

SAE – Installation d'un poste développement

Notice par Thomas Robineau S1C

1. Création des utilisateurs du poste

- 1.1. Utilisateurs standards
- 1.2. Administrateurs
- 1.3. Modification des mots de passe

2. Création et assignation des groupes aux utilisateurs

- 2.1. Création des groupes g1, g2 et admin
- 2.2. Assignation des groupes g1 et g2 aux utilisateurs standards
- 2.3. Assignation du groupe admin aux administrateurs

3. Installation des logiciels requis

- 3.1. Installation d'Emacs
- 3.2. Installation de Apache2
- 3.3. Installation Wireshark
- 3.4. Installation de IntelliJ
- 3.5. Installation de Java

4. Lancement de logiciels au démarrage

- 4.1. Apache2
- 4.2. Page Web Apache

5. Initialisation des scripts de gestion de corbeille

- 5.1. Obtention des scripts
- 5.2. Modification des droits des scripts
- 5.3. Ajout des scripts dans le PATH (Rendre les scripts utilisables par tous les utilisateurs)

6. Création des alias

- 6.1. Script à écrire et à exécuter

Partie 1 : Création des utilisateurs du poste

1.1 Utilisateurs standards

Lancement de la commande de création dans le terminal :

```
etudiant@Ubuntu20:~$ sudo useradd -m -s /bin/bash u1  
[sudo] Mot de passe de etudiant :
```

- Relancer la commande 3 nouvelles fois en remplaçant u1 par u2, puis u3 puis u4.

Il sera nécessaire de taper le mot de passe de l'utilisateur entrant la commande pour pouvoir la confirmer (une seule fois par instance de terminal)

1.2 Administrateurs

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo useradd -m -s /bin/bash a1
```

- Relancer la commande une nouvelle fois en remplaçant a1 par a2 pour créer le 2^e administrateur

Taper le mot de passe de l'utilisateur courant s'il est demandé

1.3 Modification des mots de passe

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo passwd a1  
Nouveau mot de passe :  
Retapez le nouveau mot de passe :  
passwd : le mot de passe a été mis à jour avec succès
```

- Remplacer a1 par le nom de l'utilisateur dont on veut modifier le mot de passe.

NB : Dans le cadre de cette SAE, les mots de passe sont les noms d'utilisateur

Taper le mot de passe de l'utilisateur courant s'il est demandé

Partie 2 : Création et assignation des groupes aux utilisateurs

2.1 Création des groupes g1 et g2

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo groupadd g1
```

- Remplacer g1 par g2 pour créer le second groupe et par admin pour créer le groupe administrateur

2.2 Assignation des groupes g1 et g2 aux utilisateurs standards

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo usermod -G g1,g2 u1
```

- Remplacer u1 par u2, u3, u4 pour faire l'affectation de groupes au reste des utilisateurs
- On vérifie que la commande s'est bien exécutée avec la commande *groups*

```
etudiant@Ubuntu20:~$ groups u1  
u1 : u1 g1 g2
```

- On vérifie bien que les 2 groupes figurent sur le retour la commande

2.3 Assignation du groupe admin aux administrateurs

Lancement de la commande dans le terminal

```
etudiant@Ubuntu20:~$ sudo usermod -G admin,sudo a1
```

- On renseigne le groupe admin AINSI QUE le groupe **sudo** (Tous les droits sur la machine)
- Encore une fois, on vérifie que l'attribution a bien été effectuée avec la commande *groups*

```
etudiant@Ubuntu20:~$ groups a1
a1 : a1 sudo admin
```

- On vérifie pour a1 puis pour a2 que les groupes admin et sudo apparaissent bien

Partie 3 : Installation des logiciels requis

3.1 Installation de Emacs

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo apt install emacs
```

- Le terminal va demander une confirmation de l'utilisateur pour continuer l'installation. **Répondre « O »** (oui) ou « **y** » en fonction de la lettre demandée

```
Souhaitez-vous continuer ? [O/n]
```

3.2 Installation de Apache2

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo apt install apache2
```

- Une nouvelle fois, le terminal pourra demander une confirmation de l'utilisateur. Suivre la démarche précédente

3.3 Installation de Wireshark

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo apt install wireshark
```

- Une nouvelle fois, le terminal pourra demander une confirmation de l'utilisateur. Suivre la démarche précédente

3.4 Installation de Java

Installation des dépendances requises pour l'installation de Java :

```
etudiant@Ubuntu20:~$ sudo apt install libc6-i386 libc6-x32 curl -y
```

- Cette commande télécharge les bibliothèques **libc6-i386** et **libc6-x32** de Java

Installation du package de Java

```
etudiant@Ubuntu20:~$ sudo wget https://download.oracle.com/java/17/latest/jdk-17_linux-x64_bin.deb
```

- Cette commande télécharge le fichier d'installation de java à partir de l'URL renseignée dans la commande (directement sur le site d'Oracle)
- Si la commande s'est déroulée correctement, le terminal doit renvoyer un message du type :

```
2021-12-08 17:23:03 (44,4 MB/s) - «jdk-17_linux-x64_bin.deb.1» enregistré [155215128/155215128]
```

Dépaquetage du package de Java

```
etudiant@Ubuntu20:~$ sudo dpkg -i jdk-17_linux-x64_bin.deb
```

- Cette commande décompresse le fichier téléchargé à l'étape précédente. Il sera utilisé dans la commande suivante

Installation finale de Java

```
etudiant@Ubuntu20:~$ sudo apt install openjdk-17-jre-headless
```

- Le mot de passe de l'utilisateur pourra être demandé. Attendre la fin de la commande. Java est installé.
- On lance la commande suivante pour vérifier que l'installation s'est bien passée.

```
etudiant@Ubuntu20:~$ java --version
openjdk 17 2021-09-14
OpenJDK Runtime Environment (build 17+35-Ubuntu-120.04)
OpenJDK 64-Bit Server VM (build 17+35-Ubuntu-120.04, mixed mode, sharing)
```

Partie 4 : Lancement de logiciels au démarrage

4.1 Apache2

Lancement de la commande dans le terminal :

```
etudiant@Ubuntu20:~$ sudo systemctl enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
```

- Désormais, apache2 démarrera au démarrage de l'ordinateur.
- Son activation est vérifiable via la commande top. Saisir SHIFT + L puis saisir apache2 puis ENTREE. On voit bien que apache2 est activé (Explications ci-dessous)

```
etudiant@Ubuntu20:~$ top
```

Trouve chaîne

Trouve chaîne apache2

PID	UTIL.	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TEMPS+	COM.
714	root	20	0	6532	4424	3252	S	0,0	0,2	0:00.02	apache2
715	www-data	20	0	752668	4220	2676	S	0,0	0,2	0:00.00	apache2
716	www-data	20	0	752668	4220	2676	S	0,0	0,2	0:00.00	apache2

Partie 5 : Initialisation des scripts de gestion de corbeille

5.1 Obtention des scripts

Cette étape nécessite d'avoir 4 scripts de gestion d'un corbeille :

- clt : CLean Trash (Nettoyer la corbeille)
- rft : Restore From Trash (Restaurer un fichier)
- lkt : Look in Trash (Chercher un fichier dans la corbeille)
- srm : Safe ReMove (Suppression sécurisée)

Création du dossier qui accueillera les scripts :

```
etudiant@Ubuntu20:~$ mkdir /scripts
```

Puis

```
etudiant@Ubuntu20:~$ cd /scripts
```

Pour recopier chacun de ces scripts, dans le terminal, lancer la commande : (Remplacer les points par srm, lft, rft, clt)

```
etudiant@Ubuntu20:~$ emacs ...
```

NB : Pour enregistrer un fichier sous emacs, faire CTRL + X, CTRL + S.

Code de ces scripts :

CLT :

RFT :

```
#!/bin/bash

if [ -z "$(ls -A $HOME'/trash/')" ]
then
    echo "La poubelle est vide"
else
    rm -r $HOME'/trash/'
    mkdir $HOME'/trash'
    echo "La poubelle a été vidée"
fi
exit 0
```

```
#!/bin/bash

usage="usage: $0 files..."
if [ $# -gt 0 ]
then
    echo $usage
    exit 1
fi
if [ ! -d ~/trash ]
then
    mkdir ~/trash
fi
if [ $# -eq 1 ]
then
    dir = '.'
else
    dire = $2
fi
fich = $1
if mv ~/trash/${fich} ${dir}
then
    echo "${fich} restored in ${dir}"
fi
echo 0
```

LKT :

```
#!/bin/bash
usage="usage: $0 files..."
if [ $# -gt 0 ]
then
    echo $usage
    exit 1
fi
if [ ! -d ~/trash ]
then
    mkdir ~/trash
fi
echo "trash folder :"
cd ~/trash
for machin in *
do
    if [ $machin = * ]
    then
        continue
    fi
    ls -l machin
done
exit 0
```

SRM :

```
#!/bin/bash
usage="usage: $0 files..."
if [ $# -gt 0 ]
then
    echo $usage
    exit 1
fi
if [ ! -d ~/trash ]
then
    mkdir ~/trash
fi
for f in $*
do
    if mv $f ~/trash
    then
        echo ":$f in trash"
    fi
done
exit 0
```

5.2 Modification des droits des scripts

Dans le dossier scripts, lancer cette commande

```
etudiant@Ubuntu20:/scripts$ sudo chmod 755 *
```

Les droits des fichiers doivent avoir été modifiés.

Vérifier avec « ls -l »

```
etudiant@Ubuntu20:/scripts$ ls -l
total 16
-rwxr-xr-x 1 etudiant etudiant 101 déc. 7 11:09 clt
-rwxr-xr-x 1 etudiant etudiant 267 déc. 7 11:12 lkt
-rwxr-xr-x 1 etudiant etudiant 292 déc. 7 11:15 rft
-rwxr-xr-x 1 etudiant etudiant 223 déc. 7 11:16 srm
```


5.3 Ajout des scripts dans le PATH

Lancement de la commande dans le terminal

```
etudiant@Ubuntu20:~$ cd
```

Puis recopier le script dans un fichier (emacs scriptpath)

```
#!/bin/bash

for a in $(ls /home)
do
    echo "export PATH=$PATH:/scripts" >> /home/$a/.bashrc
done
```

Puis lancer la commande `sudo bash scriptpath`

Un redémarrage de l'ordinateur est nécessaire pour que les modifications prennent effet

Partie 6 : Création des alias

6.1 Script à écrire et à exécuter

```
#!/bin/bash

for a in $(ls /home)
do
    echo "alias e='emacs'" >> /home/$a/.bashrc
    echo "alias w='wireshark'" >> /home/$a/.bashrc
done
```

Réécrire ce script avec emacs (commande : `emacs scriptalias`)

Enregistrer (CTRL + X, CTRL + S)

Exécuter le script (commande : `sudo bash scriptalias`)

Redémarrage de l'ordinateur requis