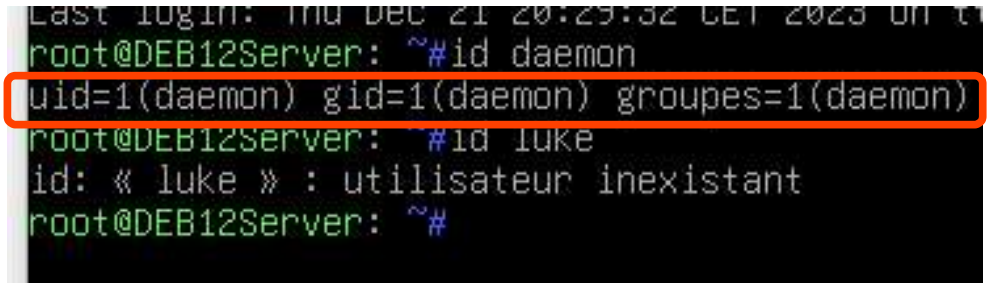
TP13 : Les utilisateurs et les droits

Sommaire

1. La gestion des utilisateurs.

1. Est-ce que les comptes utilisateurs daemon et luke existent et, si oui, quels sont leurs uid, gid et leur(s) groupe(s) ?



La commande « id » suivi du nom permet d'afficher les identités d'un compte.

UID : numéro identifiant l'utilisateur (User IDentification) ;

GID : numéro identifiant le groupe principal de l'utilisateur ;

L'utilisateur « luke » n'existe pas, alors que l'utilisateur « daemon » existe et possède :

-uid=1 (daemon)

-gid=1 (daemon)

-groupes=1 (daemon)

2. Créez les groupes jedi et rebelles

```
root@DEB12Server: ~#groupadd jedi
root@DEB12Server: ~#groupadd rebelles
root@DEB12Server: ~#_
```

La commande **« groupadd »** suivi du nom permet de créer un groupe, ici se sont les groupes « **jedi** » et le groupe **« rebelles »**

3. Consultez le manuel en ligne afin de découvrir les options de la commande useradd.

```
root@DEB12Server: ~#man
Quelle page de manuel voulez-vous ?
Par exemple, essayez « man man ».
root@DEB12Server: ~#useradd
```

On utilise la commande **« man »** pour pouvoir accéder au manuel, puis on demande à avoir les options de la commande **« useradd »**

Puis, nous avons accès à toutes les options de cette commande.

```
Options :
      --badname                 do not check for bad names
 -b, --base-dir BASE_DIR        répertoire de base pour le répertoire home du
                                nouveau compte
      --btrfs-subvolume-home    utiliser le sous-volume BTRFS comme répertoire home
  -c, --comment COMMENT         champ GECOS du nouveau compte
  -d, --home-dir HOME_DIR       répertoire home du nouveau compte
 -D, --defaults             afficher ou changer la configuration de useradd par défaut
  -e, --expiredate EXPIRE_DATE  date d'expiration du nouveau compte
  -f, --inactive INACTIVE       période d'inactivité du mot de passe pour le nouveau compte
  -F, --add-subids-for-system   add entries to sub[ud]id even when adding a system user
  -g, --gid GROUP               nom ou IDentifiant du groupe primaire du nouveau
                               compte
  -G, --groups GROUPS            liste des groupes supplémentaires du nouveau
                                compte
  -h, --help                   afficher ce message d'aide et quitter
  -k, --skel SKEL_DIR           utiliser ce répertoire skeleton alternatif
  -K, --key KEY=VALUE          remplacer les valeurs par défaut de /etc/login.defs
  -l, --no-log-init             ne pas ajouter l'utilisateur aux bases de données lastlog
                                  et faillog
  -m, --create-home             créer le répertoire home de l'utilisateur
  -M, --no-create-home          ne pas créer de répertoire home de l'utilisateur
  -N, --no-user-group           ne pas créer de groupe avec le même nom que
                               celui de l'utilisateur
  -o, --non-unique             autoriser la création d'utilisateurs avec des
                                 UID dupliqués (non-uniques)
  -p, --password PASSWORD       mot de passe chiffré du nouveau compte
  -r, --system                  créer un compte système
 -R, --root CHROOT_DIR          répertoire dans lequel faire un chroot
  -P, --prefix PREFIX_DIR       préfixe du répertoire où sont situés les fichiers etc/*
 -s, --shell SHELL               interpréteur de commandes de connexion du nouveau compte
 -u, --uid UID                  IDentifiant utilisateur du nouveau compte
 -U, --user-group               créer un groupe avec le même nom que celui de l'utilisateur
 -Z, --selinux-user SEUSER      utiliser un SEUSER spécifique pour le mappage de l'utilisateur SELin
ux
```

4. Créez des comptes utilisateurs luke, vador et solo. Visualisez-les ensuite.

Le compte **luke** appartient au groupe **jedi** (comme groupe principal) et au groupe rebelles (comme groupe secondaire). Le compte **vador** appartient au groupe **jedi**. Le compte solo fait partie du groupe rebelles.

```
root@DEB12Server: ~#useradd -g jedi -G rebelles -m luke
root@DEB12Server: ~#useradd -g jedi -m vador
root@DEB12Server: ~#useradd -g rebelles -m solo
root@DEB12Server: ~#id luke
uid=1002(luke) gid=1002(jedi) groupes=1002(jedi),1003(rebelles)
root@DEB12Server: ~#id vador
uid=1003(vador) gid=1002(jedi) groupes=1002(jedi)
root@DEB12Server: ~#id solo
uid=1004(solo) gid=1003(rebelles) groupes=1003(rebelles)
root@DEB12Server: ~#
```

La commande **« useradd »**, nous permet de créer des comptes utilisateurs.

Lorsque la commande **« useradd »** est suivit de :

- **« -g »**, cela correspond au groupe principal

- **« -G »** correspond au groupe secondaire

- **« -m »**, permet la création du répertoire personnel

Puis nous vérifions si les comptes utilisateurs ont bien été créer à l'aide de la commande **« id »** suivit du **nom d'utilisateur** , ce qui nous permet de voir son uid,gid et son groupe.

5. Affichez les dernières lignes des fichiers /etc/passwd et /etc/group

D'abord, nous vérifions les 3 dernières lignes (à l'aide de la commande **«tail -3 »**) du fichier **/etc/passwd.**

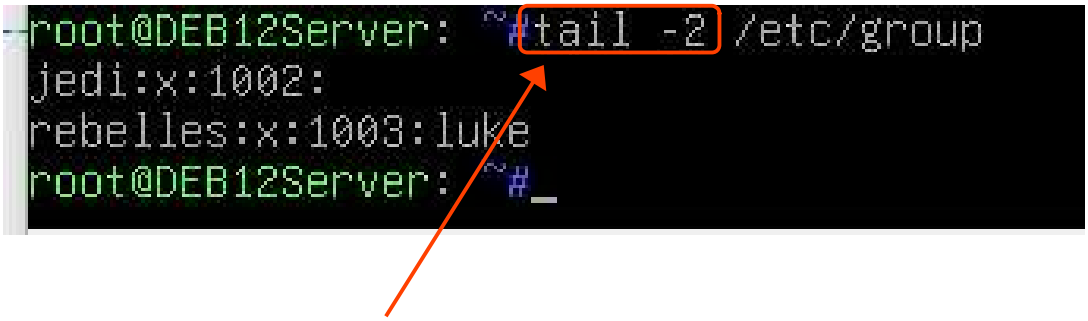
/etc/passwd : ce fichier contient la base locale des comptes utilisateurs.

/etc/group : ce fichier contient la base locale des comptes groupe.

```
root@DEB12Server: ~#tail -3 /etc/passwd
luke:x:1002:1002::/home/luke:/bin/sh
vador:x:1003:1002::/home/vador:/bin/sh
solo:x:1004:1003::/home/solo:/bin/sh
root@DEB12Server: ~#
```

On constate que les 3 dernières lignes sont nos 3 utilisateurs, ils sont suivit de leur uid, gid et leurs répertoires personnels.

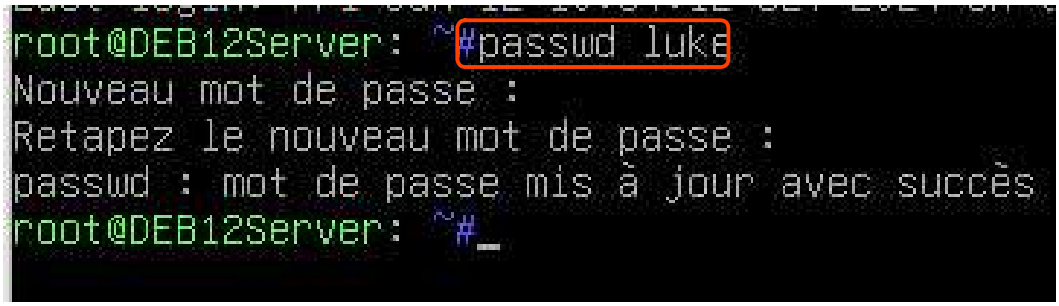
Puis, on fait de nouveau cette étape pour le fichier **/etc/group**



Ici, nous allons voir les 2 dernières lignes.

Et nous pouvons constater que le groupe **« jedi »** a comme gid :1002 et le groupe **« rebelles »** possède comme gid:1003 et luke possède comme groupe secondaire **« rebelles »** d'où plus tôt l'option **« -G »** de la commande **« useradd » .**

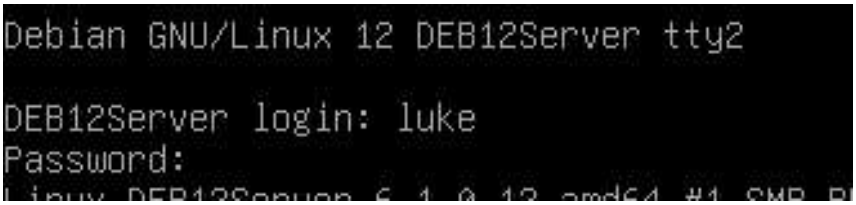
6. Mettez le mot « password » comme mot de passe à l'utilisateur luke.



La commande **« passwd »** permet de changer le mot de passe de l'utilisateur, qui est dans notre cas luke.

7. Ouvrez une seconde console (Ctrl+Alt+F2) et connectez-vous sous le compte de luke. Considérez le prompt ($).

Dans un premier temps, on se connecte en tant que luke à l'aide de mot de passe : « password » que l'on avait configurer plus haut.



```
Debian GNU/Linux 12 DEB12Server tty2

DEB12Server login: luke
Password:
Linux DEB12Server 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
$
```

8. Déconnectez-vous et retournez sur la première console. Modifiez le compte utilisateur luke afin de remplacer le shell sh par bash :

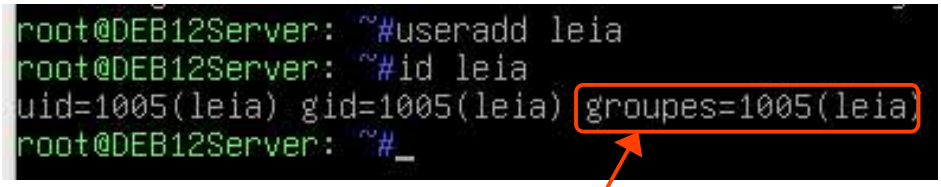
La commande **« usermod »,** permet de modifier un compte utilisateur. Ici cela nous permet de modifier le compte utilisateur luke afin de remplacer le **shell sh** par **bash.**

9. Reconnectez-vous sous le compte de luke dans la seconde console. Observez de nouveau le prompt.

```
Debian GNU/Linux 12 DEB12Server tty2

DEB12Server login: luke
Password:
Linux DEB12Server 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Jan 12 12:18:03 CET 2024 on tty2
luke@DEB12Server:~$ id
uid=1002(luke) gid=1002(jedi) groupes=1002(jedi),1003(rebelles)
luke@DEB12Server:~$
```

Remarque : la commande **« id »**sans argument indique l'identité de l'utilisateur courant.

10. Créez l'utilisateur **leia** dans la première console avec la commande **useradd** (sous le **compte root**). Quel est son groupe principal ?

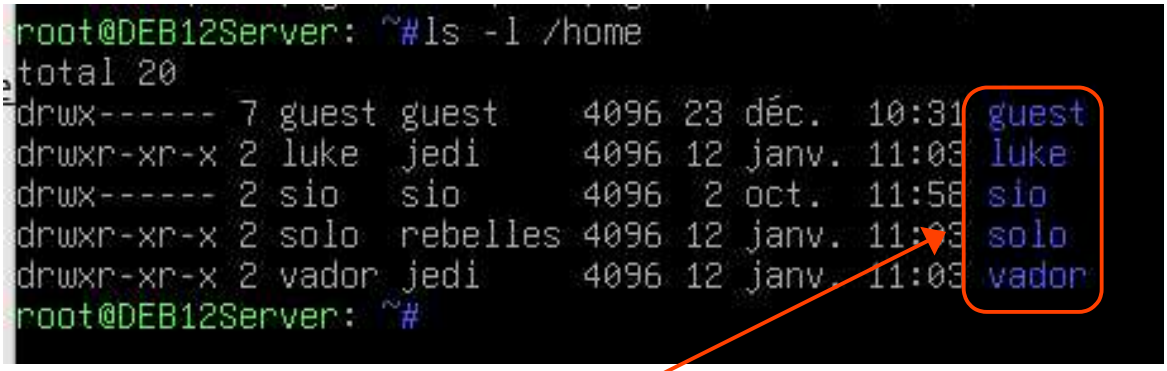


La commande **« useradd »** permet **l'ajout d'un compte utilisateur** mais, cette commande **ne permet pas de créer un répertoire personnelle** car le paramètre **« -m »** permet la création du répertoire personnel

Son groupe principal ce nomme « leia »

Remarque : par défaut, la création d'un compte utilisateur entraîne la création d'un compte groupe de **même nom** qui correspond au **groupe principal du nouvel utilisateur**.

11. Est-ce que le répertoire personnel de l'utilisateur leia a été créé ?



On peut constater que les users : guest, luke, sio, solo et vador possèdent tous un groupe et un répertoire personnel (en bleu).

Prenons l'exemple de luke : « **luke** » est son nom d'utilisateur, « **jedi** » est son groupe principal et « **luke** »(en bleu) est son répertoire personnel

Mais, on ne voit pas l'utilisateur « leia » car elle ne possède pas de répertoire personnel car lors de la création de l'utilisateur, on n'a pas spécifier la création du répertoire personnel

12. Gérez les groupes secondaires.

a) Affectez l'utilisateur leia au groupe rebelles (comme groupe secondaire)

```
root@DEB12Server: ~#usermod -G rebelles leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1003(rebelles)
root@DEB12Server: ~#
```

Remarque : La commande **« usermod »** permet la **modification d'un compte utilisateur**.

L'option **« -G »** correspond à la création d'un groupe secondaire.

b) Affectez **leia** au groupe **jedi**. **Leia** quitte le groupe rebelles.

```
root@DEB12Server: ~#usermod -G jedi leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1002(jedi)
root@DEB12Server: ~#
```

c) Affectez leia aux groupes jedi et rebelles.

```
root@DEB12Server: ~#usermod -G jedi,rebelles leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1002(jedi),1003(rebelles)
root@DEB12Server: ~#
```

La virgule entre **jedi** et **rebelles** permet de pouvoir adhérer à deux groupes, ce qui permet à leia d'être a la fois dans le groupe **jedi** et dans le groupe **rebelles**

d) On veut que leia n'appartienne plus à aucun groupe secondaire.

```
root@DEB12Server: ~#usermod -G "" leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia)
root@DEB12Server: ~#
```

le double guillemet **« "" »** comme il n'y aucun caractère entre ces deux guillemets alors, on ne lui affecte aucun groupe.

e) On veut ajouter l'utilisateur à un groupe secondaire sans le retirer des autres groupes secondaires (option -a) .

```
root@DEB12Server: ~#usermod -G jedi leia
root@DEB12Server: ~#usermod -aG rebelles leia
root@DEB12Server: ~#id leia
uid=1005(leia) gid=1005(leia) groupes=1005(leia),1002(jedi),1003(rebelles)
root@DEB12Server: ~#_
```

13. Supprimez le compte leia.

```
root@DEB12Server: ~#userdel leia
root@DEB12Server: ~#_
```

La commande « **userdel** » suivit du nom de l'utilisateur permet la suppression

14. Recréez le compte leia avec cette fois-ci un répertoire de connexion. A partir du compte leia, créez un répertoire ainsi qu'un fichier.

```
root@DEB12Server: ~#useradd -m leia
root@DEB12Server: ~#cd /home/leia
root@DEB12Server: /home/leia# su - leia
$ mkdir rep1
$ cd rep1
$ touch fichier1
$ ls -l
total 0
-rw-r--r-- 1 leia leia 0 15 janv. 20:46 fichier1
$ exit
root@DEB12Server: /home/leia#cd
root@DEB12Server: ~#_
```

Remarques : Les commandes :

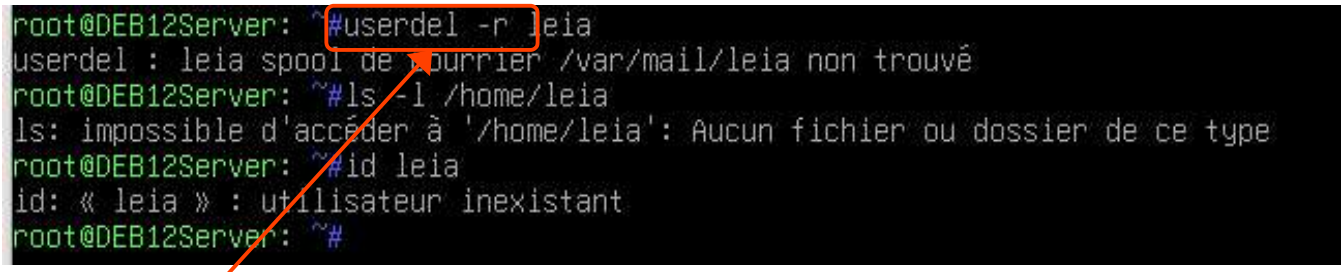
-**« useradd -m »** permet de créer un utilisateur ainsi que son répertoire personnel.

- **« cd »** (change directory), ça permet de changer de répertoire

-**« mkdir »** permet de créer un répertoire

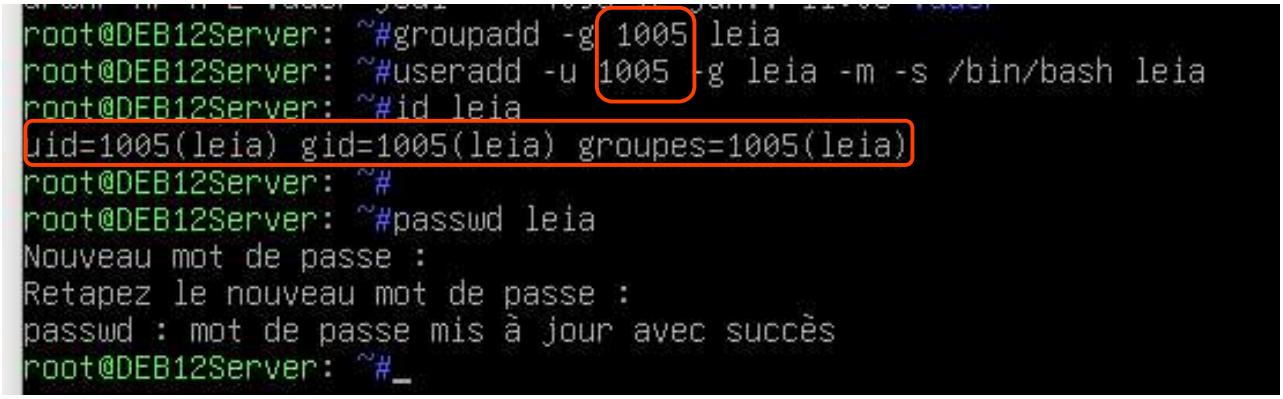
- **« touch »** permet de créer un fichier

15. Supprimez un compte utilisateur et les fichiers de son répertoire de connexion.

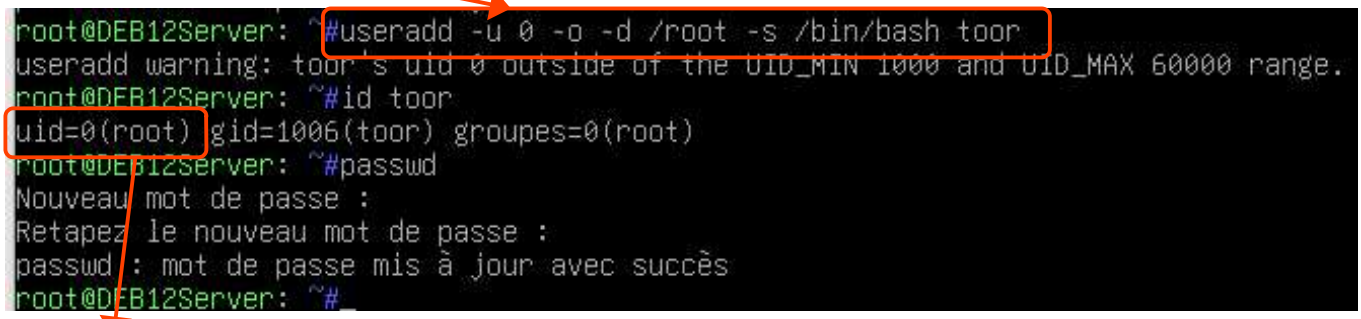


Permet de supprimer un utilisateur et l'option **« -r »** permet de retirer son répertoire personnel.

16. On veut recréer le compte leia à l'identique (mêmes uid et gid).



17. Créez le compte toor ayant les mêmes droits que root.



On voit que **toor** possède le même **uid** et le même groupe que **root.**

18. Ouvrez une seconde console et connectez-vous avec le compte toor.

```
DEB12Server login: toor
Password:
Linux DEB12Server 6.1.0-12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.52-1 (2023-09-07) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jan 15 18:39:30 CET 2024 on tty1
root@DEB12Server: ~#_
```

19. Créez un compte d'utilisateur respectant la charte Debian avec la commande adduser.

```
root@DEB12Server: ~#adduser palpatine
Ajout de l'utilisateur « palpatine » ...
Ajout du nouveau groupe « palpatine » (1007) ...
Ajout du nouvel utilisateur « palpatine » (1007) avec le groupe « palpatine » (1007) ...
Création du répertoire personnel « /home/palpatine » ...
Copie des fichiers depuis « /etc/skel » ...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mot de passe mis à jour avec succès
Modifier les informations associées à un utilisateur pour palpatine
Entrer la nouvelle valeur, ou appuyer sur ENTER pour la valeur par défaut
        NOM []:
        Numéro de chambre []:
        Téléphone professionnel []:
        Téléphone personnel []:
        Autre []:
Cette information est-elle correcte ? [O/n]O
Ajout du nouvel utilisateur « palpatine » aux groupes supplémentaires « users » ...
Ajout de l'utilisateur « palpatine » au groupe « users » ...
root@DEB12Server: ~#id palpatine
uid=1007(palpatine) gid=1007(palpatine) groupes=1007(palpatine),100(users)
root@DEB12Server: ~#_
```

La commande « **adduser** » permet de **créer un utilisateur de manière simplifier**, ce qui veut dire que cela **créer automatiquement son répertoire personnel**.

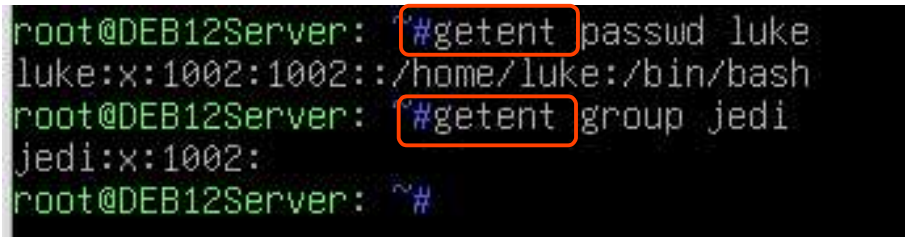
20. Affichez les caractéristiques de l'utilisateur local luke et du groupe local rebelles.

```
root@DEB12Server: ~#grep luke /etc/passwd
luke:x:1002:1002::/home/luke:/bin/bash
root@DEB12Server: ~#grep rebelles /etc/group
rebelles:x:1003:luke
root@DEB12Server: ~#
```

La commande « grep » permet de rechercher un chaîne dans un fichier

L'utilisateur **luke** appartient au groupe secondaire **rebelles**.

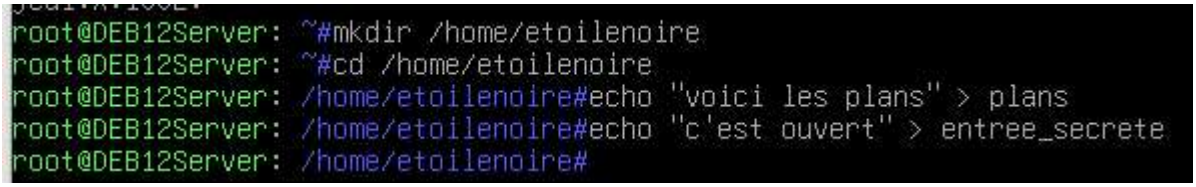
21. Affichez les caractéristiques de l'utilisateur luke et du groupe jedi.



Remarque : la commande « **getent »** permet d'interroger les annuaires locaux et réseaux (si ces derniers sont définis).

2. La gestion des droits.

1. Création d'une arborescence de fichiers.



À l'aide de la commande « **mkdir** », on peut créer le répertoire **etoilenoire** dans **home**

Puis, la commande « cd » nous permet de changer de répertoire. Ici, on veut se trouver dans le répertoire **« /home/etoilenoire »**

À l'aide de la commande « **echo** » et le **« > »** on crée **deux fichiers** : **plans** (qui contiendra « voici les plans » et **entree_secrete** (qui contiendra « c'est ouvert »)

2. Changement des caractéristiques du répertoire etoilenoire.

Son propriétaire sera **luke**, son groupe **jedi**. Il sera accessible en lecture, écriture et accès au propriétaire. Il sera accessible en lecture et accès au groupe mais pas aux autres.

Remarque : -L'option **« -d » précise le répertoire** lui-même et non pas le contenu

-La commande **« chmod »** permet de **modifier les droits** d'un fichier.

-La commande **« chgrp »** permet de **changer le groupe** d'un fichier.

-La commande **« chown »** permet **changer le propriétaire** d'un fichier.

3. Changement des caractéristiques des fichiers.

Ils seront accessibles en lecture seule pour le groupe et n'auront aucun droit pour les autres. On utilise la notation symbolique. On affilie le fichier **plans** au groupe **jedi** et le fichier **entree_secrete** au groupe **rebelles**.

À l'aide de la commande « chgrp », les fichiers **entree_secrete** et **plans** changent de groupe.

Et la commande **« chmod » modifie les droits** de tous les fichiers caracatérisé par **« * »** se trouvant dans le répertoire **/home/etoilenoire**

4. Test des accès.

a) A partir du compte **luke** :

L’utilisateur **luke**, en tant que propriétaire, a tous les droits sur le répertoire **etoilenoire** : il peut le lister, créer ou supprimer des fichiers dedans et il a accès aux fichiers qu’il contient. En tant que membre du groupe jedi, il peut lire le fichier **plans**, et en tant que membre du groupe **rebelles**, il peut lire le fichier **entree_secrete**. Par contre, il ne peut pas modifier le fichier **plans** (ainsi que le fichier **entree_secrete**). Seul **root** peut le faire.

Remarque :

-La commande **« su »** permet d**’éxécuter une commande** avec un **identifiant utilisateur** (uid) et un **identifiant de groupe** (gid) de **substitution**.

-La commande **« cat »** permet d’afficher le contenu d’un fichier.

-La commande **« rm »** permet de supprimer un fichier.

b) A partir du compte vador :

L’utilisateur **vador**, en tant que membre du groupe **jedi**, peut lister le répertoire **etoilenoire**. Il a accès également aux fichiers qu’il contient. Par contre, il ne peut ni créer ni supprimer des fichiers dedans. En tant que membre du groupe **jedi**, il peut lire le fichier plans mais pas le fichier **entree_secrete**. Il ne peut pas modifier le fichier plans. Seul **root** peut le faire.

```
root@DEB12Server: ~#su - vador
$ ls /home/etoilenoire
entree_secrete  plans
$ rm /home/etoilenoire/plans
rm : supprimer '/home/etoilenoire/plans' qui est protégé en écriture et est du type « fichier » ? y
rm: impossible de supprimer '/home/etoilenoire/plans': Permission non accordée
$ cal > /home/etoilenoire/fichier
-sh: 3: cannot create /home/etoilenoire/fichier: Permission denied
$ cat /home/etoilenoire/plans
voici les plans
$ cat /home/etoilenoire/entree_secrete
cat: /home/etoilenoire/entree_secrete: Permission non accordée
$ echo "===" >> /home/etoilenoire/plans
-sh: 6: cannot create /home/etoilenoire/plans: Permission denied
$ exit
```

c) A partir du compte solo :

L'utilisateur **solo** n'a aucun droit sur le répertoire **etoilenoire** : il ne peut pas connaître son contenu, il ne peut ni ajouter ni supprimer des fichiers à l'intérieur. Il n'a aucun accès aux fichiers de ce répertoire quels que soient leurs droits.

```
root@DEB12Server: ~#su - solo
$ ls /home/etoilenoire
ls: impossible d'ouvrir le répertoire '/home/etoilenoire': Permission non accordée
$ cal > /home/etoilenoire/fichier
-sh: 2: cannot create /home/etoilenoire/fichier: Permission denied
$ rm -f /home/etoilenoire/entree_secrete
rm: impossible de supprimer '/home/etoilenoire/entree_secrete': Permission non accordée
$ cat /home/etoilenoire/entree_secrete
cat: /home/etoilenoire/entree_secrete: Permission non accordée
$ exit
```

5. Suppression temporaire du droit d'exécution à la commande **uptime**.

Testez les conséquences à partir du compte **luke**.

```
root@DEB12Server: ~#whereis uptime
uptime: /usr/bin/uptime /usr/share/man/man1/uptime.1.gz
root@DEB12Server: ~#whatis uptime
uptime (1)           - Indiquer depuis quand le système a été mis en route
root@DEB12Server: ~#uptime
 19:31:44 up 41 min,  1 user,  load average: 0,00, 0,00, 0,00
root@DEB12Server: ~#ls -l /usr/bin/uptime
-rwxr-xr-x 1 root root 14648 19 déc.   2022 /usr/bin/uptime
root@DEB12Server: ~#chmod o-x /usr/bin/uptime
root@DEB12Server: ~#ls -l /usr/bin/uptime
-rwxr-xr-- 1 root root 14648 19 déc.   2022 /usr/bin/uptime
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ uptime
-bash: /usr/bin/uptime: Permission non accordée
luke@DEB12Server:~$ exit
```

```
root@DEB12Server: ~#chmod o+x /usr/bin/uptime
root@DEB12Server: ~#ls -l /usr/bin/uptime
-rwxr-xr-x 1 root root 14648 19 déc.   2022 /usr/bin/uptime
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ uptime
 19:35:32 up 44 min,  1 user,  load average: 0,08, 0,02, 0,01
luke@DEB12Server:~$ exit
```

Remarque :

L'option **« uptime »** indique depuis combien de temps le système a été mis en route

3. La gestion des droits, compléments.

1. Ajoutez des droits spéciaux au répertoire **etoilenoire** (SGID et sticky-bit).

Ensuite, pour vérifier l'impact de ces droits, on crée des fichiers dans le répertoire **etoilenoire**. Sous le compte **root**, on crée le fichier **f1**. Sous le compte **luke**, on crée le fichier **f2** et sous le compte **vador** on crée le fichier **f3**.

```
root@DEB12Server: ~#chmod 3770 /home/etoilenoire/
root@DEB12Server: ~#ls -ld /home/etoilenoire/
drwxrws--T 2 luke jedi 4096 16 janv. 18:57 /home/etoilenoire/
root@DEB12Server: ~#echo "fichier un" > /home/etoilenoire/f1
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ echo "bonjour" > /home/etoilenoire/f2
luke@DEB12Server:~$ exit
```

```
root@DEB12Server: ~#su - vador
$ echo "bonjour" > /home/etoilenoire/f3
$ exit
root@DEB12Server: ~#ls -l /home/etoilenoire/f?
-rw-r--r-- 1 root  jedi 11 16 janv. 19:40 /home/etoilenoire/f1
-rw-r--r-- 1 luke  jedi  8 16 janv. 19:41 /home/etoilenoire/f2
-rw-r--r-- 1 vador jedi  8 16 janv. 19:44 /home/etoilenoire/f3
root@DEB12Server: ~#
```

2. Vador va essayer de détruire le fichier de luke.

a) On conserve le droit sticky-bit.

```
root@DEB12Server: ~#su - vador
$ rm /home/etoilenoire/f2
rm : supprimer '/home/etoilenoire/f2' qui est protégé en écriture et est du type « fichier » ? y
rm: impossible de supprimer '/home/etoilenoire/f2': Opération non permise
$ exit
root@DEB12Server: ~#
```

b) On supprime le droit sticky-bit.

```
root@DEB12Server: ~#chmod -t /home/etoilenoire/
root@DEB12Server: ~#ls -ld /home/etoilenoire/
drwxrws--- 2 luke jedi 4096 16 janv. 19:44 /home/etoilenoire/
root@DEB12Server: ~#su - vador
$ rm /home/etoilenoire/f2
rm : supprimer '/home/etoilenoire/f2' qui est protégé en écriture et est du type « fichier » ? y
$ ls -l /home/etoilenoire/f2
ls: impossible d'accéder à '/home/etoilenoire/f2': Aucun fichier ou dossier de ce type
$ exit
root@DEB12Server: ~#
```

3. Qui peut formater la partition /dev/sda1 ?

```
root@DEB12Server: ~#ls -l /dev/sda1
brw-rw---- 1 root disk 8, 1 16 janv. 18:50 /dev/sda1
root@DEB12Server: ~#_
```

Seuls root et les membres du groupe disk peuvent formater cette partition.

4. L'administrateur copie les fichiers du répertoire etoilenoire dans /tmp en conservant leurs attributs.

```
root@DEB12Server: ~#cp -p /home/etoilenoire/* /tmp
root@DEB12Server: ~#ls -l /tmp/plans /tmp/entree_secrete
-rw-r----- 1 root rebelles 13 15 janv. 22:51 /tmp/entree_secrete
-rw-r----- 1 root jedi     16 15 janv. 22:50 /tmp/plans
root@DEB12Server: ~#
```

Remarque : normalement, quand on copie un fichier, la copie appartient à celui qui copie. Ici, **rebelles** et **jedi** restent les groupes auxquels sont affiliés respectivement les fichiers **entree_secrete** et **plans**. Sans l'option **-p**, le groupe **root** serait le groupe propriétaire.

5. L'administrateur donne le fichier entree_secrete à luke.

```
root@DEB12Server: ~#chown luke /tmp/entree_secrete
root@DEB12Server: ~#ls -l /tmp/entree_secrete
-rw-r----- 1 luke rebelles 13 15 janv. 22:51 /tmp/entree_secrete
root@DEB12Server: ~#_
```

6. Test des accès (r,w,x) au fichier /tmp/entree_secrete.

a) A partir du compte luke :

```
root@DEB12Server: ~#su - luke
luke@DEB12Server:~$ cat /tmp/entree_secrete
c'est ouvert
luke@DEB12Server:~$ echo "======" >> /tmp/entree_secrete
luke@DEB12Server:~$ cat /tmp/entree_secrete
c'est ouvert
======
luke@DEB12Server:~$ /tps/entree_secrete
-bash: /tps/entree_secrete: Aucun fichier ou dossier de ce type
luke@DEB12Server:~$
```

b) A partir du compte solo :

```
root@DEB12Server: ~#su - solo
$ cat /tmp/entree_secrete
c'est ouvert
======
$ echo "++++++" >> /tmp/entree_secrete
-sh: 2: cannot create /tmp/entree_secrete: Permission denied
$ exit_
```

c) A partir du compte root :

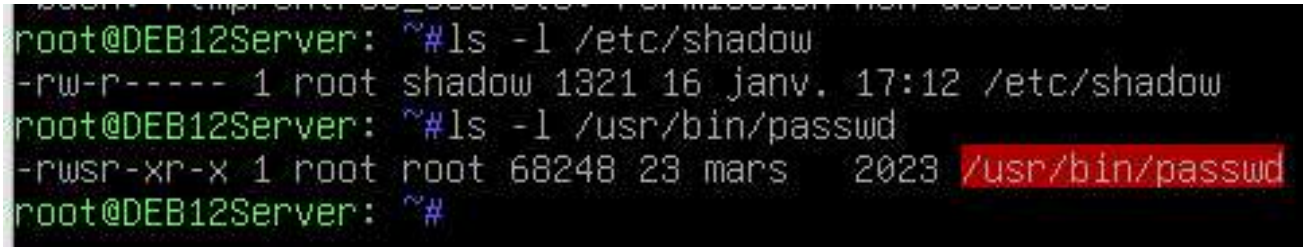
```
root@DEB12Server: ~#sysctl fs.protected_regular=0
fs.protected_regular = 0
root@DEB12Server: ~#cat /tmp/entree_secrete
c'est ouvert
======
root@DEB12Server: ~#echo "+=+=+=+=" >> /tmp/entree_secrete
root@DEB12Server: ~#cat /tmp/entree_secrete
c'est ouvert
======
+=+=+=+=
root@DEB12Server: ~# /tmp/entree_secrete
-bash: /tmp/entree_secrete: Permission non accordée
root@DEB12Server: ~#_
```

Remarques :

-les droits des utilisateurs ne s'appliquent pas à l'administrateur. Seule restriction, comme tout le monde, **root** a besoin du droit d'exécution pour activer une commande.

-Par défaut, pour des **raisons de sécurité**, **root** ne peut plus modifier le fichier **entree_secrete** dans le répertoire **« /tmp »** protégé par **sticky-bit** depuis une mise à jour de Linux. On peut modifier cela avec la commande **« sysctl fs.protected_regular=0 »**(2 par défaut).

7. Visualisez les droits du fichier shadow et de la commande passwd



Remarque : tout le monde a le droit d'exécuter la commande « **passwd** » mais pour le fichier « **shadow** » personne ne peut l'exécuter. Avec son **droit SUID**, un utilisateur endosse l**es droits de root**, ce qui lui permet d'accéder au fichier **shadow**.