TP Linux Script 5 : Script – Exploiter un fichier

Savoir faire : Utiliser les variables et les structures de contrôle dans un shell script.

Poste de travail : Linux

Pré-requis : Algorithmique, cours Linux variables, tests et itérations, TP shell précédents : cut, pipe, redirection, arguments.

Type 1 SN1 3h

1 Cahier des charges :

Avec le SFTP de BitVise, copier le fichier « lease_dhcp » dans un sous-dossier TP-EXT-SH2 du serveur Linux.

Ce fichier LOG provient d'un serveur DHCP chargé de distribuer les adresses IP aux stations accédant au réseau.

Chaque station cliente reçoit un bail (angl. *lease*) indiquant les caractéristiques du bail, notamment la durée de validité de l'adresse IP :

```
lease 192.168.0.151 {
    starts 1 2010/10/11 20:50:26;
    ends 1 2010/10/11 20:52:26;
    binding state free;
    hardware ethernet 00:1a:73:83:6a:b5;
    uid "\001\000\032s\203j\265";
    client-hostname "sweet-0ba835b4c";
}
```

Cet affichage en paragraphes ne nous convient pas. On souhaite afficher les informations de ce fichier sous forme de tableau :

```
2010/10/11 15:23:02 16:23:02 192.168.0.116 78:e4:00:a9:3d:83
2010/10/11 15:29:13 16:29:13 192.168.0.106 00:23:4e:56:b6:7e
2010/10/11 15:30:15 16:30:15 192.168.0.144 00:19:d2:4b:a6:df
2010/10/11 15:30:30 16:30:30 192.168.0.116 78:e4:00:a9:3d:83
2010/10/11 15:30:44 16:30:44 192.168.0.131 00:1a:73:81:f1:9d
2010/10/11 15:31:45 16:31:45 192.168.0.108 00:21:63:a3:94:54
2010/10/11 15:35:44 16:35:44 192.168.0.117 00:60:76:03:d2:24
```

Pour cela vous allez créer un script qui exploite le fichier ; mais avant, effectuez les tâches suivantes :

2 Saisissez et testez le script suivant (tpl5-script2):

```
for txt in $(cat ./lease_dhcp)
do
    echo "lu : $txt"

done
```

Ici, on utilise le système \$(commande) pour alimenter la boucle FOR avec le résultat de la commande « cat ».

Observation : le fichier « lease_dhcp » est-il affiché mot par mot ou ligne par ligne ?

3 Saisissez et testez le script suivant (tpl5-script3):

```
while read txt
do
    echo "lu : $txt"

done < ./lease_dhcp
```

La notation peut vous surprendre ! On utilise la redirection avec le signe <

Vous connaissez déjà le signe > qui redirige la sortie d'une commande (son affichage) vers un fichier.

Ici, on redirige cette fois l'entrée standard de la commande read : la lecture ne se fait plus au clavier mais en utilisant le contenu du fichier « lease_dhcp ». On fait cela en ajoutant < ./lease_dhcp à la fin. Comme le « read » fait partie du « while », la redirection se fait à la fin de la commande « while », après le « done ».

Conséquence : A chaque passage dans la boucle, « read » lit une ligne du fichier.

Observation : le fichier « lease_dhcp » est-il affiché mot par mot ou ligne par ligne ?

CONCLUSION : la commande CAT provoque une lecture mot par mot, et la commande READ lit jusqu'à la fin de la ligne, ou jusqu'à la touche <ENTREE>.

4 Créez et testez le script suivant (tpl5-script4):

```
while read ligne                # Pour chaque ligne du fichier
do
    mot2=$(echo $ligne | cut -d' ' -f2)
    [ "$mot2" = "ethernet" ] && echo $ligne | cut -d' ' -f3

done < ./lease_dhcp
```

Analysez le résultat obtenu.

Ajoutez des commentaires pour chaque ligne du script. Un commentaire commence par le signe #.

Le premier commentaire apparait déjà sur la première ligne.

Faites contrôler les commentaires par le professeur.

5 Découper une ligne de texte avec la commande « set »

On peut simplifier « tpl5-script4 » de la question précédente en utilisant la commande « set » à la place de la commande « cut ».

La commande « set » permet de « transférer » le contenu d'une variable dans les variables « arguments ». Tester les commandes suivantes dans une fenêtre de terminal:

```
var="BTS SN de Vauvenargues" // $var contiendra la phrase
set $var //on transfère chaque mot de $var dans $1, $2 ...

echo $1
echo $2
echo $3
echo $4
```

Constatez de quelle façon les mots de la phrase sont répartis.

Simplifions notre script du paragraphe précédent. Le nouveau script (tpl5-script5) commencera ainsi :

```
while read ligne
do
    set $ligne
    #
    # ici votre travail
    #
done < ./lease_dhcp
```

Complétez ce script pour qu'il affiche le même résultat que tpl5-script4.

NB : La commande SET pose un problème lorsque \$ligne est vide (donc quand le fichier *lease_dhcp* contient une ligne vide). Il faudra donc tester : si \$ligne est vide, on retourne au WHILE avec la commande *continue*.

6 Compléter le script (tpl5-script6) :

Copier et modifiez le script précédent (dans un nouveau fichier) pour obtenir le résultat de la forme :

Adresse IP	Adresse MAC
192.168.0.119	78:e4:00:f5:cb:62;
192.168.0.131	00:1a:73:81:f1:9d;
192.168.0.100	00:1e:4c:51:0a:2c;
192.168.0.211	00:26:b0:03:13:d1;
192.168.0.137	00:16:44:61:cd:d9;
192.168.0.146	00:25:d3:bd:76:3e;
192.168.0.118	00:0e:35:a0:ce:cc;
192.168.0.148	00:22:fb:a9:85:d8;

7 Finir le script (tpl5-script7)

Copier et modifier le script (dans un nouveau fichier) pour avoir enfin la solution du cahier des charges :

Date	Durée	Adresse IP	Adresse MAC
2010/10/11	15:23:02 16:23:02	192.168.0.116	78:e4:00:a9:3d:83
2010/10/11	15:29:13 16:29:13	192.168.0.106	00:23:4e:56:b6:7e
2010/10/11	15:30:15 16:30:15	192.168.0.144	00:19:d2:4b:a6:df
2010/10/11	15:30:30 16:30:30	192.168.0.116	78:e4:00:a9:3d:83
2010/10/11	15:30:44 16:30:44	192.168.0.131	00:1a:73:81:f1:9d
2010/10/11	15:31:45 16:31:45	192.168.0.108	00:21:63:a3:94:54
2010/10/11	15:35:44 16:35:44	192.168.0.117	0c:60:76:03:d2:24