Licence d'informatique
Module de C/Unix

Projet Unix

real-file — Localisation d'un fichier

Philippe MARQUET

Octobre 1995

Le but est d'écrire un shell-script indiquant la machine sur laquelle se trouve un fichier de nom donné en paramètre.

Dans une première partie, à partir d'un nom[1] d'entrée[2], on recherche un nom absolu désignant le fichier tel qu'aucune des composantes de ce nom ne soit un lien symbolique. La commande `real-name` que vous allez écrire assure cette fonctionnalité.

Soient les entrées suivantes dans le système de fichiers :

```
% ls -ld /
drwxr-xr-x  18 root     1024     May  4 14:12 /
% ls -ld /usr
drwxr-xr-x  20 root     512      May  4 17:02 /usr
% ls -ld /usr/local
lrwxrwxrwx  21 root     512      May  6 14:11 /usr/local -> /usr/local.luxe
% ls -ld /usr/local.luxe
drwxr-xr-x   2 root     1536     May  6 14:16 /usr/local.luxe
% ls -ld /usr/local/bin
drwxr-xr-x   2 root     1536     May  9 14:16 /usr/local/bin
% ls -ld /usr/local/emacs
drwxr-xr-x   3 root     512      May  9 14:24 /usr/local/emacs
% ls -ld /usr/local/emacs/19.28
drwxr-xr-x   3 root     512      May  9 14:25 /usr/local/emacs/19.28
% ls -ld /usr/local/emacs/19.28/bin
drwxr-xr-x   3 root     512      May  9 14:27 /usr/local/emacs/19.28/bin
% pwd
/usr/local
% ls -l bin/emacs*
lrwxrwxrwx   2 root     13       May  9 14:42 emacs -> ./emacs-19.28
lrwxrwxrwx   2 root     24       May  9 14:42 emacs-19.28 -> ../emacs/19.28/bin/emacs
% ls -l emacs/19.28/bin/emacs
-rwxr-xr-t   2 root     3366912 May  9 14:38 emacs
```

Exercice 1 (Arborescence du système de fichiers)
Donnez un schéma du système de fichiers sur lequel les commandes précédentes ont été appliquées. Vous ferez figurer les liens symboliques par une représentation adéquate dans l'arbre.

Sur ce système de fichiers, la commande `real-name` est illustrée par les exemples suivants :

```
% real-name /usr/local/emacs/19.28/bin/emacs
/usr/local.luxe/emacs/19.28/bin/emacs
% real-name /usr/local/bin/emacs
/usr/local.luxe/emacs/19.28/bin/emacs
% cd /usr
% real-name local
/usr/local.luxe
% cd /usr/local
```

[1] nom absolu ou nom relatif
[2] fichier ou répertoire

Id : realfile.tex,v 1.5 2000/12/03 10 :47 :49 marquet Exp

```
% real-name bin/emacs-19.28
/usr/local.luxe/emacs/19.28/bin/emacs
% real-name .
/usr/local.luxe
```

Exercice 2 (Algorithme `real-name`)
Donnez une description de l'algorithme de la commande `real-name`. Dans une première étape vous pouvez ne pas considérer les entrées référençant le répertoire courant (`.`) ou le répertoire père (`..`).

Exercice 3 (Commande `real-name`)
Donnez l'implémentation en Bourne Shell de l'algorithme proposé à l'exercice précédent.

On considère maintenant les partitions disques. La commande `mount`[3] affiche les partitions montées sur le système de fichiers.

```
% mount
/dev/rz3a on / type ufs (rw)
/dev/rz3g on /usr type ufs (rw)
/dev/rz3h on /mnt type ufs (rw)
lifl:/export/home1/west on /tmp_mnt/home/west type nfs (v2, rw, soft, intr)
/var/spool/mail@lifl on /usr/var/spool/mail type nfs (v2, rw, soft, intr)
/export/local.luxe@luxe on /usr/local.luxe type nfs (v3, ro, soft, intr)
```

La syntaxe d'une ligne du résultat produit par la commande `mount` est la suivante :

$$< \textit{partition} > \texttt{on} < \textit{point-de-montage} > \texttt{type} < \textit{type} >< \textit{options} >$$

Un *point-de-montage* est un nom absolu dans l'arborescence du système de fichiers sous lequel la partition sera montée. Un montage peut éventuellement masquer une partie du système de fichiers si ce nom absolu correspond à celui d'un répertoire non vide.

Trois syntaxes sont acceptées pour *partition* :

$$< \textit{machine-distante} > \textbf{:} < \textit{répertoire-distant} > \tag{1}$$

$$< \textit{répertoire-distant} > \texttt{@} < \textit{machine-distante} > \tag{2}$$

$$< \textit{disque} > \tag{3}$$

La syntaxe 3 désigne un disque situé physiquement sur la machine locale. Les syntaxes 1 et 2 sont équivalentes et autorisent le montage de répertoires ¡¡ exportés ¿¿ par une machine distante.

Notez qu'une machine n'a pas le droit de ¡¡ ré-exporter ¿¿ une partition montée depuis une autre machine.

Exercice 4 (Système de fichiers)
Complétez le schéma de l'arborescence du système de fichiers donné précédemment pour y faire figurer les partitions.

Le but de la commande `display-partition` est d'afficher la partition correspondant à un nom d'entrée donné en paramètre. Ce nom de partition est recherché à partir du nom retourné par la commande `real-name` sur le paramètre.

Sur l'exemple précédent, on obtient les résultats suivants :

```
% display-partition /usr/local
/export/local.luxe@luxe
% display-partition /usr/local/bin/emacs
/export/local.luxe@luxe
```

Exercice 5 (Algorithme de la commande `display-partition`)
Donnez une description de l'algorithme de la commande `display-partition`.

[3]Selon les systèmes, il s'agit de `/etc/mount` ou de `/sbin/mount`.

Exercice 6 (Commande `display-partition`)
Donnez l'implémentation en Bourne Shell de l'algorithme proposé à l'exercice précédent.

Le résultat retourné par la commade display-partition peut comporter des composantes qui sont des liens symboliques. L'option `-t` (*−true*) de la commande `display-partition` assure qu'aucune des composantes de ce résultat n'est un lien symbolique. Pour cela, il est nécessaire d'exécuter des commandes sur la machine distante exportant la partition. On utilise alors une commande à distance (*remote-command*). La syntaxe de `rsh` utilisée sera la suivante :

`rsh` $<$ *machine-distante* $><$ *commande* $>$ `[` $<$ *argument* $>$ `]`

Pour que cette commande puisse s'exécuter, il peut être nécessaire d'ajouter une ligne de la forme suivante dans le fichier `$HOME/.rhosts` (*cf.* `man rhosts`) :

$<$ *machine-locale* $><$ *utilisateur* $>$

On veillera à restituer le fichier `$HOME/.rhosts` dans son état initial après l'exécution de la commande à distance.

Exemple :

```
% hostname
bock
% display-partition /usr/local
/export/local.luxe@luxe
% rsh luxe ls -ld /export/local.luxe
permission denied
% cat >> ~/.rhosts
bock marquet
% rsh luxe ls -ld /export/local.luxe
drwxr-xr-x   2 root     1236    May  2 12:14 /export/local.luxe
```

Il est possible que l'utilisateur n'ait pas le droit de lancer une commande distante sur la machine sur laquelle réside la partition référencée, par exemple parce qu'il n'a pas le droit de se connecter sur le serveur exportant cette partition. Dans ce cas, la commande `display-partition -t` l'indiquera par une valeur de retour appropriée (*i.e.* strictement positive).

Exercice 7 (Option `-t` de `display-partition`)
Donner l'algorithme et son implémentation en Bourne Shell de la commande `display-partition [-t]`.

À partir des informations précédentes, on développe maintenant la commande `real-file` qui affiche le nom d'une entrée dans le système de fichiers passée en paramètre sous la forme

$<$ *nom-de-fichier* $>$ `@` $<$ *machine* $>$

telle que le nom de fichier retourné corresponde à une entrée dans le système de fichiers monté sur une partition locale à la machine et telle qu'aucune des composantes de ce nom ne soit un lien symbolique.

Voici quelques exemples d'utilisation de cette commande[4] :

```
% real-file /usr/local/bin/emacs
/export/local.luxe/emacs/19.28/bin/emacs@luxe
% hostname
bock
% real-file /usr
/usr@bock
```

[4] On suppose que `/export/local.luxe/emacs/19.28/bin/emacs` sur la machine `luxe` ne comporte pas de liens symboliques

Pour le cas ou l'utilisateur ne peut avoir confirmation que le nom retourné par la commande `display-partition -t` ne contient pas de liens symboliques, la commande `real-file` retourne une valeur strictement positive.

Exercice 8 (Algorithme de la commande `real-file`)
Donnez une description de l'algorithme de la commande `real-file`

Exercice 9 (Commande `real-file`)
Donnez l'implémentation en Bourne Shell de l'algorithme proposé à l'exercice précédent.

On étend maintenant la commande `real-file` pour supporter l'option `-d` (*-disk*) qui retourne le résultat sous la forme

$$< \textit{nom-de-fichier} > \texttt{@} < \textit{machine} > \texttt{:} < \textit{disque} >$$

indiquant le nom du disque physique sur lequel le fichier est présent sur la machine donnée. Il est nécessaire de connaître le résultat de la commande `mount` sur cette machine donnée. Pour ce faire, on peut à nouveau être amené à utiliser une commande à distance (*remote-command*).

Exemple :

```
% hostname
bock
% real-file /usr/local/bin/emacs
/export/local.luxe/emacs/19.28/bin/emacs@luxe
% rsh luxe /etc/mount
/dev/rz3a on / type ufs (rw)
/dev/rz3g on /usr type ufs (rw)
/dev/rz3b on /export type ufs (rw)
/dev/rz3h on /mnt type ufs (rw)
lifl:/export/home1/west on /tmp_mnt/home/west type nfs (v2, rw, soft, intr)
/var/spool/mail@lifl on /usr/var/spool/mail type nfs (v2, rw, soft, intr)
% real-file -d /usr/local/bin/emacs
/export/local.luxe/emacs/19.28/bin/emacs@luxe:/dev/rz3b
```

Si l'utilisateur n'a pas le droit de lancer une commande distante sur la machine sur laquelle réside la partition référencée, la commande `real-file -d` l'indiquera par une valeur de retour appropriée (*i.e.* strictement positive) et fournira le résultat sous la forme

$$< \textit{nom-de-fichier} > \texttt{@} < \textit{machine} > \texttt{:}$$

Exercice 10 (Option `-d` de `real-file`)
Donner l'algorithme et son implémentation en Bourne Shell de la commande `real-file [-d]`.

Réalisation du projet

Vous devez rendre (sous une forme à définir avec votre enseignant de TD/TP) :

- un compte-rendu de projet incluant les réponses aux exercices du présent document (dessins des systèmes de fichiers, algorithmes des commandes) ;
- les sources shell des commandes développées ;
- un test de ce projet sous la forme de résultats d'exécution des commandes.