

strlen et l'optimisation

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 25 juin 2009. Dernière mise à jour le 30 juin 2009

<https://www.bortzmeyer.org/strlen-optimise.html>

Comme le savent les lecteurs de ce blog, je suis en général très méfiant vis-à-vis de l'optimisation prématurée des programmes. Pas seulement parce que la plupart des programmeurs qui parlent de performance tout de suite ne mesurent pas <<https://www.bortzmeyer.org/mesurer-temps-execution.html>> le résultat de leurs bricolages mais aussi parce que cette « optimisation » est souvent un prétexte pour faire du code imbittable et inmaintenable. Mais, comme avec toute règle, il y a des exceptions.

Par exemple, si une fonction donnée est appelée très souvent, elle est évidemment une bonne candidate pour des optimisations très poussées. C'est le cas de `strlen`, fonction de la `libc` qui mesure la longueur d'une chaîne de caractères en C. Les programmes écrits en C (ce qui inclue les implémentations des langages comme Ruby ou Perl) appellent `strlen` tout le temps. On peut donc, une fois une mise en œuvre correcte de `strlen` produite, chercher des optimisations de bas niveau.

Il est par exemple amusant de comparer les deux versions produites par OpenBSD :

- La version portable <<http://www.openbsd.org/cgi-bin/cvsweb/src/lib/libc/string/strlen.c?rev=1.7;content-type=text%2Fplain>>, **triviale**,
- et la version optimisée <<http://www.openbsd.org/cgi-bin/cvsweb/src/lib/libc/arch/i386/string/strlen.S?rev=1.3;content-type=text%2Fplain>> pour les processeurs i386, en assembleur.

La GNU `libc` dispose également d'une telle version optimisée sur les i386 (version en assembleur pour i586 et plus <<http://sourceware.org/git/?p=glibc.git;a=blob;f=sysdeps/i386/i586/strlen.S>>). Voir aussi un bon article <<http://www.int80h.org/strlen/>> sur l'écriture en assembleur de `strlen`. Évidemment, `strcmp` serait encore plus intéressante à étudier, car plus complexe (les deux chaînes n'ont pas forcément la même longueur).

Une autre curiosité pour les amateurs d'optimisation est la discussion sur un `strlen()` rapide <<http://www.daemonology.net/blog/2008-06-05-faster-utf8-strlen.html>> pour UTF-8.

Merci à Victor Stinner pour ses intéressants ajouts.