Travaux dirigés de Perl n°2/6

Tables de hachage, regexps

—Université Gustave Eiffel—

Le but de ce TD est de continuer l'apprentissage du langage *Perl* par quelques exemples utilisant les tables de hachage et les expressions régulières.
Toutes les fonctions écrites devront être testées.

▸ Exercice 1.

Écrivez un programme qui prend sur sa ligne de commande des noms de mois (janvier ... decembre) et qui affiche leur nombre de jours (28, 30 ou 31) à l'écran. Gérez le cas de mois inexistants.

```
$ ./exo1.pl fevrier velo
fevrier: 28
velo: inconnu
```

Comment supprimer le mois de `fevrier` *de la table de hachage ?*

▸ Exercice 2. *(Table de hachage et fichiers)*

Les lignes du fichier système `passwd` *sont de format* `login:passwd:uid:gid:info:home:shell` *Chaque ligne a toujours exactement 6 signes deux-points. Écrivez un programme qui créé à partir du fichier* `/home/ens/lhullier/ens/passwd` *ou* `https://formation-perl.fr/t/passwd` *une table de hachage associant l'uid au login (le login sera la clef et l'uid sera la valeur). Utilisez la fonction* `split` *pour découper le contenu de chaque ligne.*
Voici les instructions nécessaires à la lecture ligne à ligne du fichier :

```
open( my $fd, '<' ,'nomDuFichier' ) or die( "open: $!" );
while( defined( my $ligne = <$fd> ) ) {
   chomp $ligne;  # Supprime la fin de ligne
   # Maintenant  $ligne  contient chacune des lignes du fichier
}
close( $fd );
```

Effectuez ensuite l'affichage du contenu de la table de hachage selon un ordre quelconque.
Comment trier selon l'ordre alphabétique des logins ?
Comment trier selon l'ordre croissant des uid ?
Question subsidiaire : comment affiner ce tri par ordre alphabétique des logins ?

▸ Exercice 3. *(Expressions régulières)*

Le but de cet exercice est la manipulation des expressions régulières (ne pas utiliser fonction `split` *ici). Il vous est demandé d'écrire un programme manipulant le même fichier* `passwd` *et affichant les informations qui vous seront demandées.*

Ce programme comportera uniquement les instructions nécessaires à la lecture du fichier ainsi que l'expression régulière demandée et un simple appel à la fonction `print`*. Une fois que vous avez répondu à une question, vous pouvez mettre en commentaire les deux lignes de code la concernant et passer à la suivante.*

Il vous est demandé les opérations suivantes :

1. *Affichez la ligne correspondant à l'utilisateur* `jc` *(faire simple).*
2. *Affichez les lignes correspondant aux utilisateurs qui n'ont pas* `bash` *comme shell (commencez par chercher ceux qui ont* `bash` *comme shell, puis niez).*
3. *Affichez les lignes en remplaçant* `/home/` *par* `/mnt/home/`
4. *Affichez les lignes en supprimant la séquence de mot de passe.*
5. *Affichez les lignes en échangeant login et passwd.*

Questions subsidiaires :

6. *Affichez les lignes en échangeant uid et gid. Que fait l'expression suivante ?* `((?:.*?:){2})`
7. *Affichez uniquement le champ* `gid` *de tous les utilisateurs (faites une extraction).*
8. *Affichez les lignes en multipliant le* `gid` *par 2 (option* `e`*).*

▸ Exercice 4. *(Analyse de journal Apache)*

Nous allons écrire un programme analysant les fichiers de logs Apache (serveur web). Vous utiliserez le fichier `/home/ens/lhullier/ens/access_log` *ou* `https://formation-perl.fr/t/access_log` *Ce fichier de log à le format suivant :*

```
10.0.0.1 - - [05/Apr/2042:13:26:17] "GET /index.html HTTP/1.1" 200 815 "-" "Mozilla"
IP - - [date] "XXX /url/demandée HTTP/n.m" statut volume "referer" "navigateur"
```

où `XXX` *peut valoir* `GET`, `HEAD`, `POST` *...*

Le but du programme est de répondre aux questions suivantes. Pour cela, vous aurez à écrire une regexp permettant d'extraire de chaque ligne de log les informations nécessaires.

1. *Combien de requêtes y a-t-il eu au total ?*
2. *Combien d'erreurs ? (statut autre que 200)*
3. *Quel est le nombre total d'octets transférés ? (champ volume)*
4. *Affichez toutes les urls dans l'ordre décroissant du nombre d'accès.*
5. *Quel sont les 10 adresses IP ayant accédé le plus de fois au serveur ?*
6. *Indiquez pour chacune de ces 10 IP le volume transféré.*

Pour vous aider

Quelles sont les données présentes dans chaque ligne qu'il vous faut récupérer ?
Quelles sont les structures de données nécessaires au traitement de ces questions ?
Le programme comportera 3 phases successives :

- *une phase de déclaration des variables nécessaires,*
- *une phase de lecture silencieuse des lignes de données avec extractions des variables utiles et calcul progressif des résultats,*
  NB : que pensez-vous de la regexp suivante : `m/^(.*?) .*?".*? (.*?) .*?" (.*?) (.*?) /`
- *après la fin de fichier, une phase d'affichage des résultats.*

Pour contrôle

```
nbReq=100
nbErr=3
volumeTotal=1394189

/icones/arbre.png : 6
/robots.txt : 4
/icones/courriel.png : 4

256.84.59.51 : access=13 volume=131230
48.93.60.21 : access=11 volume=225963
77.69.182.250 : access=7 volume=59429
```