

TP 7 - Boot, services et processus / Tâche d'administration

Exercice 1. Personnalisation de GRUB

2.

Pour avoir le menu de GRUB affiché pendant 10s on modifie les lignes

```
GRUB_TIMEOUT_STYLE=menu
GRUB_TIMEOUT=10
```

et on laisse la ligne GRUB_DEFAULT=0

3.

On lance la commande

```
sudo update-grub
```

afin de construire le fichier */boot/grub/grub.cfg* à partir du fichier de paramètre et des scripts.

5.

Pour modifier la résolution de GRUB on décommente la ligne ci-dessous dans le fichier */etc/default/grub* et on choisit la valeur par exemple

```
GRUB_GFXMODE=1280x800
```

6.

Pour modifier le fond d'écran on ajoute la ligne au fichier */etc/default/grub*

```
GRUB_BACKGROUND=/usr/share/images/grub/nom_image
```

7.

Pour modifier le thème on installe le paquet *grub2-themes-** et on ajoute la ligne fichier **/etc/default/grub*

```
GRUB_THEME=/boot/grub/themes/ubuntustudio/theme.txt
```

8.

On ajoute au fichier *40_custom* présent dans le dossier */etc/grub.d* les lignes :

```
menuentry "Eteindre" {
    halt
}
menuentry "Redémarrer" {
    reboot
}
```

9.

Pour modifier la langue du clavier on ajoute la ligne au fichier */etc/default/grub*

```
GRUB_LANG=fr
```

Exercice 2. Noyau

1.

On installe le paquet *build-essential*

```
sudo build-essential
```

4.

Pour compiler le module on utilise la commande *make* et on l'installe avec la commande *make install*

5.

Pour charger le module et pour vérifier qu'il a bien été chargé on utilise les commandes

```
insmod hello.ko
lsmod
```

On affiche le journal du noyau avec la commande *sudo dmesg* pour vérifier que le module est bien chargé

```
[Hello world] - La fonction init_module () est appelée .
```

6.

Pour obtenir des informations sur le module hello.ko on utilise la commande

```
modinfo hello.ko
```

On retrouve les informations figurant dans le fichier hello.c transmise dans les macros *MODULE_XXX*

7.

Pour décharger le module et pour vérifier qu'il a bien été déchargé on utilise les commandes

```
rmmod hello.ko
lsmod
```

On lit bien

```
[Hello world] - La fonction cleanup_module () est appelée .
```

dans le journal du noyau.

8.

On ajoute dans le fichier */etc/modules* la ligne hello.

Exercice 3. Interception de signaux

1.

```
#!/bin/bash

function writer() {
while ( true )
do
    read -p "" ligne;
    echo $ligne >> ./tmp.txt;
done
}
writer
```

On ligne tout ce que l'on écrit en ligne de commande et l'on redirige le contenu de la ligne dans le fichier tmp.txt. A chaque exécution la saisie est rajoutée à la fin du fichier.

2.

Lorsque l'on appui sur CTRL+Z le processus est mis en pause. Pour poursuivre son execution on utilise la commande `fg`

3.

Lorsque l'on appui sur CTRL+C le processus s'arrête.

4.

```
#!/bin/bash

trap "echo 'Impossible de me placer en arrière-plan' " 20;
trap "echo 'OK, je fais un peu de ménage avant'; rm tmp.txt && killall writer.sh" 2 3;

function writer() {

while ( true )
do
    read -p "" ligne;
    echo $ligne >> ./tmp.txt;
done
}

writer
```

5.

On test le script avec les commande clavier et les commandes suivantes

```
kill -SIGTSTP PID
kill -SIGINT PID
```

6.

Pour corriger l'erreur on modifie la ligne suivante

```
trap "echo 'OK, je fais un peu de ménage avant'; test -a tmp.txt && rm tmp.txt; killall writer.sh" 2 3;
```

Exercice 4. Surveillance de l'activité du système

1.

La commande `w` affiche les utilisateurs présents sur le système et leur activité.

2.

Afin d'afficher l'historique des dernières connexions à la machine on utilise la commande `last`

3.

Pour connaître la version du noyau en cours d'utilisation on peut utiliser les commande

```
uname -mr
ou
cat /proc/version
```

4.

Pour récupérer toutes les informations sur le processeur au format JSON on utilise la commande

```
lscpu -J
```

5.

Pour obtenir la liste des derniers démarrages de la machine avec la commande `journalctl` on utilise l'option `--list-boots`

```
journalctl --list-boots
```

Pour afficher tout ce qu'il s'est passé sur la machine lors de l'avant-dernier boot on utilise la commande

```
journalctl --boots
```

6.

Pour prévenir les utilisateurs lors d'une connexion à la machine on édite le fichier `/etc/motd` en ajoutant la ligne indiquant l'évènement le message apparaîtra après que l'utilisateur se soit connecté.

Pour que le message soit affiché avant l'invite de saisie de login il faut mettre le message dans le fichier `/etc/issue`

7.

```
#!/bin/bash

u0=1
u1=1

echo "Les $1 premiers termes de la serie sont  "

for (( i=0; i<$1; i++ ))
do
    echo -n "$u0"
    un=$((u0 + u1))
    u0=$u1
    u1=$un
done
```

On observe la montée en charge du système grâce à la représentation graphique que fournit `tload`. À l'arrêt du programme avec `CTRL+C` la charge moyenne du système redescend.

Commande

`iftop` : examine le transit réseau
`glances` : moniteur de ressources

Commande	Actions
<code>who / w</code>	qui est connecté / qui fait quoi
<code>last</code>	historique des connexions sur la machine
<code>tload</code>	affiche la charge du système
<code>dmesg (-T)(-lerr -k)</code>	affiche les messages en provenance du noyau (affiche date) (affiche message d'erreurs venant du noyau)
<code>sysctl</code>	permet d'afficher et de configurer les paramètres du noyau (à chaud)
<code>hostname</code>	affiche le nom de la machine
<code>uname</code>	affiche des informations sur le système (version du noyau...)
<code>lshw</code>	liste le matériel
<code>lsusb</code>	liste les périphériques USB
<code>at</code>	exécute une tâche en différé

Cron

Commande

Commande

minute	heure	jour du mois	mois	jour de la semaine	commande
--------	-------	--------------	------	--------------------	----------

Afficher (supprimer, éditer) une crontab

```
crontab -l (-r -e) [-u user]
```