

TP : Netadmin

Réalisé par : Chams TMAR (GL3-2)

Dans ce TP, on voulait écrire une application shell pour gérer le réseau. Ainsi, on va afficher un menu à l'utilisateur, présentant différentes options.

En fonction du choix de l'utilisateur, le script exécute différentes actions à l'aide d'une structure de contrôle `case` :

- Si le choix est **1**, on liste toutes les interfaces réseau avec leurs adresses MAC.
- Si le choix est **2**, on liste toutes les interfaces réseau avec leurs adresses IP.
- Si le choix est **3**, on affiche le contenu du fichier `/etc/services`, qui contient des informations sur les ports et les services correspondants.
- Si le choix est **4**, on permet à l'utilisateur de changer l'adresse IP d'une interface réseau spécifiée.
- Si le choix est **5**, on désactive une interface réseau spécifiée.
- Si le choix est **6**, on active une interface réseau spécifiée.
- Si le choix est **7**, on teste l'accessibilité d'une adresse IP à l'aide de la commande ping.
- Si le choix est **8**, on trouve l'adresse IP associée à une URL donnée à l'aide de la commande nslookup.
- Si le choix est **9**, on initie une session FTP pour transférer un fichier spécifié vers une adresse IP de destination.
- Si le choix est **10**, on permet à l'utilisateur de se connecter à une machine distante à l'aide de SSH.
- Si le choix est **11**, le script quitte en utilisant la commande `exit`.

Voici le script expliqué :

```
echo "Administration des réseaux"
```

```
while true
do
```

```
echo ""
```

```
echo "1- Afficher les interfaces avec adresse mac"
```

```
echo "2- Afficher les interfaces avec adresses ip"
```

```

echo "3- Afficher les ports et les services"
echo "4: Changer l'adresse IP"
echo "5: Désactiver un port"
echo "6: Activer un port"
echo "7: Troubleshooting, tester une adresse IP (ping)"
echo "8: Trouver l'adresse IP d'une URL (nslookup)"
echo "9: Transférer un fichier (FTP)"
echo "10: Se connecter sur une machine distante"
echo "11: Quitter"

echo ""
echo "Tapez votre choix"
read choix

case $choix in
1) interfaces=$(ifconfig -a \ #on récupère la liste des
interfaces réseau
| grep "^[[:alpha:]]" \ #on filtre les lignes
alphabétiques
| awk '{print $1}') #on extrait le premier champ de
chaque ligne qui consiste en l'interface_name

for interface in $interfaces #pour chaque interface
do

addr=$(ifconfig $interface \ #on affiche ses détails
| grep "HWaddr" \ #on extrait la ligne contenant
l'adresse MAC (ligne contenant la chaîne HWaddr -> HardWare
address)
| sed 's/^. *HWaddr //' ) #on supprime la première
partie de la ligne pour ne laisser que l'adresse physique de
l'interface

if [[ ! -z "$addr" ]] #si l'adresse n'est pas vide
then
addr="HWaddr $addr" #on l'assigne à une variable
fi

```

```
    echo -e "$interface\t$addr"  #on affiche le contenu de la
variable
```

```
done
```

```
;;
```

```
2)  interfaces=$(ifconfig -a \  #on récupère la liste des
interfaces réseau
```

```
    | grep "^[[:alpha:]]" \  #on filtre les lignes
alphabétiques
```

```
    | awk '{print $1}'  #on extrait le premier champ
de chaque ligne qui consiste en l'interface_name
```

```
for interface in $interfaces  #pour chaque interface
do
```

```
    addr=$(ifconfig $interface \  #on affiche ses détails
    | grep "net addr:" \  #on extrait la ligne
contenant l'adresse IP (ligne contenant la chaîne net addr ->
net/IP address)
```

```
    | sed 's/^[ \t]*inet //' \  #on supprime les
espaces au début ainsi que inet
```

```
    | sed 's/ .*//' )  #on supprime le reste qui vient
après l'adresse IP
```

```
if [[ ! -z "$addr" ]]  #si l'adresse n'est pas vide
then
```

```
    addr="inet $addr"  #on l'assigne à une variable
fi
```

```
    echo -e "$interface\t$addr"  #on affiche le contenu de la
variable
```

```
done
```

```
;;
```

```
3)  cat /etc/services  #ce répertoire contient tous les ports
et les services
```

```
;;
```

```

4)  echo "Entrez la nouvelle adresse IP:"
    read new_ip
    echo "Entrez le nom de l'interface:"
    read interface_name
    ifconfig $interface_name $new_ip #commande pour changer
l'adresse IP de l'interface indiquée
    ;;

5)  echo "Entrez le nom de l'interface à désactiver:"
    read interface_name
    ifconfig $interface_name down #commande pour désactiver
l'interface indiquée
    ;;

6)  echo "Entrez le nom de l'interface à activer:"
    read interface_name
    ifconfig $interface_name up #commande pour activer
l'interface indiquée
    ;;

7)  echo "Entrez l'adresse IP à tester (ping):"
    read test_ip
    ping -c 4 $test_ip #commande pour tester l'adresse IPv4
    ;;

8)  echo "Entrez l'URL pour trouver son adresse IP :"
    read url
    nslookup $url #commande pour chercher l'adresse IP de
l'URL saisi
    ;;

9)  echo "Entrez le nom du fichier à transférer (FTP):"
    read file_name

    echo "Entrez l'adresse IP du destinataire:"
    read destination_ip

    ftp -n $destination_ip <<END_SCRIPT #lancer une session
FTP en utilisant l'adresse IP fournie. L'option -n désactive
la connexion automatique, et <<END_SCRIPT commence un bloc de
code qui sera transmis à la commande FTP.

```

```

    quote USER anonymous #envoyer une commande au serveur
FTP en utilisant la commande quote. Ici, on spécifie le nom
d'utilisateur FTP comme "anonymous" (pratique courante en FTP)

    quote PASS guest@ #on fournit le mot de passe guest@ pour
l'utilisateur

    put $file_name #on télécharge le fichier spécifié sur le
serveur FTP par put

    quit #mettre fin à la session IP

END_SCRIPT #fin du bloc de code à transmettre à la
commande FTP

;;
10) echo "Entrez l'adresse IP de la machine distante:"
read remote_ip

echo "Entrez le nom d'utilisateur pour la connexion:"
read remote_user

ssh $remote_user@$remote_ip #commande pour se connecter
à une machine distante avec le username et l'adresse IP
spécifiés

;;
11) break

;;
esac

done

```