/to file.

a.iii

No particular data structures or algorithms used. Unfortunately I did not manage to get the multi fork pipes working properly so I did not include it in my diagram.

b.i

I tested the program by running various common commands used such as cp, vim, rm, ls, sort, mv, mkdir, and wc. I also tested the various file input/output and background running processes although for the background process test the most obvious one would be one with outputting of text to console since if it ran in background then the text would show up after the next command console line. However, without a full listing of what commands need to be supported I am not sure what I could be missing off the top of my head.

b.ii

Example tests in project1test.png included with paper

c.i

Source code included with paper

c.ii

Compile shell.cpp and run ./a.out to start shell simulation

c.iii

No parameters needed