Bash

Standard streams introduction

The terminal (and shell) is interactive

When we are working in a terminal window, we typically issue commands and read the result in the terminal itself

It is safe to say that the terminal window is operated interactively - we issue command and the terminal shows the result

If we don’t issue any commands, nothing happens and the terminal patiently waits for us to make the first draw

Inside the terminal, a shell is working (in our case - Bash)

Bash is responsible for waiting for commands and operate on them

Where does bash get our commands from?

Someone obviously wrote bash* but how would you write an interactive program so that it can read commands?

A program that reads input from a text-based interface, typically reads this input line-by-line from an inputstream called “standard in” (often shortened to stdin).

If you have programmed in Java, you have probably used the System.in object (either explicitly or wrapped in the object obtained from System.console())

* Brian Fox wrote bash back in 1989

Where does bash get our commands from?

If you have programmed in C, you might have used the stdin file handle provided by the stdio.h header file.

Regardless of your previous programming experience, what is this variable we’re reading from? It is a stream of data provided by the operating system.

The stream is called standard in and normally it gets its contents from the user (who typically types in the data in the form of text from the keyboard).

So bash gets its commands from the user who types them in interactively from the keyboard.

How does bash communicate with the user?

Since bash is typically run interactively, and gets its input from the user who types in commands via the keyboard, it needs a way to communicate back to the user. This is done via a stream called “standard out” and it is typically connected to the terminal window running bash.

In Java you might have used this stream when outputting text to the terminal using System.out.println() (where the out object is actually connected to the standard out stream). In C, you have probably used printf (which by default prints to the same stream, in C referred to as “stdout”).

How does bash read the commands?

Interactive text based programs (such as bash) typically waits for you to type something and then hit the enter key.

Such a program reads from a stream called stdin (standard in) and it reads and processes one line at the time (at least it detects the newline character originating from the user pressing enter).

We’ll look at some other interactive programs next.

An interactive calculator

In your terminal running bash, start the interactive calculator bc:

```
$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
```

Giving bc something to work with

```
$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
99+1
100
```

This was fun, how do I get out of bc?

Press Ctrl-d (the control key and ‘d’ at the same time). This will send the EOF character to the program (end-of-file).

```
$ bc
bc 1.06.95
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006 Free
Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
99+1
100
<CTRL-D> (you won’t see anything but here you press ctrl-d)
$ # and back to the bash shell we are
```

One more standard stream

Standard out (stdout) is used for normal output from a program. Standard in (stdin) is used for normal input to a program. But there is one more stream.

The standard error stream (stderr) is used to output error messages (and other diagnostic messages, like for instance a progress bar or so).

If you've used Java, you might have written to stderr with a statement such as:

```
System.err.println("The file " +
                    fileName +
                   " could not be opened.");
```

Recap

Standard out (stdout): The stream used for output from a program. Usually this is connected to the terminal where the program is running.

Standard in (stdin): The stream used for input to a program. Usually this is connected to the text the user types in via the keyboard to the terminal where the program is running.

Standard error (stderr): The stream used for output of error messages and other text not part of the normal execution output from a program. Usually this is connected to the terminal where the program is running.