

Bernard Fabrot

Aide-mémoire

Unix pour Mac OS X

© Éditions OEM (Groupe Eyrolles), 2003

ISBN 2-7464-0492-3

OEM

EYROLLES

sudo

Syntaxe

sudo commande

sudo [paramètres] commande

Description

La commande *sudo* permet d'exécuter une commande quelconque comme si l'on était un autre utilisateur.

Paramètre

-u *utilisateur* effectue la commande comme si c'était l'utilisateur spécifié qui l'appelait

Exemple

```
[pierre ~/] $ ls /Users/carrera/Music/
```

```
ls: : Permission Denied
```

```
[pierre ~/] $ sudo carrera ls /Users/carrera/Music/
```

```
Password:
```

```
musiqueFilm01.mp3 MichiganAndSmileyRubaDubStyle.ogg
```

```
[pierre ~/] $
```

L'utilisateur *pierre* n'a pas le droit de lister le contenu du répertoire *Music/* de l'utilisateur *carrera*. La commande *sudo* permet de pallier ce problème.

tail

Syntaxe

tail [paramètres] fichier

Description

La commande *tail* affiche les informations présentes à la fin d'un fichier texte. Par défaut, les dix premières lignes sont affichées.

Paramètres

- c *num* affiche les num derniers octets
- n *num* affiche les num dernières lignes
- r inverse le résultat de la sortie

Exemple

```
[pierre ~/] $ tail -n 6 /var/log/weekly.out
Subject: Ordinateur-de-Virga weekly run output
Rebuilding locate database:
_su: Trying to start from "/var/root"
_su: Trying to start from "/"
Rebuilding whatis database:
Rotating log files: ftp.log lookupd.log lpr.log mail.log netinfo.log
```

On affiche les six dernières lignes du fichier */var/log/weekly.out*

Remarque

La commande *tail* est souvent utilisée pour consulter les fichiers de log car les informations les plus récentes sont toujours ajoutées à la fin de ceux-ci.

Voir head (p. 129)

tar

Syntaxe

tar [paramètres] -f archive [fichier(s)]

Description

La commande *tar* permet de créer un fichier d'archive contenant un ou, vraisemblablement, plusieurs autres fichiers. Le fichier d'archive ainsi créé n'est pas compressé (à cet effet, l'on peut utiliser *gzip* ou *bzip2*, voir pp. 127 et 71).

Cette même commande *tar* est utilisée, avec d'autres paramètres, pour décompresser un fichier d'archive.

Paramètres

- c crée une nouvelle archive ou ajoute des fichiers à une archive déjà existante
- f spécifie le nom du fichier archive avec lequel on travaille
- x extrait les fichiers de l'archive
- t affiche la liste des fichiers de l'archive
- v affiche plus d'informations à l'écran quant à ce que la commande effectue comme manipulations
- z compresse l'archive en utilisant *gzip*
- Z compresse l'archive en utilisant *compress*

Exemples

```
[pierre ~/test/] $ ls-l
total 1048
```

```

-rwxr-xr-x 1 pierre staff 1408 Mar 17 18:18 cvGerald.pdf
-rw-r--r-- 1 pierre staff 200 Mar 29 11:45 essai
-rw-r--r-- 1 pierre staff 66272 Mar 29 12:35 photo01.gif
-rw-r--r-- 1 pierre staff 48496 Mar 29 12:35 photo01.jpg
-rw-r--r-- 1 pierre staff 397632 Mar 29 12:36 photoHiRes.tiff
-rw-r--r-- 1 pierre staff 73 Mar 17 18:15 rot13.sh
-rw-r--r-- 1 pierre staff 54 Mar 17 18:00 toDo.txt
[pierre ~/test/] $ tar -cf archive.tar *

```

```
[pierre ~/test/] $ ls -l
```

```

total 2072
-rw-r--r-- 1 pierre staff 522240 Mar 29 12:36 archive.tar
-rwxr-xr-x 1 pierre staff 1408 Mar 17 18:18 cvGerald.pdf
-rw-r--r-- 1 pierre staff 200 Mar 29 11:45 essai
-rw-r--r-- 1 pierre staff 66272 Mar 29 12:35 photo01.gif
-rw-r--r-- 1 pierre staff 48496 Mar 29 12:35 photo01.jpg
-rw-r--r-- 1 pierre staff 397632 Mar 29 12:36 photoHiRes.tiff
-rw-r--r-- 1 pierre staff 73 Mar 17 18:15 rot13.sh
-rw-r--r-- 1 pierre staff 54 Mar 17 18:00 toDo.txt

```

En appelant la commande « `tar -cf archive.tar *` », l'utilisateur *pierre* place tous les fichiers du répertoire courant dans un seul fichier d'archive (les anciens fichiers ne disparaissent pas. On peut constater que la taille du fichier `archive.tar` est environ équivalente à la somme des tailles de tous les autres fichiers (les fichiers sont placés les uns derrière les autres dans l'archive, sans être compressés).

```
[pierre ~/test/] $ tar -tf archive.tar
cvGerald.pdf
essai
photo01.gif
photo01.jpg
photoHiRes.tiff
rot13.sh
toDo.txt
```

L'option « -t » permet de consulter les fichiers présents dans un fichier d'archive, sans avoir à les décompresser.

```
[pierre ~/test/] $ ls -l archive.tar
-rw-r--r-- 1 pierre staff 522240 Mar 29 12:36 archive.tar
[pierre ~/test/] $ bzip2 archive.tar
[pierre ~/test/] $ ls -l archive.tar.bz2
-rw-r--r-- 1 pierre staff 135890 Mar 29 12:36 archive.tar.bz2
```

Le fichier *archive.tar* est compressé à l'aide de *bzip2* (voir p. 71), sa taille passe de 522240 octets à 135890 octets.

Remarques

- On donne bien souvent l'extension *.tar* à un fichier d'archive.
- Il s'agit d'une commande très puissante et très pratique pour effectuer des copies de sécurité des fichiers. Il est courant de graver les fichiers d'archives tar ou d'en placer une copie sur une autre machine (à l'aide, par exemple, de *scp*).

tcpdump

Syntaxe

```
tcpdump
```

Description

L'utilitaire *tcpdump* permet d'analyser tous les paquets d'informations transitant sur une interface réseau.

On peut utiliser ce programme pour résoudre un problème de connexion (problème de configuration, problème de *firewall*, etc.) ou pour analyser le trafic partant et arrivant à l'ordinateur (par exemple pour essayer de détecter une intrusion).

Il s'agit d'une commande extrêmement complexe disposant de nombreuses options. Pour en prendre connaissance, reportez-vous aux pages *man* de *tcpdump* (voir *man* p. 169).

Exemple

```
pierre ~/test/] $ sudo tcpdump -n
Password:
tcpdump: listening on eno
14:01:16.076264 192.168.1.5.32768 > 195.238.2.21.53: 22134+ A? iki.fi. (24) (DF)
14:01:16.093325 192.168.1.5 > 212.16.100.1: icmp: echo request (DF)
14:01:16.164787 212.16.100.1 > 192.168.1.5: icmp: echo reply
...
```

L'utilisateur *pierre* lance la commande *tcpdump* en tant que super-utilisateur (voir *sudo* p. 205). On peut constater qu'un ordinateur du réseau local (192.168.1.5) est occupé à lancer des « ping » (voir *ping* p. 180) vers un ordinateur distant (212.16.100.1).

tcsh

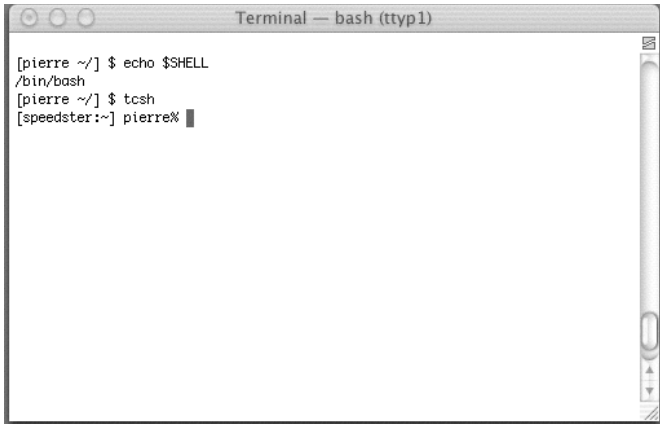
Syntaxe

tcsh

Description

La commande *tcsh* lance un nouveau shell de type *tcsh*. Le shell *tcsh* est le shell par défaut des utilisateurs de Mac OS X.

La façon la plus facile de reconnaître un shell *tcsh* est grâce à l'invite qu'il utilise : par défaut, il s'agit du caractère « % ».



```
Terminal — bash (tty1)
[pierre ~/] $ echo $SHELL
/bin/bash
[pierre ~/] $ tcsh
[speedster:~] pierre%
```

L'utilisateur lance, depuis un shell bash, un shell tcsh.

Remarque

Le shell *bash* est plus utilisé (surtout pour les scripts) que le shell *tcsh*, mais il n'est pas configuré par défaut. Pour configurer le shell *bash*, reportez-vous aux explications page 25.

telnet

Syntaxe

```
telnet [paramètres] ordinateur [port]
```

Description

La commande *telnet* permet d'établir une connexion avec un ordinateur distant. Il est ainsi possible, par exemple, d'ouvrir un shell sur un autre système Unix.

Exemple

```
[pierre ~/test/] $ telnet relay.skynet.be 25
Connected to smtp.pool.skynet.be.
Escape character is '^'.
220 vador.skynet.be ESMTP Sendmail 8.12.8/8.12.8/Skynet-OUT-2.21; Sat, 29 Mar
quit
221 2.0.0 vador.skynet.be closing connection
Connection closed by foreign host.
```

L'utilisateur *pierre* se connecte sur le port 25 d'un serveur (il s'agit du port destiné à l'envoi d'email par le protocole *smtp*) à l'aide de la commande *telnet* et se déconnecte de suite.

Remarque

Il s'agit d'une commande Unix très ancienne qui pose de nombreux problèmes de sécurité. Les utilisateurs préfèrent actuellement se tourner vers une alternative sécurisée telle *ssh*.

Voir *ssh* (p. 201)