**TP1 - p14**

**--------------------------------------------------------------------**

**Objectifs**

**--------------------------------------------------------------------**

* apprendre à écrire et exécuter des scripts shell
* Se familiariser avec l’éditeur de texte vi
* Utiliser les commandes de gestion des modules

**--------------------------------------------------------------------**

**Description du travail demandé**

**--------------------------------------------------------------------**

Dans ce TP, on propose de créer un menu à partir duquel un utilisateur choisit une des options proposées pour l'exécuter :

* \*afficher la liste des modules : avec la commande **lsmod**
* \*supprimer l’un des modules en choisissant son numéro et le sauvegarder avec son nom dans un historique : On utilisera ainsi la commande **rmmod** (equivalente à **modprobe -r** / **modprobe --remove** )
* \*ajouter un modules de la liste des modules supprimés (afficher l’historique) en saisissant son numéro :avec la commande **modprobe -v** (equivalente à **insmod** )

De plus , on a choisi **App1** comme nom de l’application et l’historique des suppression des modules sera sauvegardé dans le fichier **SuppHis**

**--------------------------------------------------------------------**

**Script**

**--------------------------------------------------------------------**

**while** **echo** "Choisir une des commandes suivantes : \n 0.exit \n 1. Lister les modules du kernel \n 2.Afficher les informations du système\n";

**read** choix ;

**[** **$choix** -ne 0 **]** ;

**do**

**if [** **$choix** -eq 1 **] ;**

**then**

**lsmod** | nl

**while** **echo** "choisir une des commandes suivantes :” ; **echo** “ 0. exit”; **echo** “ 1. Ajouter un module“; **echo** “ 2. Supprimer un module";

**read** choixModif ;

**[ $choixModif** -ne 0 **]** ;

**do**

**if [$choixModif** -eq 2 **]**

**then**

**echo** "entrer le numéro du module à supprimer :";

**read** numS ;

ligne=**$(lsmod** | head -**$numS** | tail -1 | cut -d" " -f1 **)** ;

**rmmod** **$ligne**;

**printf** "%d %s\n" **$numS $ligne** >> SuppHis;

**elif [ $choixModif** -eq 1 **]**

**then**

cat SuppHis ;

**echo** "entrer le numéro du module à ajouter :" ;

**read** numAj ;

**while** **read** num nom

**do**

**echo** **$num $nom** ;

**if [ $num** -eq **$numAj** **]** ;

**then**

**modprobe -v** **$nom** ;

**sed -i** '/'**$numAj**'/d' SuppHis ;

**fi ;**

**done** < SuppHis

**else** **echo** "Ce choix n'est pas listé";

**fi ;**

**done ;**

**elif [** **$choix** -eq 2 **] ;**

**then** uname

**else** **echo** "Ce choix n'est pas listé"

**fi ;**

**done**

**--------------------------------------------------------------------**

**Notes**

**--------------------------------------------------------------------**

* Pour créer un fichier exécutable :

1. Créer un fichier vierge avec la commande :

**touch [fileName]**

1. Ajouter la permission d’exécution au fichier :

**chmod +x [fileName]**

1. Exécuter ce script avec la commande : **./[fileName]**

* Il faut être administrateur (root) pour exécuter les commandes de gestion des modules.
* Pour exécuter la commande insmod , il faut spécifier le chemin absolu du module alors qu’on utilisant modprobe , il suffit de spécifier le nom du module
* Le nom d’un module peut différer d’une distribution linux à une autre (open source) , dans ce cas , la commande **modinfo [NomModule]** nous sera utile comme elle fournit une description du module en question ainsi que d'autres informations.

**--------------------------------------------------------------------**

**Informations sur les commandes**

**--------------------------------------------------------------------**

* **lsmod** : lists the loaded kernel modules
* **uname** : displays the information about the system
* **rmmod <moduleName>** or **modprobe -r <moduleName>** : unload the module
* **insmod <FullModulePath>** or **modprobe <ModuleName>** : loads the kernel module but the main difference between the two commands is that modprobe command loads the module with its dependencies (if it needs another module it downloads it as well )

**--------------------------------------------------------------------**

**Exécution**

**--------------------------------------------------------------------**

* Pour l’exécution , on a choisi de tester sur les deux modules concernant la **caméra** et la **souris tactile du PC** appelés respectivement **uvcvideo** et **i2c\_hid**
* L'exécution est en vidéo : Camera.mp4 et Souris.mp4

**--------------------------------------------------------------------**

**Réalisé par : Helali Hadil et Trimech Raoua**

**GL 3/1**

**--------------------------------------------------------------------**