

Homework 3 - Simple Shell

Due by 7:00 p.m. Wednesday, 3/11/15

For this assignment you will implement your own shell that runs on top of the regular command-line interpreters for Linux and Windows. Your shell should read lines of user input into a 1024 byte buffer, then parse and execute the commands by forking/creating new processes. Write both a POSIX version of your shell that calls `fork()` followed by `execvp()`, and a Win32 version that calls `CreateProcess()`. Following each command, your shell should wait for its child process to complete, and then reprint the command prompt. The user should be able to specify the command to execute by giving a path to the executable file (e.g. `./hw1`) or by using path expansion to locate the executable file (i.e. searching each directory in the `PATH` environment variable). (Note that the `execvp()` and `CreateProcess()` functions perform this processing automatically; you do not need to program it yourself.)

Before your POSIX version calls `execvp()`, it should parse the input string and separate it into a collection of substrings representing the executable file and any command-line arguments. It should then call `execvp()` with these substrings as arguments. (Note that the number of command-line arguments is variable; to handle this your code may select among various `execvp()` call statements with different numbers of arguments.) Your code must handle up to three command-line arguments (not including the executable file itself). In the Win32 version, you should be able to pass the unparsed input string directly to `CreateProcess()`. For both versions, if the user enters the `exit` command, your shell should terminate (returning to the regular shell).

Here is a sample execution on a Linux machine:

```
Myshell> ls
file1      file2      file3
Myshell> ls -l
total 3
-rwxr-xr-x  1 cassidy      None      2883 Nov  5 12:57 file1
-rwxr-xr-x  1 cassidy      None      1468 Oct 23 14:07 file2
-rwxr-xr-x  1 cassidy      None       200 Jan 24 2013 file3
Myshell> /usr/bin/echo this
this
Myshell> exit
```

You should submit your source code files (one for each platform) and a short writeup in pdf format that includes a description of what you did and the compilation and execution output from each of your programs. Your execution output should include commands with command-line arguments. Then use the `exit` command to exit your program and show the output of the same commands in the regular command-line interpreter for that machine to ensure they match. Submit everything to the regular submission link on iLearn, and then submit just the writeup to the TurnItIn link to generate an originality report.