Licence d'informatique
Module de C/Unix

La commande `disk`

Examen – Partie Unix

Philippe MARQUET

Janvier 2003

Les documents de cours et TD sont autorisés.
On rendra deux copies séparées pour la partie langage C et pour la partie Unix ; la partie Unix sur une *copie de couleur*.
On remarquera que les questions peuvent être traitées indépendamment les unes des autres.
On introduira tout code Bourne-shell par une brève description de son comportement ; on attachera un commentaire à toute portion non triviale de code.

1 Questions de cours

Exercice 1 (`ftp` *vs.* `rcp`)
Les commandes `ftp` et `rcp` permettent de transférer des fichiers entre deux machines. Expliquez les avantages respectifs de ces deux commandes. □

Exercice 2 (`cp` erroné)
Expliquez pourquoi la commande suivante peut produire un tel résultat :

```
% cp *.c~ ~/c/xam &
cp: 'main.c~' and '/users/phm/c/xam/main.c~' are the same file
cp: 'tools.c~' and '/users/phm/c/xam/tools.c~' are the same file
```

□

2 Transfert par `disk`

Afin de gérer les transferts via disquettes entre la salle de TP et son ordinateur personnel, un étudiant de licence décide de développer un utilitaire : la commande `disk`.

2.1 Une session d'utilisation de `disk`

Le principe d'utilisation de cette commande `disk` est le suivant.

Lors de son travail, l'étudiant peut à tout moment décider d'enregistrer des fichiers ou répertoires qui auront à être transférés en indiquant simplement leur nom en paramètre à la commande `disk` :

```
% disk file1 file2
```

Il peut par la suite effectuer la copie de l'ensemble des fichiers enregistrés jusqu'alors vers la zone de transit. C'est le contenu de ces fichiers à cet instant qui sera ensuite transféré :

```
% disk --cp
```

En fin de session, il demande le transfert sur disquette de l'ensemble des fichiers de la zone de transit :

Id: xam03.tex,v 1.6 2003/02/21 10:56:34 marquet Exp

```
% disk --tar
```

il peut, pour minimiser les problèmes dus au peu de fiabilité des disquettes, demander à réaliser un second jeu de disquettes :

```
% disk --tar
```

à nouveau.

Pour nettoyer et permettre un nouvelle session, il exécute un

```
% disk --rm
```

qui purge la zone de transit.

2.2 Mémoriser les informations

Entre les différentes exécutions de `disk`, deux types d'informations doivent être conservées :
- l'ensemble des noms des fichiers qui devront être copiés vers la zone de transit ;
- le contenu de la zone de transit lui-même.

Pour cela, on utilisera un fichier (par exemple `$HOME/.disk`) et un répertoire (par exemple `$HOME/disk`).

2.3 La commande **`disk`**

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

```
disk [--ll] | [--cp] | [--tar] | [--rm] | filenames...
```

Sans options, la commande enregistre les fichiers `filenames` comme devant être copiés vers la zone de transit. Ce peut être des répertoires. Ce peut être des noms relatifs ou des noms absolus.

La signification des options est la suivante :

`--ll` liste les fichiers qu'il reste à copier dans la zone de transit et les fichiers de la zone de transit elle-même.

`--cp` copie vers la zone de transit l'ensemble des fichiers enregistrés.

`--tar` transfère sur disquette le contenu de la zone de transit. Réalise préalablement une copie des éventuels fichiers encore en attente de copie vers la zone de transit.

`--rm` assure un nettoyage de la zone de transit.

2.4 Implantation

Les noms de fichiers enregistrés en attente de copie vers la zone de transit sont des noms de fichiers absolus qui ne comportent aucune composante de nom **`.`** (point) ni de nom **`..`** (point point). Nous qualifierons de tels noms de noms *absolus purs*. Pour obtenir de tels noms de fichiers, on implante les deux fonctions Bourne-shell `concat` et `normalize` :

`concat` ***`name1`*** **`name2`** retourne le nom absolu pur équivalent à *`name1`*`/`*`name2`*. Le paramètre *`name1`* est un nom absolu pur. Le paramètre *`name2`* est un nom simple (ne comportant pas de `/`).

`normalize` ***`name`*** retourne le nom absolu pur équivalent à *`name`* qui peut être un nom de fichier absolu ou relatif quelconque. Si *`name`* est relatif, il est compris comme rélatif au répertoire courant.

Exercice 3 (`concat`)
Donnez le code Bourne-shell commenté de la fonction `concat`. □

Exercice 4 (`normalize`)
Donnez le code Bourne-shell commenté de la fonction `normalize`. □

Exercice 5 (`disk filenames...`)
Donnez le code Bourne-shell d'une fonction `disk_rec` qui assure l'enregistrement des fichiers dont les noms lui sont passés en paramètre.

Ces noms peuvent être quelconques mais doivent correspondre à des fichiers existants (sinon un message d'erreur est produit et ces noms ignorés). Les noms enregistrés seront des noms absolus purs.

Si on demande l'enregistrement d'un fichier dont une racine est déjà enregistrée, un message d'avertissement est produit et le nom est ignoré.

Si on demande l'enregistrement d'un fichier qui est une racine d'un fichier déjà enregistré, il faudrait, après avoir produit un message d'avertissement, supprimer ce dernier de la liste des fichiers enregistrés. □

Exercice 6 (`disk --cp`)
Donnez le code Bourne-shell d'une fonction `disk_cp` qui assure la copie dans la zone de transit des fichiers préalablement enregistrés.

Si cette exécution demande à recopier un fichier qui existe déjà dans la zone de transit, un message d'avertissement est produit et la copie n'est pas réalisée. □

La disquette est accessible dans le système de fichiers depuis `/dev/fd0`. La commande `tar` peut écrire directement sur la disquette en spécifiant ce nom comme fichier archive. On spécifie aussi la taille du support et l'option `-M` (multi-volume) pour permettre la création sur plusieurs disquettes d'archives de taille supérieure à la taille du support. On utilise donc une commande comme

```
% tar cMF /dev/fd0 -L1440 /users/phm/take-us-with-you
```

Exercice 7 (`disk --tar`)
Donnez le code Bourne-shell d'une fonction `disk_tar` qui assure la copie du contenu de la zone de transit sur un jeu de disquettes. □

Exercice 8 (Retour à la maison)
Expliquez ce que doit faire l'étudiant une fois chez lui pour récupérer les fichiers qu'il a transférés. □

2.5 Outils

Vous pouvez utiliser les fonctions suivantes :

```
# Qui suis-je ?
Commande=`basename $0`
# Utilitaires
echo2 () { echo "$@" >&2 ; }
fatal () { echo2 "[$Commande] Fatal: $@" ; exit 1 ; }
error () { echo2 "[$Commande] Erreur: $@" ; }
warn ()  { echo2 "[$Commande] Warning: $@" ; }
usage ()
{
  echo2 "$Commande -- Gestion de transfert de fichiers par disquette."
  echo2 "Usage: $Commande filenames...|--ll|--cp|--tar|--rm"
  exit 1
}
```


Fin du sujet