Licence d'informatique
Module de C/Unix

Sauvegarde et restauration

Examen – Partie Unix

Philippe MARQUET

Septembre 1995

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

Le but de cet exercice est de proposer une commande `sauvegarde` et une commande `restaure` permettant de sauvegarder les différences entre une arborescence de fichiers et l'arborescence originelle dans un fichier de modifications d'une part, et de reconstruire une arborescence à partir d'une arborescence originelle et d'un fichier de modifications d'autre part.

Afin de faciliter les choses, certaines conventions sont systématiquement appliquées lors de la modification de l'arborescence. On qualifie une entrée dans le système de fichiers de *nouvelle* si et seulement si elle n'est pas située sous un répertoire de nom `rep` tel qu'il existe une entrée `rep.ajout` dans le même répertoire que `rep`.

Par exemple, dans le système de fichiers sur lequel sont appliquées les commandes suivantes :

```
% ls -F /
dev/            home/           tmp/            var/
etc/            home.ajout@     usr/            vmunix*
% ls -F /home
dea/            enseign/        maitrise/
dess/           licence/
% ls -F /home/licence
dupond/         durand/         martin/
dupont/         durant/
% ls -F /usr
bin/            include/        man/            tmp/
doc/            lib/            sys/
% ls -F /etc
exports         hosts           passwd          termcap
fstab           hosts.equiv     printcap
group           motd            securetty
```

les entrées `/home/licence`, `/home/maitrise`, `/home/licence/martin` ne sont pas définies comme *nouvelles*, car elles sont situées sous le répertoire `/home` et l'entrée `/home.ajout` existe. Par contre les entrées `/usr/include` et `/etc/hosts` sont définies comme *nouvelles*. De même, lors de sa création, l'entrée `/usr/local` sera qualifié de *nouvelle*.

Les conventions systématiquement appliquées sont les suivantes :

1. Lors de la création d'une *nouvelle* entrée dans le système de fichiers, on crée un lien symbolique ¡¡ `.ajout` ¿¿ vers cette entrée. Par exemple les deux commandes suivantes seront utilisées lors de la création du répertoire `/usr/local` :

   ```
   % mkdir /usr/local
   % ln -s /usr/local /usr/local.ajout
   ```

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

Par contre, lors de la création du répertoire `/usr/local/tex`, le lien symbolique `/usr/local/tex.ajout` ne sera pas créé ; `/usr/local/tex` n'étant pas une *nouvelle* entrée.

2. Lors de la première modification d'un fichier non défini comme une *nouvelle* entrée, la version originelle du fichier est préalablement copiée dans un fichier `.orig`, par exemple :

```
% cp /etc/hosts /etc/hosts.orig
% emacs /etc/hosts
```

3. (Par soucis de simplicité, et dans un premier temps, nous supposons qu'aucune destruction d'entrée dans le système de fichiers n'a lieu.)

La commande `sauvegarde`

Question 1
Explicitez la liste des entrées du système de fichiers qui doivent être sauvegardées par la commande `sauvegarde`.

Question 2
Donner le code Bourne shell de la fonction `liste-entrees` qui affiche cette liste.

La commande `tar` permet de sauvegarder un ensemble de fichiers ou répertoires (*`file`*`...`) sous forme d'un fichier archive (*`archive-file`*). Nous utiliserons l'option `cf` de `tar` dont la syntaxe est la suivante :

`tar cf` *`archive-file file`*`...`

La sauvegarde d'un répertoire mémorise l'ensemble de la sous-arborescence dont il est la racine dans le fichier archive.

Question 3
Donner le code Bourne Shell de la commande `sauvegarde` qui produit un fichier archive dont le nom est passé en paramètre.

La commande `restaure`

Pour reconstruire une arborescence à partir d'une arborescence originelle et d'un fichier archive, nous allons utiliser les deux syntaxes suivantes de la commande `tar` :

1. Pour lister les noms d'entrées contenues dans le fichier archive *`archive-file`*, nous utilisons la syntaxe :

   `tar tf` *`archive-file`*

2. Pour restaurer l'ensemble des entrées contenues dans un fichier archive, nous utilisons la syntaxe :

   `tar xf` *`archive-file`*

   Notons que les entrées déjà existantes dans le système de fichiers sont écrasées.

Il nous faut, pour chacune des entrées contenues dans le fichier archive qui vont être restaurées, respecter les conventions définies (créations de `.ajout` et `.orig`).

Question 4
Expliciter le traitement nécessaire sur chacune de ces entrées.

Question 5
Donner le code Bourne shell de la commande `restaure` à laquelle on passe le nom d'un fichier d'archive en paramètre.

Pour quelques points supplémentaires

Question 6 (facultatif)
Proposer une solution pour la gestion des entrées du système de fichiers non définies comme *nouvelles* qui sont détruites ou déplacées.