Seminar Configurable Systems

Prof. Zeller / Christian Lindig Lua

Questions

Here are questions about *Lua—an extensible extension language* by Roberto Ierusalimschy et al.; it appeared in *Software: Practice and Experience* 26(6), p. 635–652, 1996.

1. In Perl, the predominant data type is the regular expression, in Tcl it is the string, and in Scheme the list. What is the most important data type in Lua?
2. What types does the Lua language know? How are user-defined types handled?
3. Ierusalimschy et al. stress the importance of table literals in Lua syntax as a markup language. This was before XML became as popular as it is today. Compare XML and Lua's tables as markup devices.
4. Lua has fallbacks, which are similar to exceptions in Java or ML. But what is different? Compare the stack of procedure activations when the topmost procedure raises an exception or invokes a fallback, respectively. What happens in case of ML/Java-style exceptions and what in Lua? Where on the stack is an exception caught in ML/Java?
5. The paper does not talk about the C API. Skim the description of the C API from the Lua 2.5 reference manual to get an overview.
   (a) What abstractions exist to extend a Lua interpreter? Compare them with Tcl and Libscheme.
   (b) How does a C function implementing a new primitive and the Lua interpreter exchange arguments and results?
   (c) How is memory managed?
6. Libscheme's author Benson claims that Scheme is suitable for object-oriented programming, the same do Ierusalimschy et al. for Lua. What do you think?
7. According to the paper, Lua is 10-20 times faster than Tcl, and according to the *Computer Language Shootout*[1] much faster than Scheme and Tcl, and slightly faster than Perl and Python. Speculate about the reasons.

[1] `http://www.bagley.org/~doug/shootout/craps.shtml`