TP : Gestion des Modules

Réalisé par : Chams TMAR (GL3-2)

Dans ce TP, on voulait écrire une application shell pour gérer les modules du noyau Linux (Linux Kernel). 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**, le script liste les modules actifs en utilisant la commande `lsmod`. Il stocke ensuite les noms des modules dans un fichier modules.txt, extrait les descriptions de chaque module avec `modinfo`, et les stocke dans un fichier descriptions.txt. Enfin, il utilise la commande `pr` pour afficher les noms et descriptions des modules côte à côte.
- Si le choix est **2**, le script demande à l'utilisateur de fournir le nom d'un module, puis utilise `sudo modprobe -r` pour décharger (désactiver) ce module.
- Si le choix est **3**, le script demande à l'utilisateur de fournir le nom d'un module, puis utilise `sudo modprobe` pour charger (activer) ce module.
- Si le choix est **4**, le script liste les modules supprimés en comparant les modules actifs (`lsmod`) avec tous les modules disponibles dans le répertoire `/lib/modules/$(uname -r)`. Les modules qui ne sont pas actifs sont affichés.
  Si le choix est **5**, le script quitte en utilisant la commande `exit`.

Voici le script expliqué :

```
echo "Application"

while true
do

echo ""
echo "1: Lister les modules actifs"
echo "2: Désactiver un module"
echo "3: Activer un module"
```

```
echo "4: Lister les modules supprimés"
echo "5: Quitter"

echo ""
echo "Tapez votre choix"
read choix

case $choix in
1)    lsmod \   #lister les modules actifs
      | cut -f 1 -d ' ' \   #sélectionner la 1ère colonne avec espace comme délimiteur
      | tail -n +2 > modules.txt  #exclure la 1ère ligne (en-tête)
      <modules.txt xargs -d $'\n' \  #le contenu du fichier modules.txt comme entrée, transformer les lignes de ce fichier en arguments à la commande suivante
            sh -c 'for arg do modinfo "$arg" \
            | (grep description: || printf "\n") \
            | sed "s/^[[:alnum:]]*:[[:space:]]*//";done' _ > descriptions.txt  #sh pour exécuter la commande entre '' qui consiste en une boucle for qui affiche les informations détaillées de chaque module en argument, recherche les lignes de description non vides et les affiche, le résultat est redirigé dans le fichier descriptions.txt
      echo "" >> modules.txt #ajouter une ligne vide à la fin du fichier modules.txt
      sed -i "1s/^/Module\n/" modules.txt  #ajouter l'entête Module à la première ligne du fichier modules.txt
      echo "" >> descriptions.txt  #ajouter une ligne vide à la fin du fichier descriptions.txt
      sed -i "1s/^/Description\n/" descriptions.txt  #ajouter l'entête Description à la première ligne du fichier descriptions.txt
      pr -m -t modules.txt descriptions.txt \ #combiner les deux fichiers en faisant correspondre les lignes et en les séparant par une tabulation
      | head -n -1  #supprimer la dernière ligne qui sera vide
      rm modules.txt descriptions.txt  #supprimer les fichiers temporaires modules.txt et descriptions.txt
      ;;
```

```
2)    echo ""
      echo "Donnez le module à désactiver"
      read mod
      sudo modprobe -r $mod  #la commande modprobe avec
l'option -r permet de désactiver le module passé en argument
      ;;
3)    echo ""
      echo "Donnez le module à activer"
      read mod
      sudo modprobe $mod  #la commande modprobe permet
d'activer le module passé en argument
      ;;
4)    lsmod \  #lister tous les modules actifs
      | cut -f 1 -d ' ' \  #extraire la 1ère colonne avec
espace comme délimiteur
      | tail -n +2 > modules.txt  #mettre tout le contenu à
partir de la 2ème ligne (supprimer l'entête) dans le fichier
modules.txt
      find /lib/modules/$(uname -r) -type f -name '*.ko*' \
#rechercher tous les fichiers de modules (à l'extension .ko)
dans le répertoire des modules (/lib/modules/$(uname -r))
      | sed -n -e 's/^.*\///p' \  #supprimer le chemin de
chaque fichier
      | sed -n -e 's/\.ko.*//p' > all_modules.txt
#supprimer l'extension (.ko) de chaque fichier pour ne laisser
que les noms des modules et puis rediriger le résultat dans le
fichier modules.txt
      grep -v -x -f modules.txt all_modules.txt  #rechercher
tous les lignes/modules de all_modules.txt qui ne figurent pas
dans modules.txt (recherche inverse par l'option -v)
      rm modules.txt all_modules.txt   #supprimer les fichiers
temporaires modules.txt et all_modules.txt
      ;;
5)    exit
      ;;
esac

done
```