308 Project 1 Summary

Basically we write our own shell and it works just like shell but a simple version of it. So we grab user's input through command line to determine which kind of function or build-in command we need to execute or run, and also we grab the information given and store them into char array and then pass them as variable to some function when needed so achieve the required output. Here is a list of function that we can call and feed them user's input:

• fork – create a child process

• execvp – replace the current process with that of the specified program

• waitpid – wait for a child to exit (or get exit status)

• exit – force the current process to exit, with the given return value

• chdir – change working directory

• getcwd – get current working directory

• getenv/setenv – retrieve and set environment variables.

• perror – display error messages based on the value of errno

• strcmp, strcpy, strcat – string manipulation.

Here is something I need to pay attention (basically C language)

The length of command is always your input + 1 to include null char in the end.

If '\n' in string then it takes 1 char (use strlen to represent), if \0 in string then it does not take 1 char(use strlen to represent). So it is essential to modify \n with \0

I used strncmp instead of strcmp since strcmp does not work when I try to compare user command like “exit”, “pid”. I think the reason is that I declare my variable “command” as char array and need extras attention when compare to string directly.

“The only case where strncmp() would be safer than strcmp() is when you're comparing two character arrays as strings, you're certain that both arrays are at least n bytes long (the 3rd argument passed to strncmp()), and you're not certain that both arrays contain strings (i.e., contain a '\0' null character terminator).”

use Fgets() and get rid of gets()