Licence d'informatique
Module de C/Unix

Les commandes `rgrep` et `rmp3`

Examen – Partie Unix

Philippe MARQUET

Septembre 2002

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 (Renommer les `.sh`)
Expliquez pourquoi une commande `rename` qui voudrait renommer l'ensemble de fichiers `.sh` en `.bash` ne peut être appelée depuis le shell par la séquence :

```
% rename *.sh *.bash
```

□

Exercice 2 (Echo)
Soit l'extrait d'une session `csh` suivant :

```
% echo $PATH
/usr/local/bin:/bin:/usr/sbin:/sbin:/users/phm/bin/
% echo $PWD
/users/phm/ens/uu/sc/
% echo *
echo
% ./echo *
total 1
-rwxr-xr-x    1 phm user      34 sep  6 11:46 echo
```

Donnez un contenu possible pour les fichiers du répertoire `/users/phm/ens/uu/sc/`. □

2 Recherche récursive

Les dernières versions de `grep` comportent une option autorisant la recherche récursive d'un motif dans l'ensemble des fichiers contenus sous un répertoire donné.

Nous allons développer sous la forme d'un script Bourne-shell la commande `rgrep` qui ajoute cette fonctionnalité à une version de base de `grep`.

On considère la syntaxe basique de `grep` suivante :

```
grep [options]... regexpr [file]...
```

et la syntaxe de `rgrep` :

Id: xam02-sep.tex,v 1.7 2002/09/12 07:05:51 marquet Exp

```
rgrep [-r [--incl regexpr][--excl regexpr]] [options]... regexpr [file]...
```

La signification des options de `rgrep` est la suivante :

`-r` pour chacune des valeurs `file` qui désigne un répertoire, la commande va traiter l'ensemble des fichiers du répertoire, et ce récursivement.

`--incl` ***`regexpr`*** lors d'un traitement récursif, seuls les fichiers dont le nom correspond avec l'expression régulière `regexpr` doivent être traités.

`--excl` ***`regexpr`*** lors d'un traitement récursif, seuls les fichiers dont le nom ne correspond pas avec l'expression régulière `regexpr` doivent être traités.

Exercice 3 (Utilisation des options `--incl` et `--excl`)
Question 3.1 Donnez la commande `rgrep` cherchant la chaîne de caractère `main` dans les sources C du répertoire courant. □

Question 3.2 Donnez la commande `rgrep` cherchant la chaîne de caractère `main` dans les sources C du répertoire courant et de tous ses sous-répertoires. □

Exercice 4 (Traiter un fichier ?)
En supposant les valeurs des paramètres de la ligne de commande mémorisées dans des variables, il s'agit de déterminer si le nom d'un fichier doit être traité par la commande `rgrep`.

Question 4.1 Explicitez un rangement des valeurs des paramètres de la ligne de commande dans des variables. Il n'est pas demandé de donner le code Bourne-shell qui lit la ligne de commande et initialise ces variables. □

Question 4.2 Donnez le code Bourne-shell commenté d'une fonction `processfile` qui retourne, dans le cas d'un traitement récursif, un succès si et seulement le nom de fichier passé en paramètre doit être traité par la commande. □

Exercice 5 (`rgrep`)
Question 5.1 Donnez le code Bourne-shell commenté d'une commande `rgrep` à l'exclusion du code exploitant la ligne de commande pour initialiser les variables définies à la question 4.1. □

3 Rangement de mp3

Il est courant de se retrouver avec un répertoire contenant de nombreux fichiers mp3 dont les noms ne sont pas/plus significatifs. La commande `rmp3` que vous allez développer va ranger ces fichiers.

Un fichier mp3 peut contenir, en plus des données audio, un certain nombre d'identificateurs dits ID3 : nom de l'artiste, de l'album, du morceau, etc. La commande `mp3info` permet de visualiser ces identificateurs (et d'autres informations concernnat les données audio) :

```
% mp3info PoZuStSwChBa01.mp3
File: PoZuStSwChBa.mp3
Title:   Barouf
Track: 1
Artist:  Michel Portal
Album:   Dockings
Year:  1998
Comment:
Genre: Jazz [8]
Media Type:  MPEG 1.0 Layer III
Audio:       320 KB/s, 44KHz (joint stereo)
```

```
Emphasis:     none
CRC:          No
Copyright:    Yes
Original:     Yes
Padding:      Yes
Length:       6:01
```

Certaines données peuvent être manquantes (par exemple l'identificateur `Comment` du fichier précédent). L'ensemble de ces données peut être absent ; une première ligne du résultat de `mp3info` le signale :

```
% mp3info ScFgHu07.mp3
ScFgHu07.mp3 does not have an ID3 1.x tag.
File: ScFgHu07.mp3
Media Type:  MPEG 1.0 Layer III
Audio:       320 KB/s, 44KHz (joint stereo)
Emphasis:    none
CRC:         No
Copyright:   Yes
Original:    Yes
Padding:     Yes
Length:      5:41
```

La commande `rmp3` va par exemple classer le fichier `PoZuStSwChBa01.mp3` sous le nom `Michel_Portal/Dockings/01_Barouf.mp3`.

Exercice 6 (Lecture des identificateurs)
Il s'agit de développer une fonction `getid` qui affiche la valeur de l'identificateur passé en premier paramètre pour le fichier dont le nom est passé en second paramètre. En cas d'absence de l'identificateur, la chaîne `Various` accolée au nom de l'identificateur est retournée. Dans le résultat retourné, les caractères espace ␣, guillemet `"`, apostrophe `'`, et barre oblique `/` sont remplacés par un trait bas _. Exemples :

```
% getid Artist PoZuStSwChBa01.mp3
Michel_Portal
% getid Comment PoZuStSwChBa01.mp3
Various_Comment
% getid Artist ScFgHu07.mp3
Various_Artist
```

Question 6.1 Donnez le code Bourne-shell commenté d'une fonction `getid`. □

Exercice 7 (Renommage)
À partir des identificateurs contenus dans un fichiers mp3, il est maintenant facile de créer un nom du type

<*Artist*>/<*Album*>/<*Track*>_<*Title*>.mp3

Ce nom est le « nom parlant » du fichier mp3.

Cependant, il est possible que deux fichiers mp3 différents génèrent tous deux un nom parlant identique. Dans ce cas, on conservera toutes les versions en utilisant le nommage suivant identifiant les différentes versions d'un même morceau :

<*Artist*>/<*Album*>/<*Track*>_<*Title*>.v<*i*>.mp3

Ce nom est le « nom parlant unique » du fichier mp3.

Question 7.1 Donnez le code Bourne-shell commenté d'une fonction `newname` qui prend en paramètre le nom d'un fichier mp3 est retourne son nom parlant unique.

Le résultat de cette fonction sera fonction de l'éventuelle présence d'un fichier portant déjà ce nom parlant ; dans ce cas, et si les contenus des fichiers diffèrent, il pourra aussi être nécessaire de renommer un tel fichier existant.

Fin du sujet