Creating a Shell Interface in Java

Nicholas Alexander

Division of Computer Science

Marshall University, Huntington, WV 25703

Linux Shell interface implemented in the Java programming language

I. Problem Statement

The problem was stated to create a Linux shell interface using the Java programming language. The purpose of the problem was to understand how Java can spawn processes and execute internal commands. In Part 1, the shell was to execute basic commands, with or without parameters, and print the output of the shell command. In part 2, the change directory command was implemented.

II. Methodology

A code skeleton was provided to lay the initial foundation of the program. Alongside the code, an algorithm was provided to complete Part 1. The most important aspect of Part 1 was using the Java API to learn and understand the ProcessBuilder class. Beyond the ProcessBuilder, Part 1 consisted only of simple I/O operations. The major challenge encountered during Part 1 was capturing the output of the command and printing it to the console.

The fatal flaw of ProcessBuilder in terms of a Linux shell was the inability to change directories inherently; however, the API provided a method to change the directory for all subprocesses. The key point was to understand that the directory only changed for commands executed after the ProcessBuilder's start method was called. The major challenge in Part 2 was parsing the directory input for both relative and absolute addressing schemes.

III. Results

The program worked great and passed all testing conducted with the exception of a single, now resolved, bug. The change directory method cannot handle directories that are encapsulated in quotes and, by extension, directories that include a space (e.g. "My Desktop"). The bug does not crash the program; the directory simply will not change. The bug was resolved by use of a string Regex expression conserving spaces within double quotes.

As for distribution of labor, I implemented the given skeleton code as well as the algorithm from Part 1. After all basic commands would execute and the shell would print the output, my partner devised and implemented the algorithm for changing directories. Afterwards, the code was handed back to me for bug testing. I found that the change directory would not allow for going back (e.g. “cd ..”) nor absolute addressing (e.g. “/home/user/Downloads”). With clearance from my partner, I refactored some of the code for readability and implemented the absolute addressing and the fixed the relative addressing bug.

IV. References

1. *Java ProcessBuilder API* (http://docs.oracle.com/javase/7/docs/api/java/lang/ProcessBuilder.html)

2. *Java Process API* (http://docs.oracle.com/javase/7/docs/api/java/lang/Process.html)