COP 4600 - Homework 1

Due September 15, 2014

Write a C or C++ program called **mysh** replaces the command shell in Unix.

-After started, it prints a prompt “#” and reads a command line terminated by newline.

-Interpret the following command lines:

# quit

terminates the mysh shell

# run program [parameters]

Interprets “program” as the full path to the program to execute, with the optional “parameters”. It uses fork() + exec() to start the program with the corresponding parameters, and waits until the program terminates (use the waitpid() call).

For instance

 run /usr/bin/xterm –fs 30

would bring up a terminal with font size 30, but the prompt would not return until the terminal is closed.

Display an error message if the specified program cannot be found or cannot be executed.

# background program [parameters]

It is similar to the run command, but it immediately prints the PID of the program it started, and returns the prompt.

# murder PID

Immediately terminate the program with the specific PID (presumably started from this command line interpreter). Use the kill() function call to send a SIGKILL signal to the program. Display success or failure.

To help you, you might want to read some of the readings associated with this homework at the class webpage. I have also linked from the webpage a code segment which might help you in reading the commands.

As a note: the easiest way to program this assignment is if you are using a native Unix system, such as Linux. Alternatively, you can use the Cygwin environment in Windows (as far as I know, there is nothing in this project which would prevent you fully implementing it in Cygwin, but your mileage might vary).

## Extra credit (10 points)

Implement a repeat command as follows:

# repeat n command …

Interprets n as the number of times the command must be run, command as the full path to the program to execute, and the others as parameters. The command should start the specified number of program instances, print the PIDs of the created processes and then return to the prompt, without waiting for the processes to terminate. For instance:

 repeat 4 /usr/bin/xterm

would bring up 4 terminals and print out something like:

PIDs: 31012, 31013, 31014, 31015.

## More extra credit (10 points)

Implement the following command:

# murderall

Which immediately terminates all the programs previously started by the mysh shell which had not been previously terminated by it, or by murder. It should output something like this:

Murdering 3 processes: 16000 31012 31013

## What to submit:

* The code as a single .c or .cpp file.
* If you implemented the extra credit part: a text file describing the syntax of the implementation, and example of use.