

# Pour tester, il faut utiliser une commande d'une ligne, rappelable

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 avril 2009

<https://www.bortzmeyer.org/commande-rappelable-pour-tests.html>

---

À chaque fois que je donne des cours d'administration système et réseaux sur Unix (par exemple pour les FFTI <<http://www.afnic.fr/afnic/international/college/ffti>>), je suis surpris de voir que beaucoup d'étudiants, pour tester leur serveur de messagerie qu'ils viennent de configurer, utilisent un MUA interactif et graphique (comme Thunderbird) où beaucoup de clics sont nécessaires avant d'avoir pu envoyer le message. En fait, le test leur prend plus de temps que le changement de configuration qu'ils voulaient tester. Utiliser une commande d'une ligne, facilement rappelable avec la flèche vers le haut, ce que permettent tous les shells, leur faciliterait pourtant beaucoup la vie.

Autre exemple, pris sur une liste de diffusion, un conseil donné à quelqu'un qui devait tester son Postfix : utiliser, certes, la traditionnelle commande Unix `mail` mais en tapant un contenu du message (terminé par un point) à chaque fois. C'est presque aussi compliqué que d'utiliser un MUA graphique. Comme le courrier ne va pas marcher du premier coup, il faudra sans doute refaire le test plusieurs fois, ce qui est peu compatible avec ce mode interactif.

La bonne méthode est de fabriquer une commande d'une seule ligne, qu'on pourra rappeler avec la fonction de rappel du shell (typiquement la flèche vers le haut), autant de fois qu'il faut. Ici, par exemple, on peut utiliser :

```
echo Test | mail -s "Essai" user@example.org
```

`echo` envoie le corps du message à la commande `mail` qui transmettra le tout à `user@example.org`. (Au fait, pour tester un serveur de messagerie, vous aurez peut-être besoin de répondeurs automatiques <<https://www.bortzmeyer.org/repondeurs-courrier-test.html>>.)

Cette méthode simplifie la vie du testeur : il n'a que deux touches à taper (flèche vers le haut et <Entrée>) pour refaire un test. Elle permet en outre d'utiliser des variables qui peuvent être bien pratiques lors de l'examen du courrier envoyé, par exemple :

```
echo Test | \
  mail -s "Essai depuis le processus $$, utilisateur $(id -u)" \
    user@example.org
```

Autre avantage : si on demande de l'aide sur une liste de diffusion ou un forum Web, il sera plus facile d'indiquer la commande réellement effectuée pour le test (beaucoup de messages sur de tels forums sont inutilisables car le questionneur a juste dit « Ça ne marche pas » au lieu de reproduire littéralement et intégralement la commande tapée et le résultat obtenu).

Mais il y a une autre raison pour laquelle la technique de la commande sur une ligne `<https://www.bortzmeyer.org/command-line.html>` est préférable, une raison sans doute plus importante que la seule facilité de frappe. Les actions menées interactivement sont difficiles à reproduire. Si on est lancé dans un cycle `essai -z` modification des paramètres `-z` nouvel essai, il est primordial que les essais soient absolument identiques, pour qu'on ne fasse pas varier à la fois la configuration et le test. C'est un principe de base de la méthode scientifique. L'informatique étant une science expérimentale, suivre cette méthode est nécessaire.

J'ai vu souvent des stagiaires croire que la modification qu'ils venaient d'appliquer à leurs fichiers de configuration avait provoqué un changement, alors qu'ils s'étaient simplement trompés en essayant de reproduire le test. Avec une commande d'une seule ligne, qu'on rappelle à l'identique, on a la garantie que le test ne change que si on le veut explicitement.

Bien sûr, ce conseil ne s'applique pas qu'à la mise au point des serveurs de messagerie. Les serveurs Web sont également concernés. Il ne faut évidemment pas les tester avec un navigateur (je parle bien de la configuration du serveur HTTP, pour le contenu de la page, HTML et CSS, c'est évidemment différent), outil lent, lourd et qui fait bien des choses sans le dire, rendant le test plus difficile à interpréter. Les meilleurs outils de mise au point sont des programmes en ligne de commande comme `curl` et `wget`. Par exemple, avec `wget` :

```
wget -O /dev/null http://www.example.org/
```

`teste http://www.example.org/` sans sauvegarder la page (option `-O` ou, en version longue, `--output-document`). Parmi les choses qu'affiche `wget` par défaut, le code de statut HTTP (200 si tout va bien) tel que normalisé dans le RFC 7231<sup>1</sup>. Si on veut davantage de détails, on peut utiliser l'option `-S` (ou, en version longue, `--server-response`) qui affiche tous les en-têtes HTTP. Mais attention : un test trop bavard n'est pas une bonne idée car l'information importante risque d'être noyée au milieu de tout ce qui est affiché.

Autre logiciel de test de serveurs HTTP, `curl` :

```
curl http://www.example.org/ > /dev/null
```

Cette commande n'affiche pas le code de retour mais on peut changer cela :

```
curl --write-out 'HTTP status: %{http_code}\n%{size_download} bytes downloaded\n' \
  http://www.example.org/ >
```

`curl` va alors écrire le code de statut HTTP et le nombre d'octets transférés.

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7231.txt>