Topics: A Programmable Shell - Creating A Programming Environment

Approach: What's it do? How's it work? How'd I code it?

Today's System Calls:
review of fork(), execvp(), wait()

Outline

Features of a Unix Shell

a. Runs programs
b. Allows users to program the execution of programs
  1. simple batch files - a list of commands
  2. real programs - variables, user input, control flow
  3. global variables - passing values to other programs

Let's add these features to our shell

Review of How the Shell Runs Programs

a. forkquiz.c - shows how fork, exec, exit, wait work
b. Details of how wait() works
c. Add status report to our shell (smsh2.c)

Control Flow in the Shell

a. The if statement - why
b. The if statement - what it does
c. The if statement - how it works
d. adding if to our shell
e. Intro to other control structures

Variables in the Shell

a. shell variables - why
b. assign, reference, input, list all variables
c. variables - how it works
d. adding variables to our shell - the varlib (smsh3.c)

Global Variables in the Shell: the environment

a. environment variables - why ?
b. environment - what it is/does
c. the getenv() function
d. the export shell command
e. environment - how it works
f. some demos - showenv.c, envchange.c
g. adding environment processing to our shell
  - loading our variable list from environment
  - sending our variables back to the environment
  - adding new variables to the environment