Licence d'informatique
Module de C/Unix

Localiser un fichier

Examen – Partie Unix

Philippe MARQUET

Juin 1995

Les documents de cours et TD sont autorisés.
On rendra deux copies séparées pour la partie Unix et pour la partie langage C.

1 Questions de cours

Question 1 (Système ouvert/système propriétaire)
Chapitre ¡¡ Introduction — Historique & Philosophie ¿¿
Caractérisez un système propriétaire (par rapport à Unix qui est un système ouvert).

Question 2 (`rehash`)
Chapitre ¡¡ Interpréteur de commandes ¿¿
Donner un extrait de session C-shell nécessitant l'utilisation de la commande `rehash`.

2 Où est le fichier ?

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

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 (questions 4 à 6) 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
% cd /usr/local
% ls -l bin/emacs*
```

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

Id : xam95.tex,v 1.5 2000/11/28 18 :34 :55 marquet Exp

```
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
```

Question 3 (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
% real-name bin/emacs-19.28
/usr/local.luxe/emacs/19.28/bin/emacs
% real-name .
/usr/local.luxe
```

Question 4 (Algorithme `real-name`)
Donnez une description de l'algorithme de la commande `real-name`. Dans cette première version, vous ne considérerez pas les entrées référençant le répertoire courant (`.`) ou le répertoire père (`..`). Le résultat affiché par `real-name` pourra donc comporter de telles composantes.

Question 5 (Commande `real-name`)
Donnez l'implémentation en Bourne Shell de l'algorithme proposé à la question précédente.

Question 6 (Commande `real-name` étendue)
La commande doit maintenant prendre en compte les composantes de nom `.` et `..`. Le résultat affiché ne comportera donc pas de telles composantes.

On considère maintenant les partitions disques. La commande `mount` 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)
/usr/local.luxe@luxe on /usr/local.luxe type nfs (v3, rw, soft, intr)
```

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

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

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*> :

$$< \textit{machine-distante} > \texttt{:} < \textit{répertoire-distant} > \quad (1)$$

$$< \textit{répertoire-distant} > \texttt{@} < \textit{machine-distante} > \quad (2)$$

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

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.

Question 7 (Système de fichiers)
Complétez le schéma de l'arborescence du système de fichiers donné précédemment pour y faire figurer les partitions.

Les questions précédentes constituent le sujet d'examen. Il est facultatif de répondre aux questions suivantes. La lecture de la fin de cette section n'est pas même indispensable à la compréhension du sujet.

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

Question 8 (Algorithme de la commande `display-partition`)
Donnez une description de l'algorithme de la commande `display-partition`

Question 9 (Commande `display-partition`)
Donnez l'implémentation en Bourne Shell de l'algorithme proposé à la question précédente.

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

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

telle que le nom de fichier retourné corresponde à une entrée dans le système de fichiers monté sur une partition locale à la machine.

Voici quelques exemples d'utilisation de cette commande :

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

Question 10 (Algorithme de la commande `real-file`)
Donnez une description de l'algorithme de la commande `real-file`.

Question 11 (Commande `real-file`)
Donnez l'implémentation en Bourne Shell de l'algorithme proposé à la question précédente.

3 Aide

Les quelques commandes suivantes peuvent vous être utiles.

```
% echo lrwxrwxrwx | cut -c1
l
% echo -rwxr-xr-t | cut -c1
-
% echo /usr/local/emacs/19.28/bin/emacs | tr / " "
 usr local emacs 19.28 bin emacs
```