

TD_9_Linux_Guillaume_Sanchez

Exercice 1

```
#!/bin/bash

# Liste des utilisateurs à créer
tab_user=('yannick' 'eulalie' 'jean' 'justine' 'pierre' 'sandrine' 'cecile' 'michel')

# Boucle pour créer chaque utilisateur
for user in "${tab_user[@]}"
do
    # Création de l'utilisateur
    useradd -m -s /bin/bash $user
    echo -e "$user\n$user" | passwd $user
    echo "Utilisateur $user créé avec succès avec le mot de passe par défaut."
done
```

Voici un **for** qui boucle le tableau contenant les noms d'utilisateurs.

useradd -m -s /bin/bash \$user permet de créer l'utilisateur.

echo -e "\$user\n\$user" | passwd \$user permet d'attribuer le mot de passe de chaque utilisateur.

Exercice 2

Voici le code de mon script pour réaliser les consignes de l'exercice 2

```
#!/bin/bash

echo "Exercice 2"
echo "Créer un groupe visiteurs avec le GID 1404"
groupadd visiteurs -g 1404
echo "Créer un utilisateur invite avec le UID 140401 et le GID 1404"
useradd -u 140401 -g visiteurs -m -s /bin/bash invite
echo "affichez les lignes correspondantes des fichiers utilisateurs et groupes"
cat /etc/passwd | grep invite
cat /etc/group | grep visiteurs
echo "modification des droits de changement de mdp de l'utilisateur invite"
chage -M 90 -W 14 -I 7 invite
echo "affichez la lignes correspondant à l'utilisateur invite dans le fichier shadow"
cat /etc/shadow | grep invite
echo "modification du mot de passe de l'utilisateur invite"
echo -e "welcome\nwelcome" | passwd "invite"

echo "créer un utilisateur intrus avec le UID 140402 et le GID 1404"
useradd -u 140402 -g visiteurs -m -s /bin/bash intrus
echo "modification des droits de changement de mdp de l'utilisateur intrus"
chage -M 90 -W 14 -I 7 invite
echo "modification du mot de passe de l'utilisateur intrus"
echo -e "1234\n1234" | passwd intrus
echo "afficher les 2 dernière lignes du fichier shadow"
tail -n 2 /etc/shadow
```

Voici le résultat lors de l'exécution de mon script :

```
root@ubuntu-RSX112:~/tp-28-04-2025# ./exercice2.sh
Exercice 2
Créer un groupe visiteurs avec le GID 1404
Créer un utilisateur invite avec le UID 140401 et le GID 1404
useradd: user 'invite' already exists
affichez les lignes correspondantes des fichiers utilisateurs et groupes
invite:x:140401:1404:./home/invite:/bin/bash
visiteurs:x:1404:
modification des droits de changement de mdp de l'utilisateur invite
affichez la lignes correspondant à l'utilisateur invite dans le fichier shadow
invite:$6$ZSA4ZzSk4cCxMdfH$6L.2/QxVUWV8l9p6yuHCrtQ5aHESL4aScCIS5fVdqE/e3YD/An.dGtqWBTsZvQUKy8mBcilG4IUT.P.eJ0p0x/:20206:0:90:14:7::
modification du mot de passe de l'utilisateur invite
New password: Retype new password: passwd: password updated successfully
créer un utilisateur intrus avec le UID 140402 et le GID 1404
modification des droits de changement de mdp de l'utilisateur intrus
modification du mot de passe de l'utilisateur intrus
New password: Retype new password: passwd: password updated successfully
affichez les 2 dernière lignes du fichier shadow
invite:$6$rnukBu8/ffeoQ/B$XmJKezXD3ER0HD5xM95YpA.RL/jxI94CwhpYq3cX.LNYcc94nUVLNZuxEHpzwR/dkiDhQ.850dSGS.Fx4.JJz/:20206:0:90:14:7::
intrus:$6$B6HpC5eAcvuy$60A$aovF9qkXuinzXY00K8m3VusQ/ljCayGzQo47oyQbcqfr22BrX6hLe2JeVk6QZutEHRNSV0zVRasDSnMpFIXlW0:20206:0:99999:7:::
```

Pour répondre au point 10, nous pouvons remarquer que malgré l'attribution d'un changement de mot de passe tous le 90 jours, l'utilisateur intrus a son changement de mot de passe initié à 99999 jours.

Exercice 3

Voici le résultat des consignes de l'exercice 3 :

```
root@ubuntu-RSX112:~/tp-28-04-2025# su - invite
invite@ubuntu-RSX112:~$ echo "date" >> utcd.sh
invite@ubuntu-RSX112:~$ chmod u+x utcd.sh
invite@ubuntu-RSX112:~$ ls -l utcd.sh
-rwxr--r-- 1 invite visiteurs 5 Apr 28 11:44 utcd.sh
invite@ubuntu-RSX112:~$ ./utcd.sh
Mon Apr 28 11:45:01 UTC 2025
invite@ubuntu-RSX112:~$
```

Exercice 4

Voici un script qui rassemble les différentes consignes de l'exercice 4 :

- Les tableaux **tab_group** et **tab_user** :

```
tab_group=("technique" "formation" "commercial" "administratif")
tab_user=("philippe" "harouna" "emmanuelle" "nicolas" "laurent"
"franck" "patrick" "ali" "eric")
```

- La fonction **create_user** :

```
function create_user {  
    echo  
    echo "Création des utilisateurs"  
    echo  
    for user in "${tab_user[@]}"  
    do  
        useradd "$user" -m -N -p $user  
        chage -M 90 -m 5 $user  
        echo "echo 'Bienvenue, $(whoami), sur le système Ubuntu-RSX112 !'" >> "/home/$user/.bashrc"  
        echo "Utilisateur $user créé avec succès avec le mot de passe par défaut."  
    done  
}
```

- La fonction **create_group** :

```
function create_group {  
    echo  
    echo "Création des groupes"  
    echo  
    for group in "${tab_group[@]}"  
    do  
        groupadd $group  
        mkdir /$group  
        chgrp $group /$group  
        chmod 770 /$group  
        echo "Groupe $group créé avec succès."  
    done  
}
```

- La fonction **Attribution_user_group** :

```
function Attribution_user_group {
    echo
    echo "Attribution des utilisateurs à leur groupe principale et secondaire"
    echo
    for user in "${tab_user[@]}"
    do
        if [[ $user == "philippe" ]]; then
            usermod -g technique $user
            echo "L'utilisateur $user a été ajouté au groupe principal technique."
        elif [[ $user == "harouna" ]]; then
            usermod -g technique $user
            echo "L'utilisateur $user a été ajouté au groupe principal technique."
        elif [[ $user == "emmanuelle" ]]; then
            usermod -g technique $user
            usermod -aG formation $user
            echo "L'utilisateur $user a été ajouté au groupe principal technique et secondaire formation."
        elif [[ $user == "nicolas" ]]; then
            usermod -g formation $user
            usermod -aG technique $user
            echo "L'utilisateur $user a été ajouté au groupe principal formation et secondaire technique."
        elif [[ $user == "laurent" ]]; then
            usermod -g technique $user
            usermod -aG formation $user
            usermod -aG commercial $user
            echo "L'utilisateur $user a été ajouté au groupe principal technique et secondaire formation, commercial."
        elif [[ $user == "franck" ]]; then
            usermod -g formation $user
            usermod -aG technique $user
            usermod -aG administratif $user
            echo "L'utilisateur $user a été ajouté au groupe principal formation et secondaire technique, administratif."
        elif [[ $user == "patrick" ]]; then
            usermod -g administratif $user
            echo "L'utilisateur $user a été ajouté au groupe principal administratif."
        elif [[ $user == "ali" ]]; then
            usermod -g commercial $user
            echo "L'utilisateur $user a été ajouté au groupe principal commercial."
        elif [[ $user == "eric" ]]; then
            usermod -g commercial $user
            usermod -aG administratif $user
            echo "L'utilisateur $user a été ajouté au groupe principal commercial et secondaire administratif."
        else
            echo "Aucun groupe n'a été attribué à l'utilisateur $user."
        fi
    done
}
```

- Voici le lancement du programme :

```
echo "=====
echo "Exercice 4 - Gestion des utilisateurs et des groupes"
echo "=====
create_user
create_group
Attribution_user_group
echo
echo "Mise en place du mot de passe de philippe"
echo
echo -e "My-Pe$$word\nMy-Pe$$word" | passwd philippe
echo
echo "Ajout de Ali dans le groupe de techniciens"
echo
usermod -aG technique ali
echo
```

On y retrouve également le changement de mot de passe de philippe et l'ajout de Ali dans le groupe des techniciens.

Voici le résultat de ce script :

```
=====
Exercice 4 - Gestion des utilisateurs et des groupes
=====

Création des utilisateurs

Utilisateur philippe créé avec succès avec le mot de passe par défaut.
Utilisateur harouna créé avec succès avec le mot de passe par défaut.
Utilisateur emmanuelle créé avec succès avec le mot de passe par défaut.
Utilisateur nicolas créé avec succès avec le mot de passe par défaut.
Utilisateur laurent créé avec succès avec le mot de passe par défaut.
Utilisateur franck créé avec succès avec le mot de passe par défaut.
Utilisateur patrick créé avec succès avec le mot de passe par défaut.
Utilisateur ali créé avec succès avec le mot de passe par défaut.
Utilisateur eric créé avec succès avec le mot de passe par défaut.

Création des groupes

Groupe technique créé avec succès.
Groupe formation créé avec succès.
Groupe commercial créé avec succès.
Groupe administratif créé avec succès.

Attribution des utilisateurs à leur groupe principale et secondaire

L'utilisateur philippe a été ajouté au groupe principal technique.
L'utilisateur harouna a été ajouté au groupe principal technique.
L'utilisateur emmanuelle a été ajouté au groupe principal technique et secondaire formation.
L'utilisateur nicolas a été ajouté au groupe principal formation et secondaire technique.
L'utilisateur laurent a été ajouté au groupe principal technique et secondaire formation, commercial.
L'utilisateur franck a été ajouté au groupe principal formation et secondaire technique, administratif.
L'utilisateur patrick a été ajouté au groupe principal administratif.
L'utilisateur ali a été ajouté au groupe principal commercial.
L'utilisateur eric a été ajouté au groupe principal commercial et secondaire administratif.

Mise en place du mot de passe de philippe

New password: Retype new password: passwd: password updated successfully

Ajout de Ali dans le groupe de techniciens
```

Naturellement, il est innavigable de laisser les noms des personnes dans le cadre professionnel, je les ai volontairement laissés pour l'exercice.

La fonction **Attribution_user_group** est réalisé uniquement dans le cadre de cet exercice. Un tableau contenant le nom des personnes et leurs groupes aurait été plus propre.

Exercice 5

Prérequis

- 1)
 - c. Un UID généralement supérieur ou égal à 1000
- 2)
 - Le fichier peut être lu, modifié et exécuté
 - L'utilisateur peut lire le contenu du fichier (r).
 - L'utilisateur peut ajouter, écrire dans le fichier (w).
 - L'utilisateur peut exécuter le fichier (x).
- 3)
 - L'utilisateur peut lister son contenu (r).
 - L'utilisateur peut ajouter, supprimer ou renommer des fichiers dans ce répertoire (w).
 - L'utilisateur peut entrer dans le répertoire et accéder à son contenu (x).
- 4)
 - 754
- 5)
 - rW-r-----
- 6)
 - SUID

Partie 1

Voici un whoami et groups de user1, user2 et root :

```
user1@ubuntu-RSX112:~$ whoami
user1
user1@ubuntu-RSX112:~$ groups
user1
user1@ubuntu-RSX112:~$ █

██
user2@ubuntu-RSX112:~$ whoami
user2
user2@ubuntu-RSX112:~$ groups
user2
user2@ubuntu-RSX112:~$ █

██
root@ubuntu-RSX112:~# whoami
root
root@ubuntu-RSX112:~# groups
root
root@ubuntu-RSX112:~# █
```

Partie 2

```
user1@ubuntu-RSX112:~$ mkdir chapitre1
user1@ubuntu-RSX112:~$ cd chapitre1/
user1@ubuntu-RSX112:~/chapitre1$ ls -ld
drwxrwxr-x 2 user1 user1 4096 Apr 28 14:14 .
user1@ubuntu-RSX112:~/chapitre1$ umask
0002
user1@ubuntu-RSX112:~/chapitre1$ touch fic1
user1@ubuntu-RSX112:~/chapitre1$ mkdir rep1
user1@ubuntu-RSX112:~/chapitre1$ ls -ld
drwxrwxr-x 3 user1 user1 4096 Apr 28 14:14 .
user1@ubuntu-RSX112:~/chapitre1$ ls -l
total 4
-rw-rw-r-- 1 user1 user1 0 Apr 28 14:14 fic1
drwxrwxr-x 2 user1 user1 4096 Apr 28 14:14 rep1
user1@ubuntu-RSX112:~/chapitre1$ umask 0027
user1@ubuntu-RSX112:~/chapitre1$ touch fic2
user1@ubuntu-RSX112:~/chapitre1$ mkdir rep2
user1@ubuntu-RSX112:~/chapitre1$ ls -l
total 8
-rw-rw-r-- 1 user1 user1 0 Apr 28 14:14 fic1
-rw-r----- 1 user1 user1 0 Apr 28 14:15 fic2
drwxrwxr-x 2 user1 user1 4096 Apr 28 14:14 rep1
drwxr-x--- 2 user1 user1 4096 Apr 28 14:15 rep2
user1@ubuntu-RSX112:~/chapitre1$ █
```

- 3) La valeur affichée 0002 définit les autorisations lors de la création des fichiers et des répertoires.
- 5) Oui cela correspond à la valeur attribué par le mask. 0 donne lecture et écriture (rw-) pour un fichier et exécution en plus pour un dossier (rwx) et le 2 ne

donne que la lecture (r-) pour un fichier et l'exécution en plus pour un dossier (r-x)

- 6) Cela signifie que lors de la création d'un fichier, l'utilisateur aura les droits rw, le groupe r et les autres aucun droit. Pour la création d'un dossier, rwx pour l'utilisateur, r-x pour le groupe et rien pour les autres. 2
- 8) Non les droits de fic1 et rep1 n'ont pas changés car ils ont été créés avant l'application du nouveau mask

Partie 3

```
user1@ubuntu-RSX112:~/chapitre1$ chmod o+rwx rep1
user1@ubuntu-RSX112:~/chapitre1$ chmod 777 rep2
user1@ubuntu-RSX112:~/chapitre1$ ls -l
total 8
-rw-rw-r-- 1 user1 user1  0 Apr 28 14:14 fic1
-rw-r----- 1 user1 user1  0 Apr 28 14:15 fic2
drwxrwxrwx 2 user1 user1 4096 Apr 28 14:14 rep1
drwxrwxrwx 2 user1 user1 4096 Apr 28 14:15 rep2
user1@ubuntu-RSX112:~/chapitre1$ mkdir /tmp/user1
user1@ubuntu-RSX112:~/chapitre1$ chmod o+rwx /tmp/user1/
user1@ubuntu-RSX112:~/chapitre1$ touch /tmp/user1/secret
user1@ubuntu-RSX112:~/chapitre1$ chmod 600 /tmp/user1/secret
user1@ubuntu-RSX112:~/chapitre1$

user2@ubuntu-RSX112: ~ 11
user2@ubuntu-RSX112:~$ cat /tmp/user1/secret
cat: /tmp/user1/secret: Permission denied
user2@ubuntu-RSX112:~$ rm /tmp/user1/secret
rm: remove write-protected regular empty file '/tmp/user1/secret'? y
user2@ubuntu-RSX112:~$ ls -l /tmp/user1
total 0
```

- 8) Oui, on peut supprimer, car les autres utilisateurs (dans notre exemple user2) ont le droit d'exécution sur le dossier user1 (chmod o+rwx user1) donc même s'il ne peut pas lire le fichier, car il n'a pas les droits, il peut tout de même le supprimer.

Partie 4

Voici les commandes avec user1 :

```
user1@ubuntu-RSX112:~$ cd chapitre1/
user1@ubuntu-RSX112:~/chapitre1$ umask 0002
user1@ubuntu-RSX112:~/chapitre1$ rm -Rf docperso/
user1@ubuntu-RSX112:~/chapitre1$ mkdir docperso
user1@ubuntu-RSX112:~/chapitre1$ chmod 777 docperso/
user1@ubuntu-RSX112:~/chapitre1$ chmod 700 docperso/
user1@ubuntu-RSX112:~/chapitre1$ cd docperso/
user1@ubuntu-RSX112:~/chapitre1/docperso$ touch fica
user1@ubuntu-RSX112:~/chapitre1/docperso$ touch ficb
user1@ubuntu-RSX112:~/chapitre1/docperso$ touch ficc
user1@ubuntu-RSX112:~/chapitre1/docperso$ touch fidd
user1@ubuntu-RSX112:~/chapitre1/docperso$ ls -l
total 0
-rw-rw-r-- 1 user1 user1 0 Apr 28 15:42 fica
-rw-rw-r-- 1 user1 user1 0 Apr 28 15:42 ficb
-rw-rw-r-- 1 user1 user1 0 Apr 28 15:42 ficc
-rw-rw-r-- 1 user1 user1 0 Apr 28 15:42 fidd
user1@ubuntu-RSX112:~/chapitre1/docperso$ ln fica /tmp/lienfica
user1@ubuntu-RSX112:~/chapitre1/docperso$ ln -s ficb /tmp/lienficb
user1@ubuntu-RSX112:~/chapitre1/docperso$
```

Voici les commandes avec user2 :

```
user2@ubuntu-RSX112:~$ ls -l /home/user1/chapitre1/docperso/
ls: cannot open directory '/home/user1/chapitre1/docperso/': Permission denied
user2@ubuntu-RSX112:~$ cat /tmp/lienfica
user2@ubuntu-RSX112:~$ cat /tmp/lienficb
cat: /tmp/lienficb: Permission denied
user2@ubuntu-RSX112:~$
```

- 6) Ce n'est pas possible car user1 a positionné les droits du dossier à 700 donc personne d'autre que lui ne peut y accéder.
- 7) C'est possible, car il s'agit d'un lien en dur (hard link). Il a donc les droits du fichier d'origine, c'est-à-dire 664.
- 8) Ce n'est pas possible, car il s'agit d'un lien symbolique (soft link). Le fichier possède donc les droits qu'user1 a défini à l'origine sur le dossier docperso.