Département Informatique 13 Janvier 2006

IUT Bordeaux 1

ASR2 – Utilisation des systèmes d'exploitation

Devoir surveillé

Durée : 1h30
Sans documents

Exercice 1

Un programmeur écrit un script qui manipule un fichier `agenda.txt` contenant, sur chaque ligne, un nom, un prénom et un numéro de téléphone séparés par le caractère deux-points.

Extrait du fichier « `agenda.txt` »

```
marcel:dupont:01234567
gerard:durand:23456789
...
```

Dans son script, le fragment de code ci-contre lui donne quelques soucis, en effet le numéro de téléphone de « marcel dupont » semble ne pas vouloir changer dans le fichier `agenda.txt`.

Extrait du shell script

```
N=dupont
P=marcel
T=9988776655
..
# remplacement numéro de téléphone
sed -e 's/^$P:$N:.*$/$P:$N:$T/' \
      agenda.txt > temporaire
cp temporaire agenda.txt
```

Question 1.1 Expliquez pourquoi, quand on fait une substitution par sed dans un fichier, on écrit

```
sed -e s/X/Y/  f > g ; cp g f
```

plutôt que

```
sed -e s/X/Y/ f > f
```

Question 1.2 Pour la mise au point des scripts, une bonne pratique est de faire un `echo` des lignes de commandes pour lesquelles on a des doutes. Le programmeur insère donc la commande

```
echo "sed -e 's/^$P:$N:.*$/$P:$N:$T/' agenda.txt > temporaire"
```

qui semble montrer que la commande `sed` du script est correcte. Qu'affiche-t'elle exactement?

Question 1.3 Dîtes et expliquez ce qu'afficherait

```
echo 'sed -e \'s/^$P:$N:.*$/$P:$N:$T/\' agenda.txt > temporaire'
```

Question 1.4 Quel était le problème avec la ligne de commande `sed` ? Proposez une correction.

Question 1.5 Écrire une fonction « `extractTelParis` » qui affiche, à partir du fichier `agenda.txt` (ou d'un autre de même structure), une liste des parisiens (dont le numéro de téléphone commence par `01`), classée dans l'ordre alphabétique des noms.

$ extracTelParis agenda.txt

```
dupont marcel 01234567
gerard:durand:23456789
```

Avant le traitement on s'assurera que le fichier (dont le nom est donné en paramètre) existe.

Exercice 2

Rappel: la commande `useradd` permet de créer un compte d'utilisateur, exemple

```
useradd -m -g users -c "Dr. Scott"  scott
```

Rappel des options :
-m (make) crée le répertoire de travail
-g groupe
-c nom complet

Question 2.1
-Quelle est la différence avec la commande "adduser" ?
- dans quel(s) fichier(s) les comptes sont-ils mémorisés ?

Question 2.2 Écrivez un script « `enregistrer` » permettant, en quelques lignes, de créer les comptes d'utilisateurs décrits dans un fichier texte comme ci-contre, (le second champ est un mot de passe en clair, qui servira à la question suivante). Le nom du fichier sera transmis en paramètre. Exemple d'utilisation : `enregistrer NOUVEAUX`

Question 2.3 Une fois les comptes créés, il restera à les « ouvrir » en attribuant des mots de passe. On utilise pour cela la commande "chpasswd", dont voici un extrait de la documentation :

Fichier « NOUVEAUX »

```
brad g56h54 Brad Majors
janet k76c78 Janet Weiss
frank a23b43 Frank N. Furter
rocky l87g69 Rocky Horror
```

man chpasswd

CHPASSWD(8)

NOM

chpasswd - met à jour des mots de passe par lot

SYNOPSIS

chpasswd [options]

DESCRIPTION

Chpasswd lit une liste de paires de noms d'utilisateurs et de mots de passe depuis l'entrée standard et utilise ces informations pour mettre à jour un groupe d'utilisateurs existants. Chaque ligne est au format suivant :

nom_utilisateur:mot_de_passe

L'utilisateur doit exister. Par défaut, le mot de passe doit être fourni en clair. (...)

Cette commande est destinée aux gros systèmes pour lesquels un nombre importants de comptes sont créés en une seule fois.

OPTIONS

Les options disponibles pour la commande chpasswd sont :

.....

Écrire un script qui prend comme paramètre le nom d'un tel fichier des comptes, et produit en sortie les paires voulues. Exemple d'exécution du script :

```
$ fabriquer-paires NOUVEAUX
brad:g56h54
janet:k76c78
frank:a23b43
rocky:l87g69
```

Question 2.4 Comment l'articuler avec `chpasswd` pour attribuer les mots de passe ?

Exercice 3

Question 3.1 Pour les interpréteurs de commandes *shell* on distingue les commandes internes et les commandes externes. Quelle est la différence ? Donnez des exemples.

Question 3.2 Comment obtenir de l'aide pour une commande interne ? pour une commande externe ?

Question 3.3 Comment avoir la liste des commandes internes ? Combien y a-t-il à votre avis ? Mêmes questions pour les commandes externes.

Exercice 4

Le script suivant (`mystere.sh`) semble comporter quelques erreurs

```
1. !#C:\bin\bash
2. // substitution deux-points par espace
3. $DIRS  =  { echo $PATH || sed ':' ' '}
4.   for $D in $DIRS {
5.     for $F in $D/*$1* {
6.       if (-x $F ) {
7.          echo ($F)
8.       }
9.     }
10. }
```

Question 4.1 Réécrivez-le correctement.

Question 4.2 Expliquez ce que fait la commande « `mystere.sh mac` » ?

(on suppose que la variable `PATH` contient « `/usr/bin:/usr/local/bin/opt/bin:.` »)