The sys admin's daily grind: SPL

All the World's a Stage

This is the ultimate programming language for decelerated feature page readers and the counterpart to agile software development: the Shakespeare Programming Language.

By Charly Kühnast

You that choose not by the view. Chance as faire, and choose as true.

Even though "To be or not to be" translates to a really neat regular expression – `/(bb|[^b]{2})/` – the link between Shakespeare and sys admins isn't immediately apparent. But as the playwright once wrote: "A good wit will make use of any thing." So, let's set aside Perl and Bash for a short while and look at a programming language that is full of drama and poetry.

The structure of the Shakespeare Programming Language [1] follows that of a play. Following a freely selectable title, which the SPL parser only perceives as a comment, the developer declares the variables, which appear in the form of a dramatis personae of the major actors:

```
Caesar, a successful and soon dead Roman
        emperor
Othello, a Venetian general
```

The fact that these two characters don't occur in the same Shakespeare play might worry literature buffs, but it doesn't faze the parser one bit. In fact, the parser just remembers the names and uses them as integer variables. The parser doesn't evaluate the text after the comma, so you can make the description as flowery as you like.

The approach of dividing the program code into acts and scenes is also typical of the theater. These dividers not only visibly organize the program but are also the markers for the theatrical equivalent of the go-to command:

```
Act I: A battle of words
Scene I: Caesar insults Othello
```

Before you can assign a value to a variable, the character first needs to enter the stage:

```
[Enter Caesar and Othello]
Caesar: You stupid rotten beggar!
```

The value assigned to the variable is defined by how friendly the lines spoken by the actor are. Nouns represent a value of 1 if the statement is friendly or neutral, or -1 if the words have negative connotation.

In the previous example (`beggar`), the value is -1. Each preceding adjective multiplies the value by 2. In other words, `Stupid rotten beggar` means:

```
-1*2*2=-4.
```

Because Caesar is speaking to Othello (`you`), a value of -4 is assigned to the variable `Othello`.

Ay, Much is the Force of Heaven-Bred Poesy

Of course, the output is just as stylish: The `Open your heart!` command triggers the output of the numeric value. `Speak your mind!` outputs the letters whose ASCII value corresponds to this number. Go-to commands are just as easy to handle:

```
Othello: Let us then proceed to
         Act II!
```

The character can add a condition to the go-to by asking a question, which triggers a comparison of the variables and jumps to a scene or an act depending on the results:

```
Caesar: Am I better than you?
Othello: If not, let us return to
         Scene I.
```

There is no such thing as an SPL compiler, but there is `spl2c`, an SPL to C translator, which was written in Flex and Bison. Because of their poetic nature, programs in the Shakespeare Programming Language can be amazingly wordy. Even a classic "Hello World" takes a couple of hundred lines. Or to put this in the playwright's own words: "Though this be madness, yet there is method in't" (Polonius in Act 2 of *Hamlet*). ■■■

INFO

[1] Shakespeare Programming Language: *http://shakespearelang.sourceforge.net*

CHARLY KÜHNAST

Charly Kühnast is a Unix operating system administrator at the Data Center in Moers, Germany. His tasks include firewall and DMZ security and availability. He divides his leisure time into hot, wet, and eastern sectors, where he enjoys cooking, freshwater aquariums, and learning Japanese, respectively.