Background:

Project # 1: Build a minimal shell (minShell).

We have seen that a Shell (Like BASH) is a sophisticated command-interpreter and scripting language that provides a powerful user interface to operating system. In the most basic sense, a Shell is a program that can execute other programs.

The basic processing steps for a minimalist shell might be summarized by the following pseudocode logic:

1. Display prompt to the user (e.g. minShell$)
2. Wait for a command from the user
3. parse the command
4. execute the command
5. return to step 1

To begin building your minShell you might start by building a program in C or C++ that run (executes) other programs. To do that, consider that steps (1-5) outlined above, and then following description:

1. Write a while loop that will display a prompt, and then waits/reads input from the user.
   1. Reading input from the user means the user types a command line and presses enter/return
2. Parse the command line
   1. Parsing means to process the command and extract the *command* and the *arguments*. For example, consider the following command line: $ ls -l

The full command line entered by the user is ls -l

The parsed command line extracts the command and the arguments such as in the example above, the following: command=”ls” and args=”-l”

1. Executing the parsed command requires the creation of a new (child) process. Creating a new process is performed with the *fork* system call, followed by an *exec* system call to load the command process image.
   1. When a shell executes the command, it first creates a copy of itself (the child process by calling *fork*). It then calls *exec i*n the child process to load/run the command (***See Figure 1 below for an illustration of the process).***
      1. Please read the following Lecture from USNA to begin getting a grasp on the concept: <https://www.usna.edu/Users/cs/aviv/classes/ic221/s16/lec/14/lec.html>

See: lecture 9 webpage -> lecture9\_examples.tar.gz -> c\_examples/shell/minShell.c

See: lecture 9 webpage -> lecture9\_examples.tar.gz -> c\_examples/parseString/parseString.c for an example of how to parse (split) strings in C

The first version of your minimal shell should support the following features:

1. Display prompt to the user (e.g. minShell$)
2. Wait for a command from the user
3. parse the command and the arguments: assume all commands use the following form:
   1. [command] [arguments]
      1. examples
         1. /bin/ls -l

Command = “/bin/ls” and one argument = “-l”

* + - 1. /bin/echo “Hello minShell”

command=”/bin/echo” argument=”Hello minShell”

1. execute the command
2. return to step 1

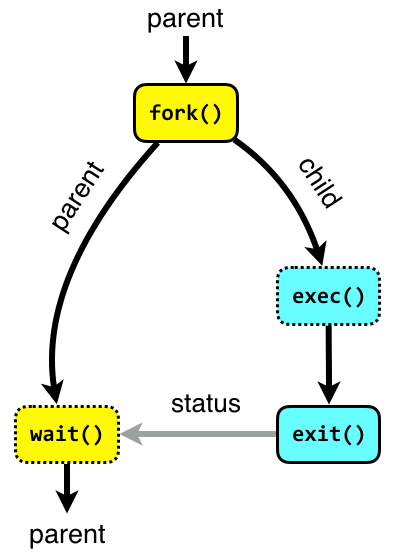


Figure 1: Command Execution

Image source: <http://www.it.uu.se/education/course/homepage/os/vt18/module-2/process-management/>