Tirer davantage du shell Unix avec les alias

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 9 juin 2008

https://www.bortzmeyer.org/shell-alias.html

Beaucoup d'utilisateurs du shell Unix ne se servent jamais d'**alias** ni de fonctions. Ils tapent et retapent de longues commandes compliquées, sans jamais consacrer une demi-journée à apprendre les possibilités avancées du shell. « Je n'ai pas le temps » est un excuse souvent entendue. Mais, s'ils calculaient le Temps Total (comme on définit désormais un Coût Total), peut-être trouveraient-ils que cette demi-journée d'apprentissage leur ferait gagner pas mal de temps pour leurs tâches futures. Lorsque je montre une commande compliquée à des utilisateurs Unix, leur réponse est souvent « Mais je ne vais pas taper un truc aussi long à chaque fois ! ». Bien sûr, c'est pour cela que les alias existent.

Les alias sont une fonction du shell qui permet de définir un équivalent court pour une commande. Si on veut simplement mettre quelques options, toujours les mêmes, certaines commandes Unix permettent de les fixer dans un fichier de configuration. Mais toutes les commandes n'ont pas cette possibilité. Et elle ne résout pas tous les problèmes. Si je veux enchaîner plusieurs commandes ? Si je veux utiliser parfois certaines options et parfois d'autres ? Je peux écrire un script, bien sûr, mais les alias sont une solution plus légère.

La définition d'un alias dépend légèrement du shell utilisé (en gros, la famille sh et la famille csh divergent). Tous les exemples qui suivent ont été testés avec zsh, un shell de la famille sh.

On définit un alias avec la commande `dumłmenom` :

```
% alias dir="ls -lt"

% dir
total 4
-rw-r--r-- 1 stephane stephane 30 Jun  9 16:55 truc.c
...
```

Si on veut que cette définition soit permanente, on peut mettre la commande `alias` dans le fichier de démarrage (pour zsh, en `/.zshrc`).

Voici quelques alias que j'utilise souvent. Taper `alias NOM` va afficher la définition de l'alias de ce nom.

Pour voir facilement les en-têtes renvoyés par un serveur HTTP, ce qui aide souvent au débogage, faire un telnet à la main est une solution mais elle est souvent pénible. wget fournit ce service, s'il est appelé avec les bonnes options :

```
% alias hh
hh='wget --server-response --output-document /dev/null'

% hh http://www.example.org/
...
  HTTP/1.1 200 OK
  Date: Mon, 09 Jun 2008 15:02:40 GMT
  Server: Apache/2.2.8 (Ubuntu) mod_python/3.3.1 Python/2.5.2 PHP/5.2.4-2ubuntu5.1 with Suhosin-Patch mod_pe
  Last-Modified: Wed, 19 Sep 2007 06:43:55 GMT
  ETag: "17c0cf-203-43a775c8b60c0"
  Content-Length: 515
  Content-Type: text/html
...
```

J'utilise, dans la définition de l'alias, les noms longs des options (`--server-response` au lieu de juste `-S` car l'alias est ainsi mieux documenté. Les noms courts des options sont très pratiques en interactif mais déconseillés dans un alias ou un script.

Un alias peut être composé de plusieurs commandes, reliées par un point-virgule (exécution séquentielle) ou un tube (execution séquentielle et la sortie de la première commande est l'entrée de la seconde). Ainsi, pour voir les 20 derniers fichiers chargés via mlDonkey :

```
 % alias lastdonkey
lastdonkey='ls -lt /var/lib/mldonkey/ | head -n 20'
```

Il n'y a pas de limite au nombre de commandes utilisables. Ici, l'alias `svnpurge` supprime dans une copie de travail Subversion, tous les fichiers qui ne sont **pas** gérés par ce logiciel (attention, c'est une commande dangereuse). Elle est définie ainsi dans mon `/.zshrc` :

```
alias svnpurge="svn status | awk '/^\?/ {print $2}' | xargs rm -f"
```

Comment marche t-elle ? `svn status` affiche les fichiers non gérés par Subversion avec un point d'interrogation. `awk` les extrait et `xargs` appelle `rm` pour les détruire.

Après, si on veut des paramètres, des structures de contrôle un peu compliquées, il faut mieux passer aux fonctions `<https://www.bortzmeyer.org/shell-functions.html>`.