

Le langage de programmation Lua sur une machine généraliste

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 11 septembre 2007

<https://www.bortzmeyer.org/lua-sur-machine-generaliste.html>

Après un petit essai, je ne crois pas que je continuerai à utiliser Lua. En effet, il dispose en standard de moins de possibilités que Python, qui est actuellement mon langage de choix pour les scripts.

Sur le papier, Lua a des forces intéressantes : interpréteur ultra-léger, langage simple et très bonnes performances. Conçu dans un pays du Sud (au Brésil), ce qui est rare pour un langage de programmation, Lua a une communauté réduite mais dynamique et est utilisé pour certains gros projets comme WoW dont il est le principal langage d'extension.

Motivé par la lecture de l'excellent article "*The evolution of Lua*" <<https://www.bortzmeyer.org/evolution-de-lua.html>>, j'ai essayé Lua pour un petit projet <<https://www.bortzmeyer.org/unicode-to-sql.html>>, qui a montré les limites de Lua.

Venant de Python, on est habitué à ce que tout marche dès la sortie de la boîte, ce que les Pythoniens appellent l'approche "*batteries included*". Rien de tel sur Lua. Ses mainteneurs sont très soucieux de garder une stricte portabilité, ce qui veut dire que Lua s'en tient à la norme ANSI du langage C. Celle-ci n'offre que très peu de possibilités d'interaction avec le monde extérieur. Sans Posix, une tâche aussi simple que de lister le contenu d'un répertoire nécessite l'installation d'un module extérieur <<http://www.keplerproject.org/luafilesystem>>!

En outre, l'absence de support pour Unicode dans Lua était inacceptable pour moi.

Il est clair que Lua, avec son approche minimaliste, vise bien plus le monde de l'« embarqué logiciel » (langage pour scripter des gros programmes) que celui des machines Unix généralistes, disposant de ressources matérielles et logicielles suffisantes pour que Python (ou ses autres concurrents comme Perl) puisse y tourner à l'aise.