

CORRECTION ATELIER :

Planification des tâches

Exercice 1 :

Configurer **votre** table de cron « **crontab** » pour ajouter les travaux suivants :

```
Latifa@esprit-virtual-machine:~$ crontab -e
```

Ajouter les tâches correspondantes à votre fichier crontab

1. Copier les fichiers de journalisation dans un répertoire nommé **archive_log**. La commande sera exécutée tous les 1 et 15 de chaque mois à 8h.30.

```
30 8 1,15 * * cp -r /var/log archive_log
```

2. Afficher les partitions que vous avez dans un fichier nommé **disque**. La commande sera exécutée tous les 5 jours à 15h.30

```
30 15 */5 * * fdisk -l | grep ^/dev/sd[a-z][1-9]* > disque
```

3. Lister les disques HDD dans votre machine. La commande sera exécutée le premier et le dernier jour du mois à 1 heure.

```
0 1 1,30 * * fdisk -l | grep '^Disk /hd' | wc -l
```

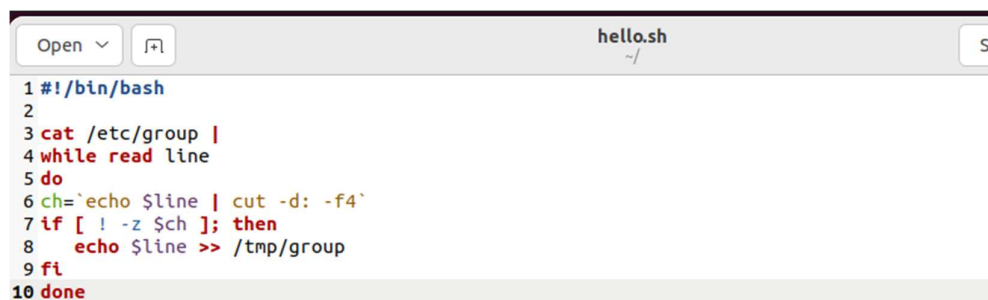
4. Copier les espaces de travail de tous les utilisateurs sous le répertoire **/tmp/backup**, chaque semaine à 20h22.

```
22 20 */7 * * cp -r /home /tmp/backup
```

5. Ecrire la ligne dans la **table cron** qui permet d'exécuter le script **hello.sh** une fois par heure entre trois heures et cinq heures de l'après-midi chaque lundi et jeudi.

Ce script permet d'afficher la liste des groupes qui sont des groupes secondaires pour certains utilisateurs et les enregistrer dans un fichier **/tmp/group**.

```
*/59 15-17 * * 1,4 /home/Latifa/hello.sh
```



```
Open hello.sh
1 #!/bin/bash
2
3 cat /etc/group |
4 while read line
5 do
6 ch=`echo $line | cut -d: -f4`
7 if [ ! -z $ch ]; then
8   echo $line >> /tmp/group
9 fi
10 done
```

N'oublier pas d'ajouter le droit d'exécution au script

```
Latifa@esprit-virtual-machine:~$ chmod +x hello.sh
Latifa@esprit-virtual-machine:~$
```

Exercice 2 :

Ajouter les travaux (jobs) suivants dans **votre table cron** :

1. Le job fait appel à un script Shell « *script.sh* ». Il permet d’afficher les partitions de type « *Linux* » ainsi que les partitions qui contiennent vos « *bootloaders* » dans un fichier nommer **disque**. Ce script sera exécuté chaque mois vers minuit.

```
@monthly /latifa/home/script.sh
```

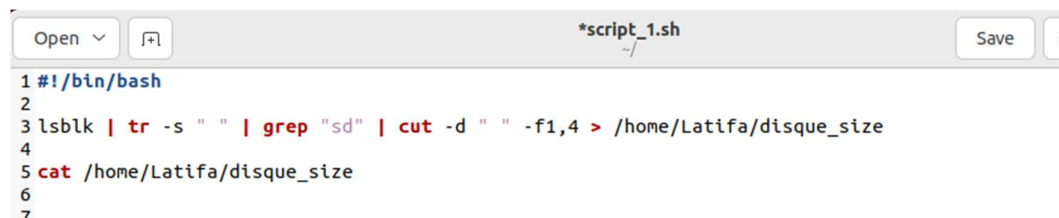
```
Latifa@esprit-virtual-machine:~$ gedit script.sh
Latifa@esprit-virtual-machine:~$ chmod +x script.sh
Latifa@esprit-virtual-machine:~$
```



```
script.sh
~/
1 #!/bin/bash
2
3 sudo fdisk -l | grep "Linux" > /home/Latifa/disque
4 sudo fdisk -l | grep "EFI" >> /home/Latifa/disque
5 cat /home/Latifa/disque
```

2. Le job fait appel à un script Shell « *script_1.sh* ». Il permet d’afficher uniquement le nom de chaque disque et sa taille. Ce script sera exécuté chaque 10 minutes le 1er janvier à 13h

```
*/10 13 1 1 * /latifa/home/script_1.sh
```



```
*script_1.sh
~/
1 #!/bin/bash
2
3 lsblk | tr -s " " | grep "sd" | cut -d " " -f1,4 > /home/Latifa/disque_size
4
5 cat /home/Latifa/disque_size
6
7
```

Exercice 3 :

Pour les questions suivantes, il faudra créer un utilisateur “**test**” sans aucun droit particulier.

```
Latifa@esprit-virtual-machine:~$ sudo useradd test
Latifa@esprit-virtual-machine:~$
```

1. Vérifier que la configuration actuelle permet à n’importe quel utilisateur d’utiliser cron

```

Latifa@esprit-virtual-machine:~$ sudo su - test
test@esprit-virtual-machine:~$ crontab -e
no crontab for test - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]: 1
crontab: installing new crontab
test@esprit-virtual-machine:~$

```

L'utilisateur « test » peut gérer sa table cron.

2. Créer un fichier **/etc/cron.allow** vide. Quelle conséquence cela a-t-il pour “test” ?

```

Latifa@esprit-virtual-machine:~$ sudo touch /etc/cron.allow
Latifa@esprit-virtual-machine:~$ sudo su - test
test@esprit-virtual-machine:~$ crontab -e
You (test) are not allowed to use this program (crontab)
See crontab(1) for more information
test@esprit-virtual-machine:~$

```

Après la création du fichier « **/etc/cron.allow** » aucun utilisateur ne peut configurer ses jobs.

Seuls les utilisateurs qui vont être ajoutés à ce fichier peuvent le faire.

3. Supprimer le fichier **/etc/cron.allow** et créer un autre fichier vide **/etc/cron.deny**. L'utilisateur « test » peut-il programmer des tâches ?

```

Latifa@esprit-virtual-machine:~$ sudo rm /etc/cron.allow
Latifa@esprit-virtual-machine:~$ sudo touch /etc/cron.deny
Latifa@esprit-virtual-machine:~$ sudo su - test
test@esprit-virtual-machine:~$ crontab -e
No modification made
test@esprit-virtual-machine:~$

```

Maintenant « test » peut programmer ses tâches, sauf lorsqu'il sera ajouté au fichier **/etc/cron.deny**

4. Quel est le comportement si un nom d'utilisateur apparait dans le fichier **cron.allow** et aussi dans le fichier **cron.deny** ?

```

Latifa@esprit-virtual-machine:~$ sudo touch /etc/cron.allow
Latifa@esprit-virtual-machine:~$ su -
Password:
root@esprit-virtual-machine:~# echo test > /etc/cron.allow
root@esprit-virtual-machine:~# echo test > /etc/cron.deny
root@esprit-virtual-machine:~# cat /etc/cron.allow
test
root@esprit-virtual-machine:~# cat /etc/cron.deny
test
root@esprit-virtual-machine:~# exit
logout
Latifa@esprit-virtual-machine:~$ sudo su - test
test@esprit-virtual-machine:~$ crontab -e
No modification made
test@esprit-virtual-machine:~$

```

L'utilisateur « test » est ajouté au deux fichiers **/etc/cron.allow** et **/etc/cron.deny** mais il peut gérer ses jobs.

Ces deux fichiers de contrôle d'accès fonctionnent ensemble comme suit :

- ✓ Si **cron.allow** existe, seuls les utilisateurs répertoriés dans ce fichier peuvent créer, modifier, afficher ou supprimer des fichiers crontab. → Le fichier le plus prioritaire
- ✓ Si **cron.allow n'existe pas**, tous les utilisateurs peuvent envoyer des fichiers crontab, à l'exception des utilisateurs répertoriés dans **cron.deny**.
- ✓ Si ni **cron.allow** ni **cron.deny** n'existent, les privilèges de super utilisateur sont requis pour exécuter la commande crontab.

Les privilèges de super utilisateur sont requis pour modifier ou créer les fichiers **cron.deny** et **cron.allow**.