Licence d'informatique
Module de C/Unix

Vérification de duplication d'exécutables, distribution de logiciels

Examen – Partie Unix

Philippe MARQUET

Janvier 1998

Les documents de cours et TD sont autorisés.
On rendra deux copies séparées pour la partie Unix et pour la partie langage C. La partie Unix est à rendre sur une *copie de couleur.*

Important Plusieurs commandes Bourne-shell sont à développer. On veillera particulièrement à détailler pour chacune des commandes un algorithme avant de donner le code Bourne-shell lui-même. La notation sera répartie entre ces deux composantes de vos réponses.

1 Vérification de duplication d'exécutables

Les commandes exécutables sont recherchées par le shell dans les répertoires listées dans les variables `$path` ou `$PATH`. Il est courant que plusieurs problèmes surviennent à ce niveau : duplication d'un répertoire parmi ceux listés dans `$PATH`, duplication d'exécutables entre deux répertoires distincts listés dans `$PATH`, etc. Le but des commandes Bourne-shell qui seront développées est d'identifier ces problèmes.

On développera successivement :

- une commande `uniqpath` qui vérifie l'absence de doublons dans les répertoires listés par la variable `$PATH` ;
- une commande `vrfydupexe` qui liste les doublons d'exécutables dans les répertoires de recherches.

Question 1 (Pourquoi les doublons sont-ils gênants)

- Expliquer en quoi la présence de doublons dans les répertoires listés dans `$PATH` est gênante.
- Expliquer en quoi la présence de doublons parmi les exécutables trouvées dans les répertoires listés dans `$PATH` est gênante.

1.1 `uniqpath`

La syntaxe de la commande `uniqpath` est la suivante :

`uniqpath [-quiet] [-path]`

La commande retourne sur la sortie standard une liste de répertoires au format de `$PATH` qui est un sous ensemble des répertoires listés dans `$PATH` dans lequel les répertoires relatifs (y compris `.` et `..`) sont supprimés et tel que aucun des répertoires listés ne soit présent plusieurs fois.

Par défaut, la commande `uniqpath` produit sur la sortie d'erreur la liste des répertoires doublons de `$PATH` ; l'option `-quiet` indique de ne pas produire ce résultat.

Le résultat produit par la commande `uniqpath` est par défaut au format utilisé par la variable `$PATH` ; l'option `-path` indique de produire un résultat au format de la variable `$path`.

Id : xam98.tex,v 1.5 2000/11/28 18 :34 :55 marquet Exp

Note Le format de la variable `$PATH` est construit sur l'exemple suivant :

```
% echo $PATH
/usr/local/tex/bin/:/usr/local/bin:/usr/sbin:/sbin:/etc:/home/west/marquet/bin\
/linux:/home/west/marquet/bin:.:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/u\
sr/X11R6/bin
```

Le format de la variable `$path` est une liste de répertoires :

```
% echo $path
/usr/local/tex/bin/ /usr/local/bin /usr/sbin /sbin /etc /home/west/marquet/bin\
/linux /home/west/marquet/bin . /usr/local/bin /bin /usr/bin /usr/X11R6/bin /u\
sr/X11R6/bin
```

Exemples d'utilisation

```
% uniqpath -path
/usr/X11R6/bin : 2 occurrences dans $PATH
/usr/local/bin : 2 occurrences dans $PATH
/usr/local/tex/bin/ /usr/local/bin /usr/sbin /sbin /etc /home/west/marquet/bin\
/linux /home/west/marquet/bin /bin /usr/bin /usr/X11R6/bin
% setenv $PATH  .:`uniqpath -quiet`
```

Question 2 (Commande `uniqpath`)
Donner l'algorithme puis le code Bourne-shell de la commande `uniqpath`.

1.2 `vrfydupexe`

La commande `vrfydupexe` liste les doublons dans la liste des exécutables trouvés dans les répertoires passés en paramètre.

Exemple d'utilisation

```
% vrfydupexe `uniqpath -path`
/usr/X11R6/bin : 2 occurrences dans $PATH
/usr/local/bin : 2 occurrences dans $PATH
-rwxr-xr-x   3 root     system    139264 May 11  1996 /usr/local/bin/zcat
-rwxr-xr-x   3 bin      bin        32768 Feb  3  1995 /usr/bin/zcat
-rwxr-xr-x   1 marquet  west        2341 Sep  8  1997 /home/west/marquet/bin/l\
ndir
-rwxr-xr-x   1 root     root        5768 Nov  6  1996 /usr/X11R6/bin/lndir
```

Question 3 (Commande `vrfydupexe`)
Donner l'algorithme puis le code Bourne-shell de la commande `vrfydupexe`.

2 Distribution de logiciels

La gestion par un administrateur système d'un parc de machines peut nécessiter la mise à disposition sur plusieurs machines d'un même logiciel. Une solution est de rendre le logiciel disponible via l'exportation depuis une machine serveur d'un système de fichiers sur lequel est installé le logiciel. Une alternative est de dupliquer ce logiciel sur un disque local de chacune des machines clientes.

Question 4 (Pourquoi une commande `dist`)
Discuter brièvement des avantages et inconvénients des deux approches. En particulier, considérer successivement les points de vue de l'administrateur système et de l'utilisateur.

Nous proposons ici de mettre en place un environnement permettant de joindre les avantages des deux solutions. Le logiciel est installé sur une machine dite *machine de référence* par l'administrateur système. L'environnement de *distribution du logiciel* est ensuite configuré pour permettre la distribution automatique du logiciel sur l'ensemble des machines des utilisateurs.

La distribution d'un logiciel sur une machine est caractérisée par

- le nom de la machine cible ;
- une liste de fichiers (ou répertoires) présents sur la machine de référence qui doivent être recopiés sur la machine cible.

Ces informations sont présentes dans un fichier de configuration (`.cfg`). Ces fichiers comportent des lignes de commentaires (débutant par `#`), des lignes de définition de domaine, et des lignes d'action.

Les lignes de définition de domaine permettent de construire des listes de machines cibles sur lesquelles le logiciel devra être distribué. Ces lignes sont au format :

```
<Domain-Name> := <Liste de Domain- ou Machine-Name>
```

La partie droite est une liste de noms de domaines précédemment définis dans le fichier de configuration ou de noms de machines. Les différents termes de la liste sont séparés par des espaces.

Une ligne d'action donne, pour une machine ou un domaine donné, un ensemble de fichiers sources qui devront être distribués sur chacune des machines du domaine. Ces lignes sont au format suivant :

```
<Domain-Name> : <Entrée Source> [-> <Entrée Destinataire>]
```

L'entrée destinataire est par défaut la même que l'entrée fichier source. Dans le cas d'un répertoire, l'ensemble des entrées de ce répertoire sont récursivement traitées.

Exemple de fichier de configuration :

```
% cat ns.cfg
# Fichier de configuration de distribution du logiciel ns
# sur les machines de mon domaine

# Les machines OSF Unix
OSF := luxe bock
# Les machines Linux
LINUX := sol coors
# L'ensemble des machines de mon domaine
TOUT := LINUX OSF palm

# Distribution du logiciel
TOUT : /usr/local/ns
# Distribution des pages de manuel du logiciel
TOUT : /usr/local/man/man1/ns.1
# Distribution du binaire du logiciel, un binaire par type
#  d'architecture
OSF : /usr/local/osf/ns -> /usr/local/bin/ns
LINUX : /usr/local/linux/ns -> /usr/local/bin/ns
palm : /usr/local/linux/ns -> /usr/local/bin/ns
atlas : /usr/local/linux/ns -> /usr/local/bin/ns
```

Question 5 (Commande `listhost`)
Donner l'algorithme puis le code d'une commande Bourne-shell `listhost` qui affiche l'ensemble des machines impliquées dans la distribution d'un logiciel dont le fichier de configuration de distribution est donné en paramètre. Exemple :

```
% listhost ns.cfg
luxe bock sol coors palm atlas
```

Question 6 (Commande `entry`)
Donner l'algorithme puis le code d'une commande Bourne-shell `entry` qui affiche les parties droites des lignes d'action pour une machine et un fichier de configuration donné. Exemple :

```
% entry ns.cfg palm
/usr/local/ns
/usr/local/man/man1/ns.1
/usr/local/linux/ns -> /usr/local/bin/ns
```

La commande Bourne-shell `dist` :

`dist` [`-remove`] [`-savetargets`] *`cfgfile`* [*`domain`*]...

diffuse le logiciel dont le fichier de configuration *`cfgfile`* est passé en paramètre. Par défaut la distribution se fait sur l'ensemble des machines concernées par ce fichier de configuration. La présence sur la ligne de commande d'une liste de domaine ou de machine (paramètre *`domain`*) indique de limiter la distribution à ces domaines/machines.

L'option `-savetargets` réalise une copie des fichiers distants avant mise à jour. Tout fichier `file` mis a jour est préalablement recopié dans `file.OLD`

L'option `-remove` supprime les fichiers supplémentaires. Pour chacun des répertoires mis à jour par la distribution, les entrées sur la machine cible qui n'existent pas dans la distribution de référence sont supprimées.

Exemple :

```
% rsh palm ls /usr/local/ns
/usr/local/ns:
ns.dat          ns.vtk          version
% ls /usr/local/ns/ns.{dat,vtk}
/usr/local/ns/ns.dat
/usr/local/ns/ns.vtk
% ls /usr/local/ns/version
/usr/local/ns/version: No such file or directory.
% dist -savetargets -remove ns.cfg palm
-- Sauvegarde de palm:/usr/local/ns/ns.dat dans palm:/usr/local/ns.dat.OLD
-- Sauvegarde de palm:/usr/local/ns/ns.vtk dans palm:/usr/local/ns.vtk.OLD
-- Suppression de palm:/usr/local/ns/version
-- Copie de /usr/local/ns sur palm:/usr/local/ns
-- Sauvegarde de palm:/usr/local/man/man1/ns.1 dans palm:/usr/local/man/man1/ns.1.OLD
-- Copie de /usr/local/man/man1/ns.1 sur palm:/usr/local/man/man1/ns.1
-- Sauvegarde de palm:/usr/local/bin/ns dans palm:/usr/local/bin/ns.OLD
-- Copie de /usr/local/linux/ns sur palm:/usr/local/bin/ns
```

Question 7 (Commande `dist`)
Donner l'algorithme puis le code de la commande Bourne-shell `dist`.