

Garder N copies d'un fichier avec une commande Unix simple

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 septembre 2010. Dernière mise à jour le 20 septembre 2010

<https://www.bortzmeyer.org/copy-n.html>

Dans la série « Partageons nos petits scripts shell et le monde sera meilleur », voici un petit programme qui permet de renommer des fichiers journaux ou des sauvegardes, un peu comme le fait `logrotate`, mais sans fichier de configuration, uniquement sur la ligne de commande.

Il me sert dans le cas où j'ai besoin de garder les N dernières copies d'un fichier. Par exemple, sur un serveur de noms, il est utile de garder localement les dernières versions du fichier de zone, même si celui-ci est généré depuis une base de données, afin de pouvoir repartir rapidement en cas de pannes comme celle de .SE. De même, lorsqu'on sauvegarde ses données stockées dans le nuage, il est utile de garder les précédentes versions, au cas où la copie du nuage ait été tronquée. Voici par exemple comment je sauvegarde mes marque-pages chez `del.icio.us` :

```
copy-n bookmarks-at-delicious
# http://del.icio.us/help/api/
wget -O bookmarks-at-delicious --no-check-certificate \
    --http-user=bortzmeyer --http-passwd=jevaispasledire \
    'https://api.del.icio.us/v1/posts/all'
```

`copy-n` fait donc l'équivalent de :

```
cp $LOG.1 $LOG.2
cp $LOG.0 $LOG.1
cp $LOG $LOG.0
touch $LOG
```

mais en plus paramétrable (nombre de copies gardées avec l'option `-n, mode`), et en plus robuste (le code ci-dessus plante si `$LOG.3` n'existe pas), etc.

Le programme complet, normalement en shell portable, est disponible (en ligne sur <https://www.bortzmeyer.org/files/copy-n>). Voici un exemple d'usage :

```
% copy-n -n 2 -v sources.org
Moving sources.org.1 to sources.org.2...
Moving sources.org.0 to sources.org.1...
Copying sources.org to sources.org.0...
```

Outre `logrotate`, déjà cité (son principal avantage sur `copy-n` est l'option de compression), on peut bien sûr préférer un VCS ou un autre programme comme `rdiff-backup` (<http://rdiff-backup.nongnu.org/>), suggéré par Sébastien Sauvage, mais qui a un cahier des charges assez différent, puisqu'il ne stocke pas les versions précédentes en clair et qu'il ne fonctionne qu'avec des répertoires, pas de simples fichiers). Autre possibilité, `savelog` (<http://manpages.debian.net/cgi-bin/man.cgi?query=savelog>), spécifique à Debian (paquetage `debianutils` (<http://packages.debian.org/debianutils>), mais les sources sont disponibles), proposé par Gregory Colpart. Prévu uniquement pour les journaux, il a quelques limites pour d'autres utilisations mais, autrement, semble très satisfaisant. Autres possibilités : les traditionnels `cp` et `rsync` disposent d'options qui font la moitié du travail souhaité. GNU `cp` a l'option (non-standard) `--backup` donc on peut écrire des choses comme `cp --backup=numbered monfichier masauvegarde` et il gardera alors les précédentes versions de `masauvegarde`, agrémentées d'un `~` et d'un numéro (qui suit l'ordre inverse de celui qui est habituel : le numéro le plus élevé est le fichier le plus récent). Il n'y a apparemment pas de moyen de ne garder que les `N` dernières versions donc on remplit le disque petit à petit. C'est encore pire avec `rsync` qui ne permet de garder que la toute dernière version (`rsync --backup --backup-dir=backups monfichier masauvegarde`). Les autres suggestions de logiciels et argumentations sur les raisons du choix sont les bienvenues.

En attendant, merci à Mathieu Arnold, Marc Baudoin et Kim-Minh Kaplan pour leur aide en programmation shell, surtout pour les calculs arithmétiques et à Jean-Philippe Pick pour sa maîtrise de `cp`.