TP Commandes Linux

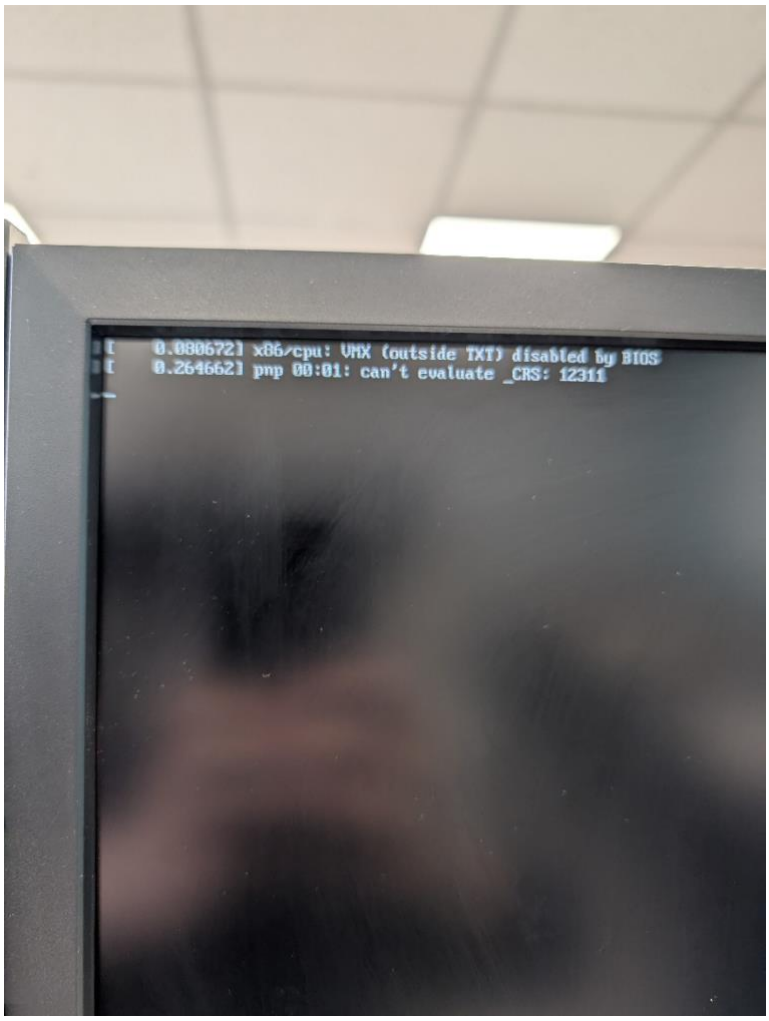
Objectif : Utilisation des commandes Linux pour gérer l'ensemble du système.
On pourrait ouvrir des consoles dans l'interface graphique, mais nous allons utiliser le mode console directement, comme si nous étions sur un système Linux sans interface graphique. C'est généralement le cas sur un serveur : aucun utilisateur n'est censé travailler dessus directement ; il n'a pas d'écran de clavier ou de souris ; les utilisateurs se connectent à distance...

Remarque : Utiliser la commande man pour trouver l'aide sur une instruction, sa syntaxe, et ses options. Faire ce TP avec le cours à côté peut aider à comprendre le cours...

I. Le mode console

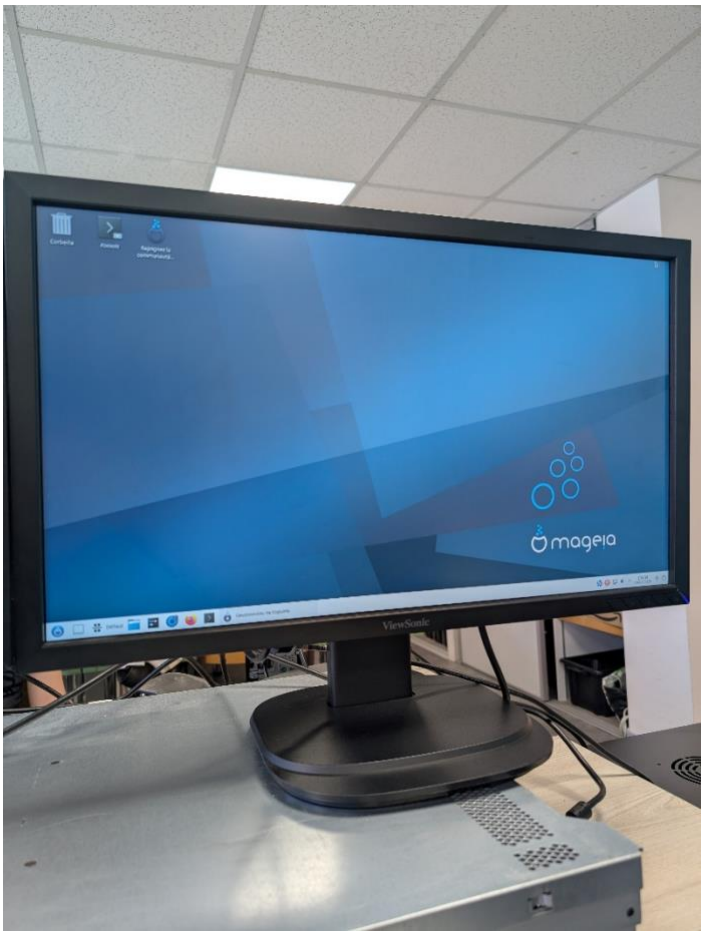
1. Au démarrage de votre machine, vous vous trouvez sur votre bureau ou sur l'écran de connexion des utilisateurs. Tapez Ctrl-Alt-F2 : vous êtes sur la console tty2 (teletypewriter). Tapez Alt-F3 : vous êtes sur la console tty3. Testez les consoles suivantes jusqu'à F12. Vous allez remarquer que certaines sont noires ou déjà utilisées par le système, le serveur graphique notamment. Alt-F1 vous ramène sur l'interface graphique et il faut rajouter la touche Ctrl pour repasser en mode console.

Console TTY1 :



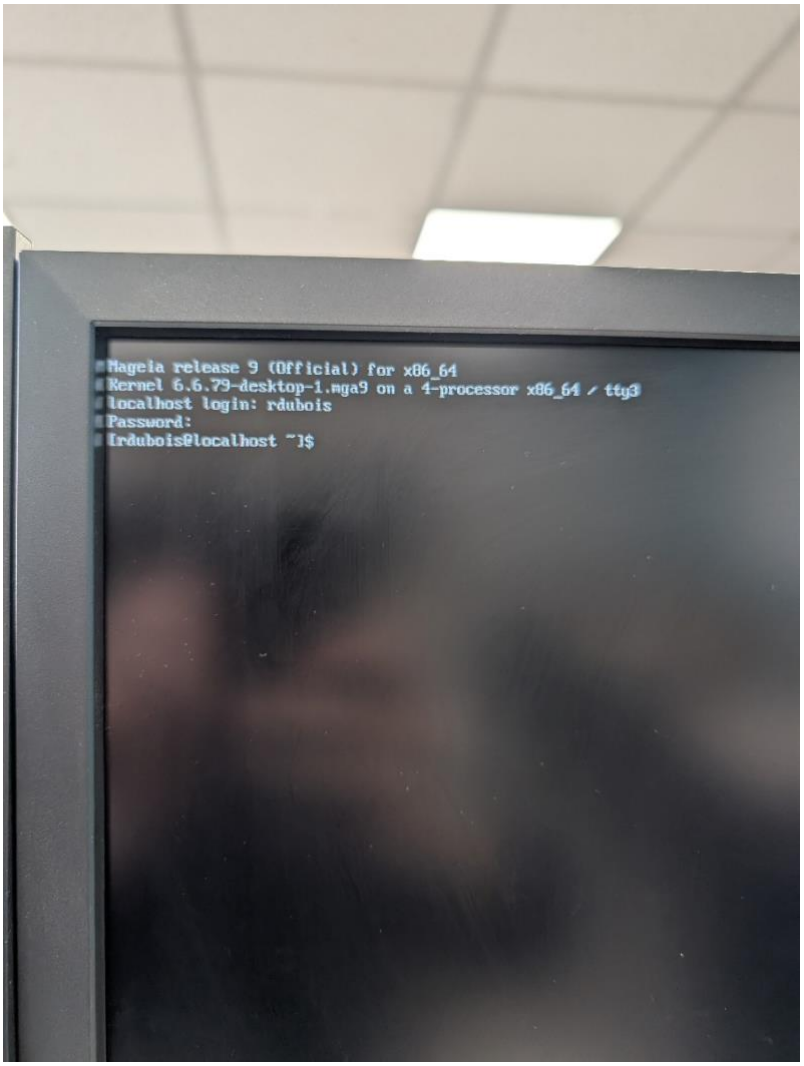
Cette console est une console réservée pour le démarrage du système Linux.

Console TTY2 :



Console réservée pour le serveur graphique et donc KDE.

Console TTY3 jusqu'à TTY6 :



Ces consoles sont réservées pour entrer des commandes Linux, avant de pouvoir exécuter une commande, on doit avant de pouvoir lancer des commandes se connecter à notre compte utilisateur.

2. Revenez sur la deuxième console, entrez votre identifiant utilisateur et votre mot de passe. Passez sur la troisième console et entrez l'identifiant du super-utilisateur *root* et le mot de passe administrateur. Comparez le *prompt* entre les deux consoles et notez la différence entre le mode utilisateur et le mode administrateur. Cela ne saute pas aux yeux !
Prompt en tant qu'utilisateur rdubois :

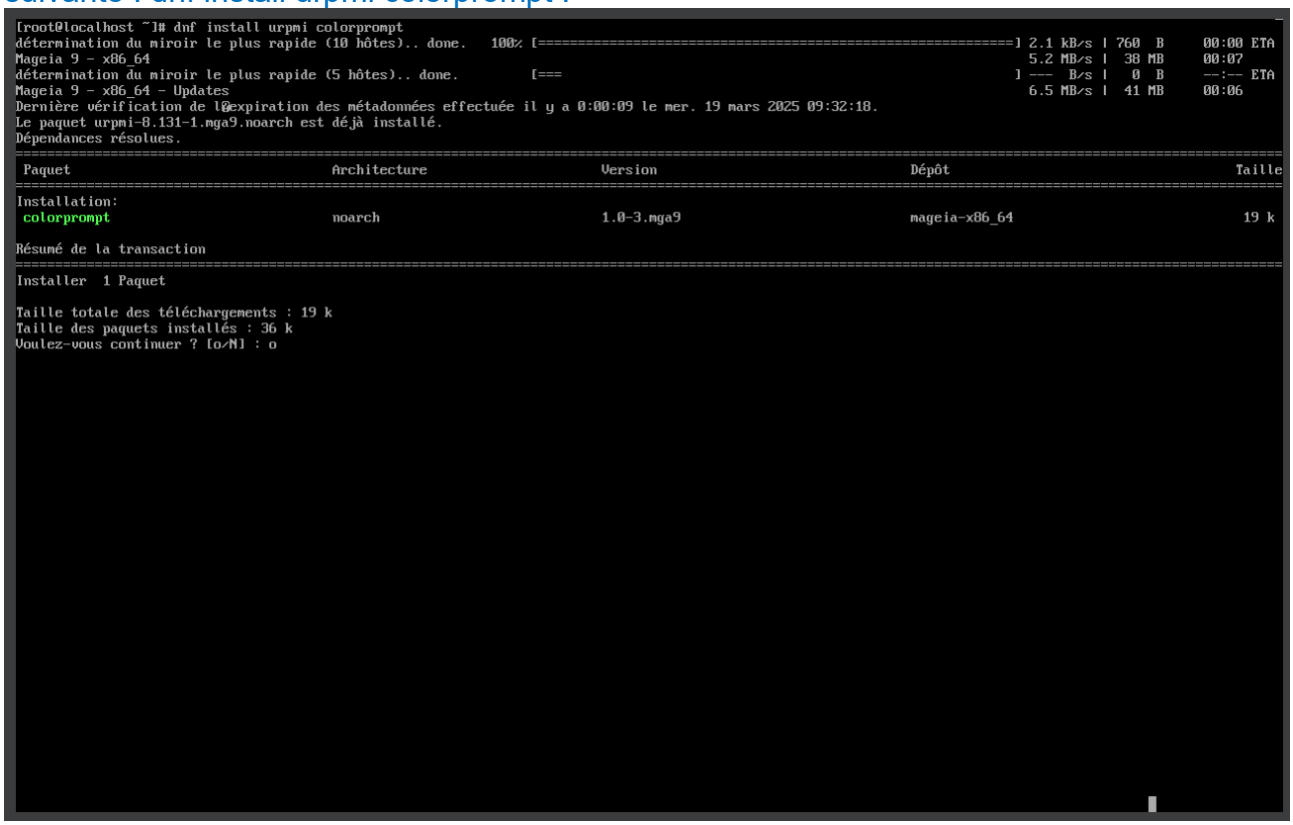
```
rdubois@DL5420:~$
```

Prompt en tant que root (Administrateur) :

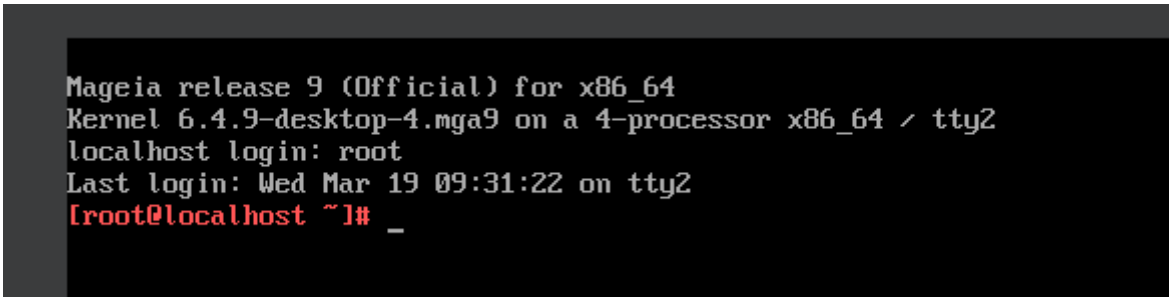
```
root@DL5420:~#
```

3. Pour améliorer la distinction, installons un petit paquet dans la console super-utilisateur : *urpmi colorprompt*. Déconnectez-vous avec la commande *exit* et reconnectez-vous. Vous voyez la différence ?

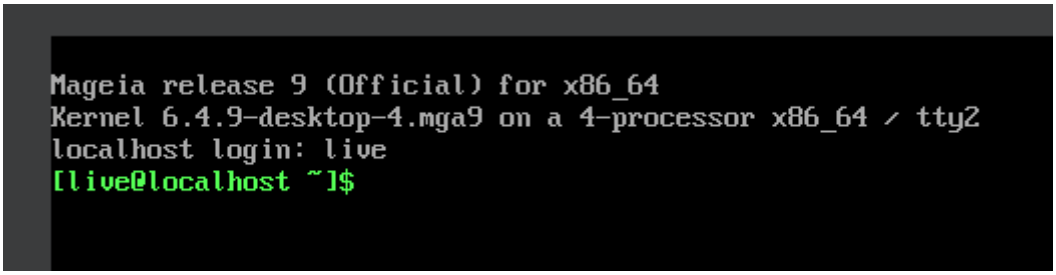
Pour installer un paquet dans la console super utilisateur, il suffit de lancer la commande suivante : dnf install urpmi colorprompt :



Après avoir installé ces paquets, il suffit de se déconnecter avec la commande « exit » et de se reconnecter dans le terminal, on observe bien que le prompt est devenu rouge en mode root :



Et vert en mode utilisateur :



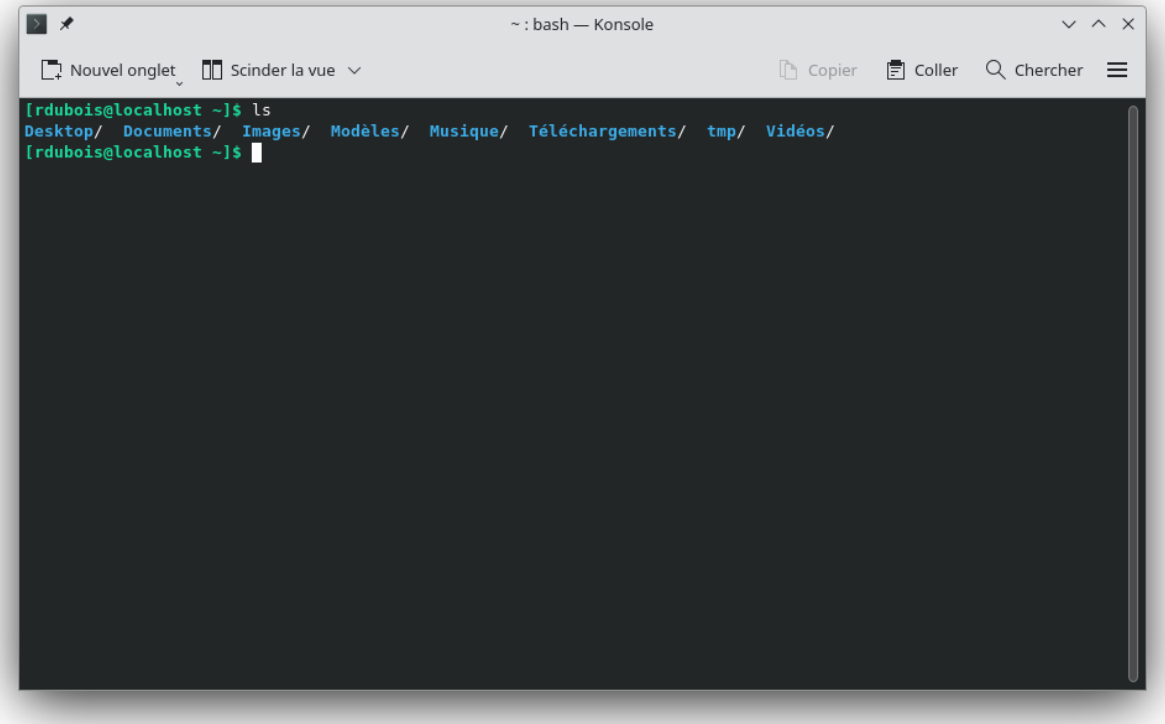
4. Déconnectez-vous et reconnectez-vous aussi dans la console utilisateur. Passez en super-utilisateur avec la commande *su*, puis revenez en utilisateur avec *exit*.

II. Arborescence

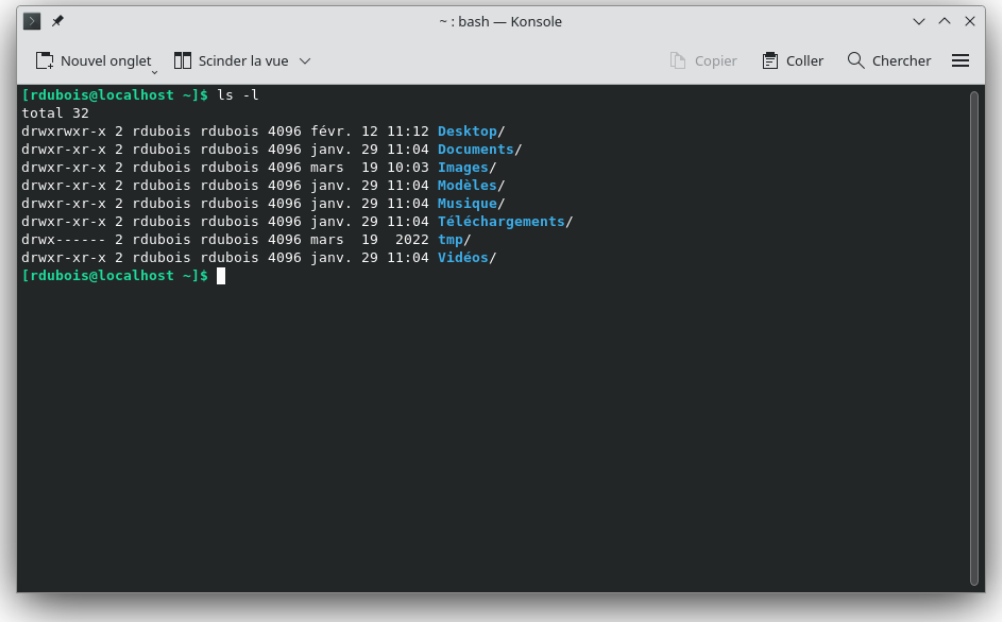
1. Dans la console utilisateur, tapez la commande *pwd* qui vous donne l'endroit où vous vous trouvez dans l'arborescence.
La commande pwd donne l'emplacement de mon dossier utilisateur étant donné que par défaut, le terminal se rend dans ce répertoire :

```
rdubois@DL5420:~$ pwd
/home/rdubois
rdubois@DL5420:~$
```

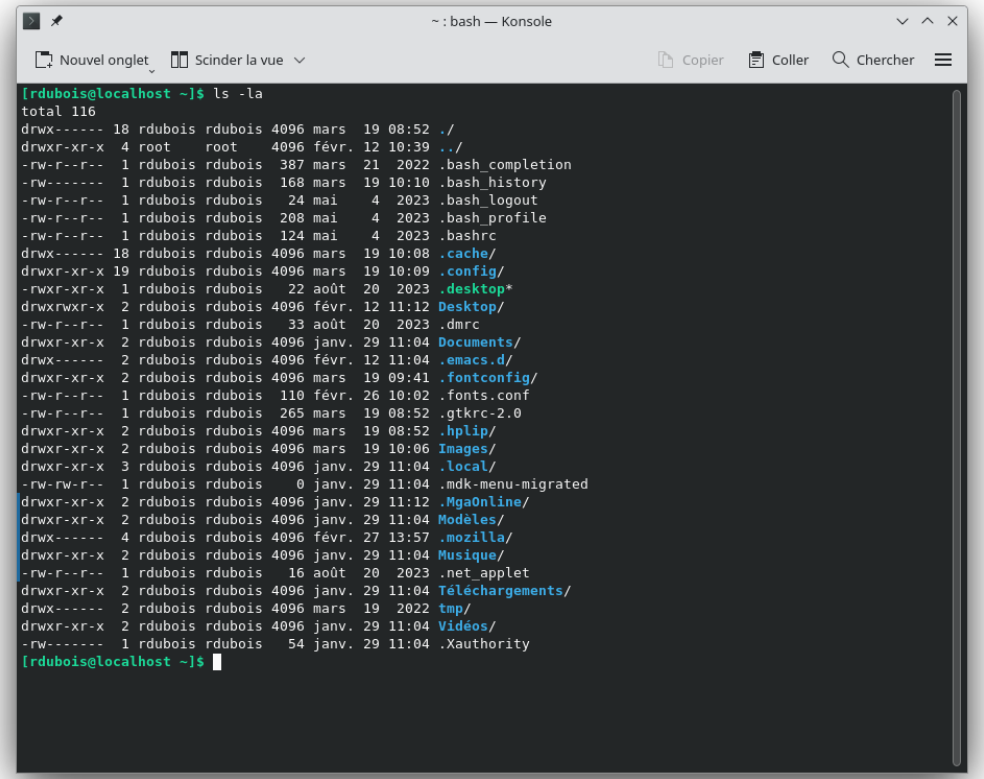
2. Tapez *ls* pour lister votre répertoire. Puis *ls -l* pour avoir un affichage au format long. Puis *ls -la* pour avoir un affichage long avec les fichiers cachés. Notez la présence des répertoires cachés qui commencent par un point. Tapez la commande *ll* : c'est un alias plus rapide de la commande *ls -la --color=auto* qui rajoute aussi l'option de coloration !
Résultat de la commande ls :



Résultat de la commande ls-l avec l'affichage long (on peut observer que nous avons plus d'informations sur les fichiers, notamment sur les permissions qu'ils ont et à qui ces derniers appartiennent) :



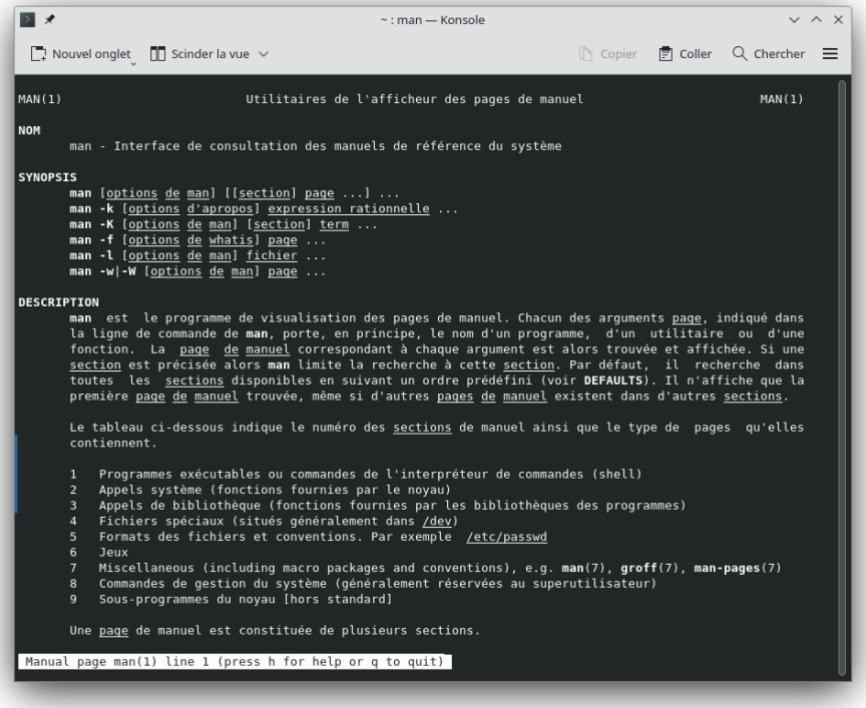
Résultat de la commande ls –la (on peut remarquer que plusieurs fichiers et dossiers cachés sont présents dans notre répertoire personnel, ces derniers sont représentés par un . puis le nom du fichier. Par exemple le fichier .bash_history est un fichier caché contenant l'historique des commandes que l'on a pu taper dans le terminal) :



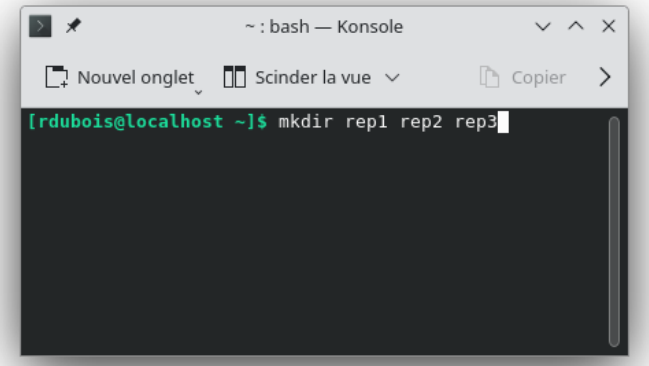
Résultat avec l'alias ll :

```
[rdubois@localhost ~]$ ll
total 32
drwxrwxr-x 2 rdubois rdubois 4096 mars  19 09:40 Desktop/
drwxr-xr-x 2 rdubois rdubois 4096 mars  19 09:30 Documents/
drwxr-xr-x 2 rdubois rdubois 4096 mars  19 09:30 Images/
drwxr-xr-x 2 rdubois rdubois 4096 mars  19 09:30 Modèles/
drwxr-xr-x 2 rdubois rdubois 4096 mars  19 09:30 Musique/
drwxr-xr-x 2 rdubois rdubois 4096 mars  19 09:30 Téléchargements/
drwx------ 2 rdubois rdubois 4096 mars  19  2022 tmp/
drwxr-xr-x 2 rdubois rdubois 4096 mars  19 09:30 Vidéos/
[rdubois@localhost ~]$
```

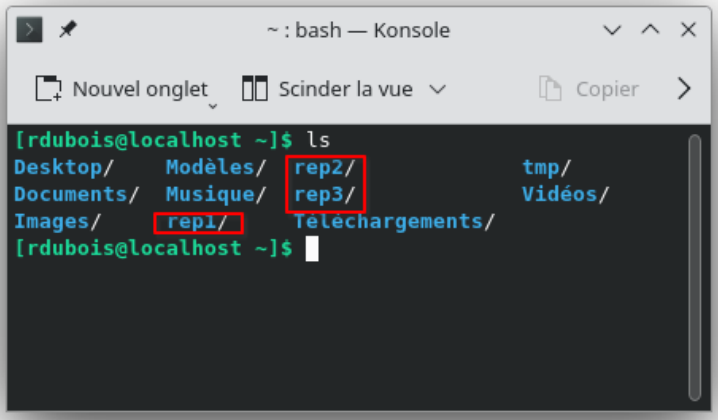
3. Tapez *man man* pour afficher la page de manuel de la commande man. Parcourez les options et tapez q pour quitter. Si c'est en anglais, installez les pages de manuel en français : dans la console administrateur, tapez *urpmi man-pages-fr* et vérifiez.
La commande "man man" fonctionne et permet d'afficher le manuel de cette commande. Ce dernier est déjà défini en français :



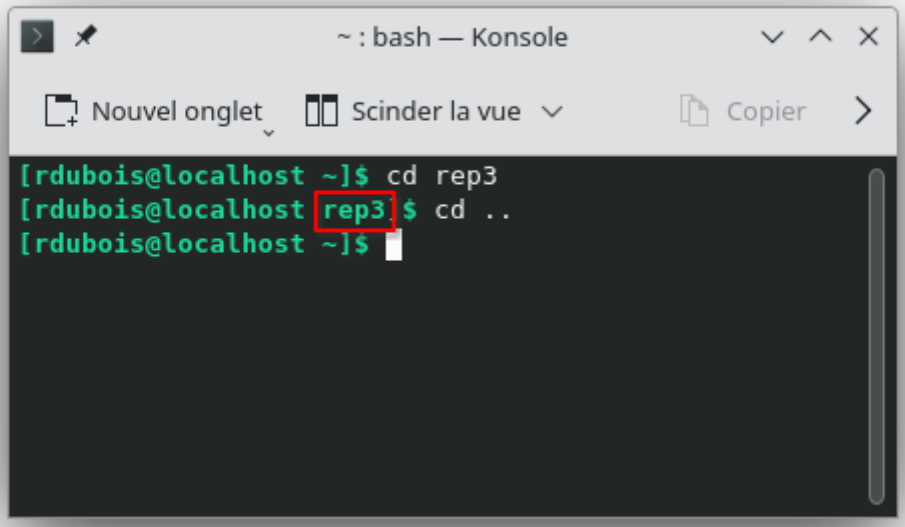
4. Dans la console utilisateur, utilisez la commande *mkdir* pour créer trois répertoires (rep1, rep2 et rep3). Utilisez la commande *ls* pour vérifier le résultat. Déplacez-vous avec la commande *cd* dans le répertoire rep3. Remontez dans le répertoire précédent. Utilisez la commande *rmdir* pour effacer le répertoire rep3 et vérifiez le résultat.
Commande mkdir pour créer les trois répertoires :



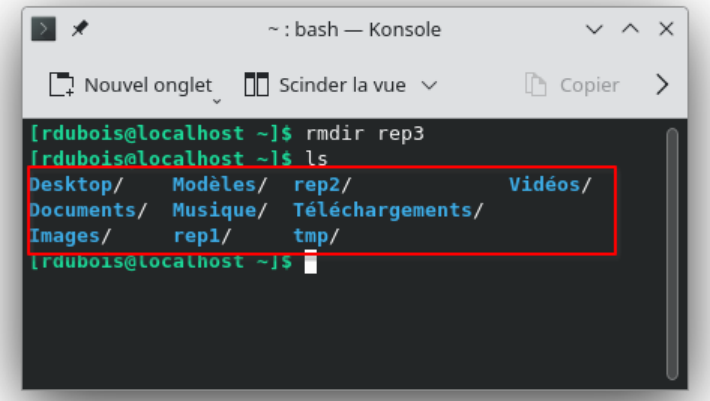
Commande ls utilisé pour vérifier le résultat (on peut observer que les dossiers ont bien été créer) :



Déplacement vers rep3 avec cd pour retour au répertoire précèdent (on peut observer dans le prompt que nous somme bien dans le répertoire rep3) :



Suppression du dossier rep3 avec la commande rmdir et vérification avec ls :



5. Déplacez vous dans rep2 et listez le répertoire avec *ll*. Copiez le fichier /etc/passwd dans ce répertoire avec la commande *cp* et faites une copie de ce fichier sous le nom passwd.old. Déplacez le fichier passwd.old dans le répertoire rep1 en le renommant en passwd en utilisant la commande *mv*.

Pour se déplacer dans le dossier rep2, il suffit de faire un « cd » dans ce dernier et de lancer la commande ll pour lister le contenu du répertoire :



Pour copier le fichier dans /etc/passwd, il suffit de faire un « cp /etc/passwd ./passwd »

/etc/passwd étant un chemin absolu
./passwd étant le chemin de destination relatif (étant donné que nous sommes déjà dans le répertoire rep2)

Pour faire une copie du fichier, il suffit juste d'exécuter la commande suivante « cp passwd passwd.old » qui va faire une copie du fichier en « passwd.old » :

```
[user@localhost rep2]$ cp /etc/passwd ./passwd
[user@localhost rep2]$ cp passwd passwd.old
[user@localhost rep2]$ ll
total 8
-rw-r--r-- 1 user user 2669 avril  2 08:37 passwd
-rw-r--r-- 1 user user 2669 avril  2 08:37 passwd.old
[user@localhost rep2]$
```

Pour déplacer le fichier avec la commande rep2 tout en restant dans le répertoire, on peut utiliser cette commande : « mv passwd.old ../rep1/ »

passwd.old étant le chemin relatif du fichier dans le répertoire rep2
../rep1/passwd étant le chemin relatif du dossier rep1 (en étant dans le dossier rep2)

```
[user@localhost rep2]$ mv passwd.old ../rep1/
[user@localhost rep2]$ cd ..
[user@localhost ~]$ cd rep1
[user@localhost rep1]$ ll
total 4
-rw-r--r-- 1 user user 2669 avril  2 08:24 passwd.old
[user@localhost rep1]$
```

6. Revenez dans votre répertoire personnel en tapant *cd*. Supprimez le répertoire rep2. Que se passe-t-il ? Essayez avec la commande *rm -rf rep2*. Que représentent les options r et f ? Pourquoi cette commande est-elle dangereuse ?
Si l'on essaie de supprimer le dossier « rep2 », on se rend compte qu'il n'est pas possible de le supprimer car il s'agit d'un dossier :

```
[user@localhost ~]$ rmdir rep2
rmdir: impossible de supprimer 'rep2': Le dossier n'est pas vide
[user@localhost ~]$
```

Si l'on souhaite pouvoir supprimer un dossier, il faut ajouter utiliser la commande rm avec

les paramètres -r qui permet de supprimer les répertoires et leur contenu récursivement. Et le paramètre -f qui permet de forcer la suppression sans demander une confirmation à l'utilisateur

```
[user@localhost ~]$ rm -rf rep2
[user@localhost ~]$ ll
total 36
drwxrwxr-x 2 user user 4096 avril  2 01:04 Desktop/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Documents/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Images/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Modèles/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Musique/
drwxr-xr-x 2 user user 4096 avril  2 08:29 rep1/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Téléchargements/
drwx------ 2 user user 4096 mars  19  2022 tmp/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Vidéos/
[user@localhost ~]$
```

Cette commande est cependant assez dangereuse car elle ne demande aucune confirmation à l'utilisateur avant la suppression et ce qui peut dans certains cas si un mauvais le chemin est mal indiqué, supprimé des fichiers non voulus ou endommager le système…

III : Administration des utilisateurs

1. Affichez le contenu du répertoire /home.

Commande utilisée : « ll /home »

```
[user@localhost ~]$ ll /home
total 20
lrwxrwxrwx  1 root root    10 avril  1 23:19 live -> /home/user/
drwx------  2 root root 16384 avril  2 01:02 lost+found/
drwx------ 18 user user  4096 avril  2 08:44 user/
[user@localhost ~]$
```

On peut observer qu'il y a dans ce dossier notre dossier utilisateur ainsi que certains dossiers système.

2. Utilisez la commande *adduser* pour créer un utilisateur util1.

Commande utilisée (en tant que root) : « adduser util1 »

```
[root@localhost user]# adduser util1
```

3. Qu'est-ce qui a changé dans le répertoire /home ?

Dans le répertoire /home, on peut observer qu'un nouveau dossier « util1 » a été créer, on peut observer que le propriétaire de ce dossier est util1 et qu'il appartient à ce dernier :

```
[root@localhost user]# ll /home
total 24
lrwxrwxrwx  1 root  root     10 avril  1 23:19 live -> /home/user/
drwx------  2 root  root  16384 avril  2 01:02 lost+found/
drwx------ 18 user  user   4096 avril  2 08:44 user/
drwx------  4 util1 util1  4096 avril  2 08:52 util1/
[root@localhost user]#
```

4. Avec la commande *cat*, affichez le contenu du fichier /etc/passwd et du fichier /etc/shadow.

Contenu du fichier /etc/passwd :

```
operator:x:11:0:operator:/var:/bin/sh
games:x:12:100:games:/usr/games:/bin/sh
nobody:x:65534:65534:Nobody:/:/bin/sh
messagebus:x:999:999:system user for dbus:/:/sbin/nologin
systemd-coredump:x:998:998:systemd Core Dumper:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
systemd-resolve:x:193:193:systemd Resolver:/:/sbin/nologin
systemd-oom:x:993:993:systemd Userspace OOM Killer:/:/usr/sbin/nologin
systemd-journal-remote:x:992:992:systemd Journal Remote:/:/usr/sbin/nologin
systemd-timesync:x:991:991:systemd Time Synchronization:/:/usr/sbin/nologin
polkitd:x:990:990:system user for polkit:/usr/lib/polkit-1:/sbin/nologin
rpm:x:989:989:system user for rpm:/var/lib/rpm:/bin/false
colord:x:988:988:system user for colord:/var/lib/colord:/usr/sbin/nologin
rtkit:x:987:987:system user for rtkit:/proc:/sbin/nologin
geoclue:x:986:986:system user for geoclue:/var/lib/geoclue:/sbin/nologin
pipewire:x:985:984:PipeWire System Daemon:/run/pipewire:/sbin/nologin
rpc:x:984:983:system user for rpcbind:/var/lib/rpcbind:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
openvpn:x:983:981:system user for openvpn:/var/lib/openvpn:/bin/true
nm-openvpn:x:982:980:system user for networkmanager-openvpn:/var/lib/openvpn:/bin/false
partimag:x:981:979:system user for partimage:/var/lib/partimage:/bin/false
saned:x:980:977:system user for saned:/etc/sane.d:/bin/false
vnstat:x:979:976:system user for vnstat:/var/lib/vnstat:/sbin/nologin
chrony:x:978:975:system user for chrony:/var/lib/chrony:/sbin/nologin
usbmux:x:977:974:system user for usbmuxd:/proc:/sbin/nologin
sddm:x:976:973:system user for sddm:/var/lib/sddm:/sbin/nologin
nm-openconnect:x:975:972:NetworkManager user for OpenConnect:/:/sbin/nologin
tss:x:971:971:tss user for tpm2:/:/usr/sbin/nologin
flatpak:x:970:970:User for flatpak system helper:/:/sbin/nologin
mysql:x:969:969:system user for mariadb:/var/lib/mysql:/bin/bash
rpcuser:x:968:968:system user for nfs-utils:/var/lib/nfs:/bin/false
avahi:x:967:967:system user for avahi:/var/run/avahi-daemon:/bin/false
avahi-autoipd:x:966:966:system user for avahi:/var/lib/avahi-autoipd:/bin/false
davfs2:x:965:962:system user for davfs2:/run/mount.davfs2:/bin/false
user:x:1000:1000:Utilisateur:/home/user:/bin/bash
sshd:x:964:958:system user for openssh:/usr/share/empty.sshd:/sbin/nologin
util1:x:1001:1001::/home/util1:/bin/bash
[root@localhost user]#
```

Ce fichier contient chaque utilisateur (et l'utilisateur util1 que nous avons créé précédemment) ainsi que leur configuration :

Explication des champs :

1. **nom_utilisateur** : Nom de l'utilisateur (ex. util1).
2. **x** : Anciennement le mot de passe crypté, maintenant déplacé vers /etc/shadow pour plus de sécurité.
3. **UID** : Identifiant unique de l'utilisateur (ex. 1001).
4. **GID** : Identifiant du groupe principal de l'utilisateur (ex. 1001).
5. **Description** : Champs facultatif, souvent utilisé pour le nom complet de l'utilisateur.
6. **Home** : Répertoire personnel de l'utilisateur (ex. /home/util1).
7. **Shell** : Interpréteur de commandes par défaut (ex. /bin/bash ou /bin/zsh).

Contenu du fichier /etc/shadow :

```
games:*:19072:0:99999:7:::
nobody:*:19072:0:99999:7:::
messagebus:!:19589::::::
systemd-coredump:!:19589::::::
systemd-network:!:19589::::::
systemd-resolve:!:19589::::::
systemd-oom:!*:19589::::::
systemd-journal-remote:!*:19589::::::
systemd-timesync:!*:19589::::::
polkitd:!:19589::::::
rpm:!:19589::::::
colord:!:19589::::::
rtkit:!:19589::::::
geoclue:!:19589::::::
pipewire:!:19589::::::
rpc:!:19589::::::
vcsa:!:19589::::::
openvpn:!:19589::::::
nm-openvpn:!:19589::::::
partimag:!:19589::::::
saned:!:19589::::::
vnstat:!:19589::::::
chrony:!:19589::::::
usbmux:!:19589::::::
sddm:!:19589::::::
nm-openconnect:!:19589::::::
tss:!*:19589::::::
flatpak:!:19589::::::
mysql:!:19589::::::
rpcuser:!:19589::::::
avahi:!:19589::::::
avahi-autoipd:!:19589::::::
davfs2:!:19589::::::
user:$6$F9crUe2SKAqS0j4Y$9T/aPVLdKTi5wO6q4Bf07s942S8.hxCqSG2muaGHXcxKtU1Jf0.0z0YzndhhAbHAONviH4fAc.HPsMWwogn.N0:20179:0:9999
9:7:::
sshd:!:20180::::::
util1:!:20180:0:99999:7:::
[root@localhost user]#
```

Explication des champs :

1. **nom_utilisateur** : Identique à celui de /etc/passwd.
2. **mot_de_passe_chiffré** : Mot de passe haché (ex. 6salt$hashedpassword pour SHA-512).
   - S'il contient juste ! ou *, le compte est verrouillé.
   - Si c'est vide (::), l'utilisateur n'a pas de mot de passe.
3. **dernier_changement** : Nombre de jours depuis le **1er janvier 1970** où le mot de passe a été modifié.

4. Définir avec la commande *passwd* un mot de passe pour util1. Vérifiez les changements dans /etc/shadow.

Commande utilisée : « passwd util1 »

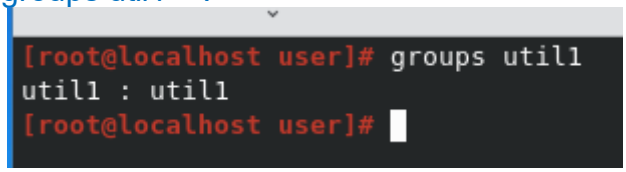
```
[root@localhost user]# passwd util1
Changement de mot de passe pour l'utilisateur util1.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd : mise à jour réussie de tous les jetons d'authentification.
[root@localhost user]#
```

Le fichier /etc/shadow contient maintenant le mot de passe de notre compte chiffré :

```
davfs2:!:19589::::::
user:$6$F9crUe2SKAqS0j4Y$9T/aPVLdKTi5wO6q4Bf07s942S8.hxCqSG2muaGHXcxKtU1Jf0.0z0YzndhhAbHAONviH4fAc.HPsMWwogn.N0:20179:0:9999
9:7:::
sshd:!:20180::::::
util1:$6$.dU1QzAQOm9s.elD$oKT4sJuIjWbOZ8KHMGmmTqUKxL09lZjyLgb5zPtxlPQ.wtLL9ny93KlVFGaePLfHO5Qh/39ZVv.YcXN85/vOU/:20180:0:999
99:7:::
[root@localhost user]#
```

6. Utilisez la commande *groups* pour afficher les groupes de util1. Quel est le nom du groupe par défaut ?

Commande utilisée : « groups util1 » :

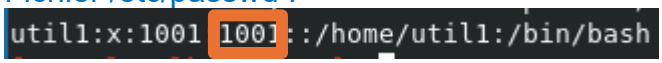


Le groupe par défaut de l'utilisateur « util1 » est le groupe « util1 ». En effet, si l'on vérifie le dossier /etc/group, on observe que le groupe « util1 » avec le GID « 1001 » a été créer, et si l'on observe le fichier /etc/passwd, on peut observer que ce GID par défaut est lui aussi défini sur 1001 :

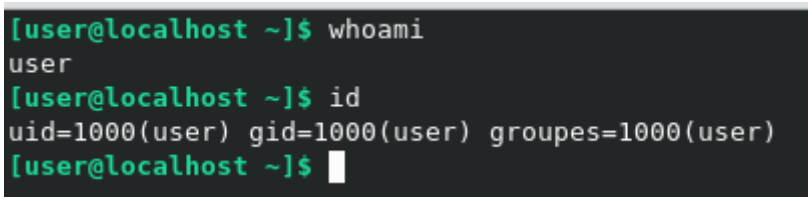
Fichier /etc/group :



Fichier /etc/passwd :



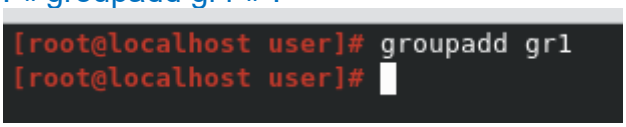
7. Utilisez les commandes *whoami* et *id*. Les informations affichées concernent quel compte ?



La commande whoami affiche le nom de l'utilisateur actuel, tandis que la commande id affiche plus de détails comme l'UID de l'utilisateur actuel, le GID et les différents groupes sur lequel l'utilisateur actuel est.

Cette commande concerne uniquement le compte qui est actuellement connecté dans le shell (en l'occurrence user).

8. Utilisez la commande *groupadd* pour créer un groupe gr1.
9. Commande utilisée : « groupadd gr1 » :



9. Trouvez les options de la commande *usermod* pour rajouter util1 dans le groupe gr1. Vérifiez le fichier /etc/group.

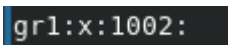
Pour ajouter l'utilisateur util1 au groupe gr1, on utilise la commande suivante :
usermod -aG gr1 util1

Explication des options :

- -a (**append**) : Ajoute l'utilisateur à un groupe **sans retirer** ses autres groupes. **(Important !)**
- -G (**groups**) : Spécifie-le ou les groupes secondaires à ajouter.
- **nom_du_groupe** : Spécifier le nom du groupe que l'on souhaite ajouter
- **nom_utilisateur** : Spécifier le nom de l'utilisateur que l'on souhaite ajouter au groupe

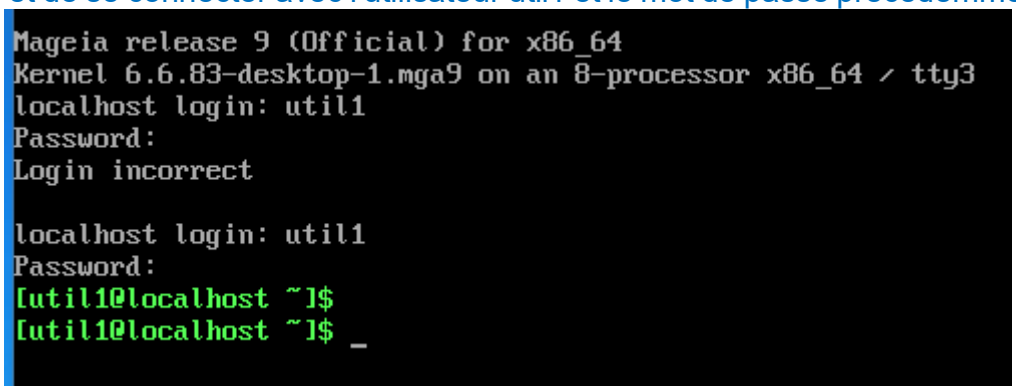


Si l'on observe le fichier /etc/groups, on peut observer que le groupe « gr1 » a été créer avec le GID « 1002 » :

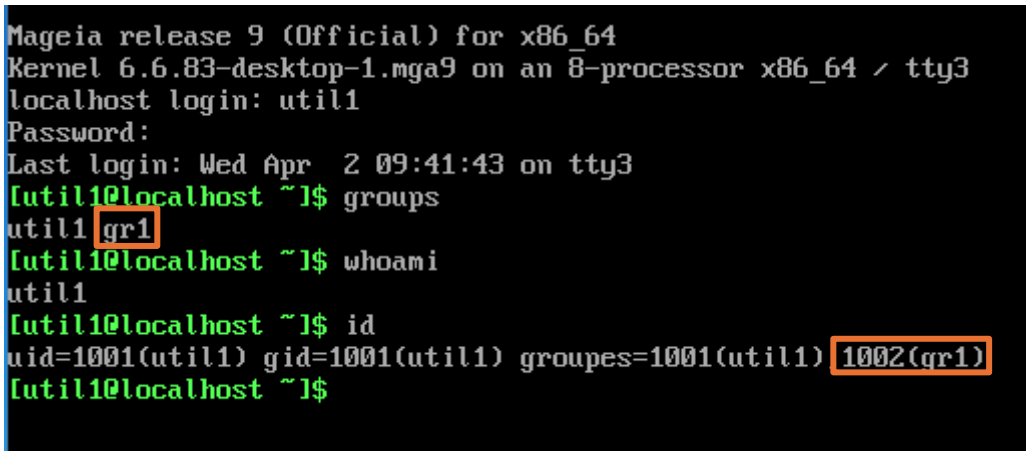


10. Lancez une nouvelle console et ouvrez une session avec le compte util1.

Pour se connecter au compte util1, il suffit de lancer une console grâce au raccourci « Ctrl Alt + F3 » et de se connecter avec l'utilisateur util1 et le mot de passe précédemment défini :



11. Dans la console util1, affichez les groupes de l'utilisateur avec la commande *groups*. Utilisez les commandes *whoami* et *id*. Fermez la connexion avec util1.

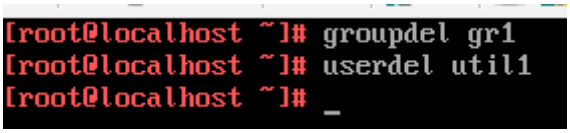


On observe que nous somme bien connecter en tant que « util1 » et que l'utilisateur est bien présent dans le groupe « gr1 ». Le groupe par défaut (avec le GID=1001) tant qu'à lui n'a pas bougé étant donné qu'il s'agît du groupe par défaut de l'utilisateur util1.

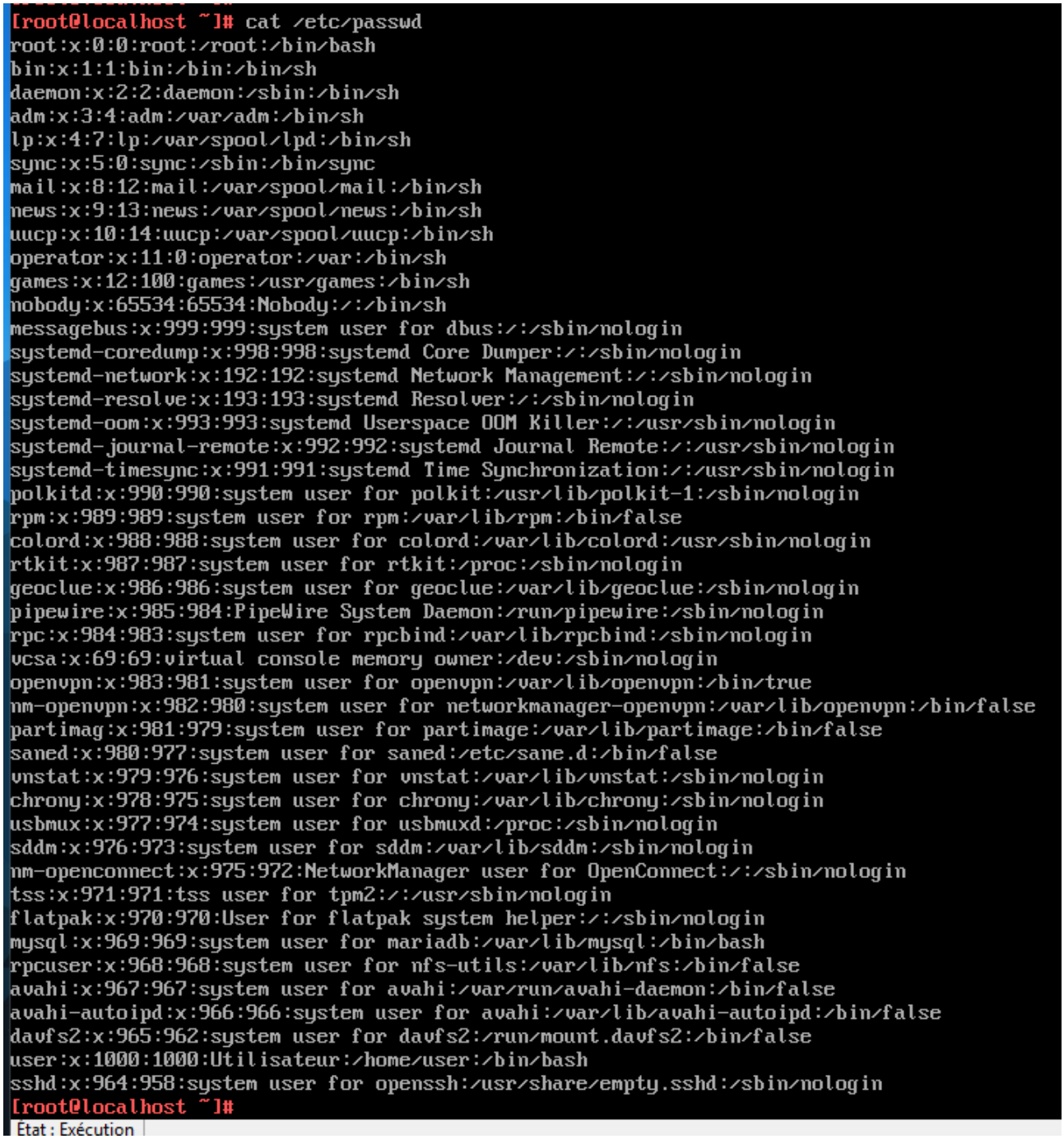
12. Dans la console administrateur, supprimez le groupe gr1 puis l'utilisateur util1. Vérifiez dans les fichiers /etc/passwd, /etc/shadow et /etc/group.

Suppression du groupe avec la commande : « groupdel gr1 »

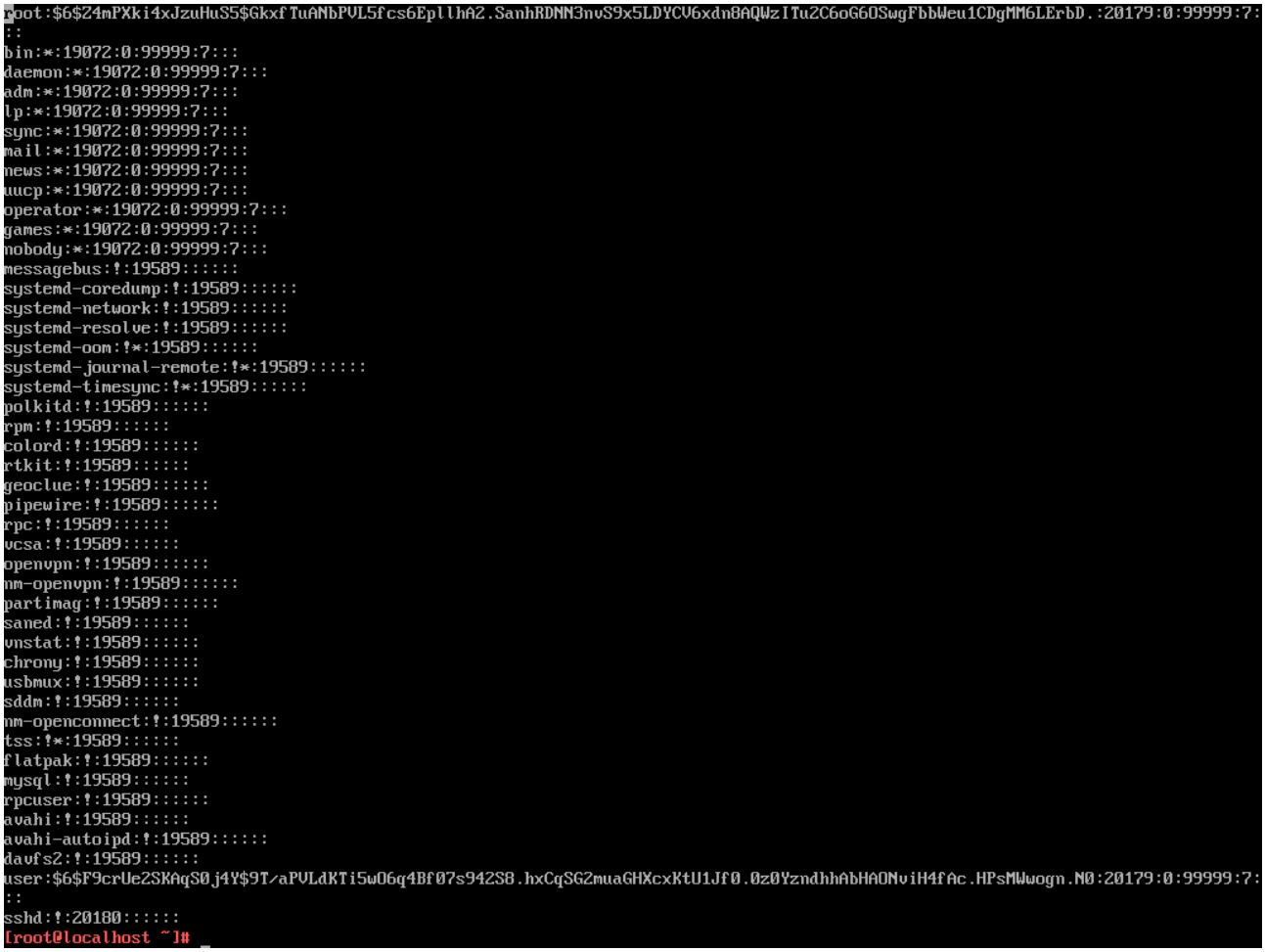
Suppression de l'utilisateur avec la commande : « userdel util1 »



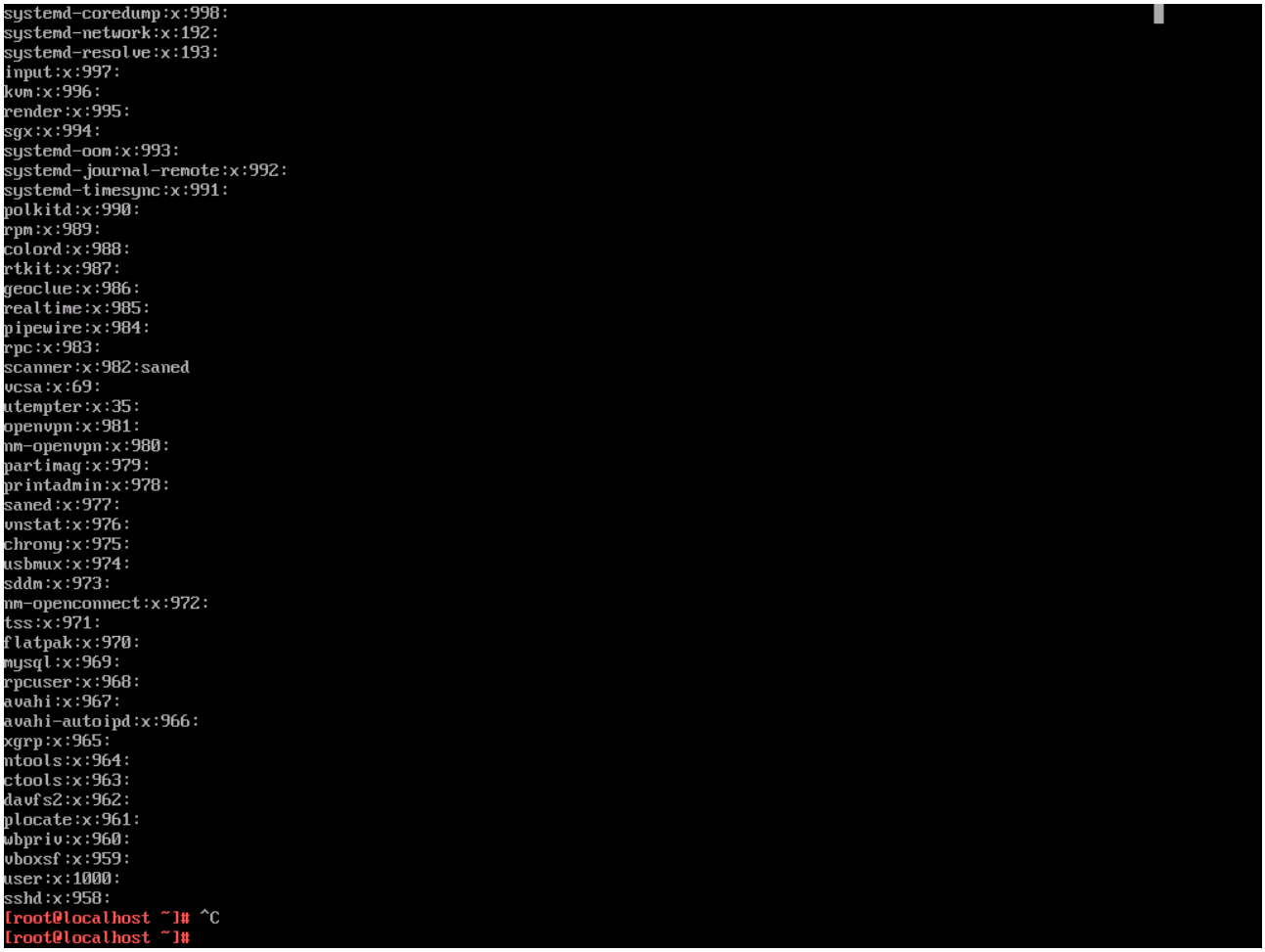
Vérification du dossier /etc/passwd (l'utilisateur util1 n'est plus présent et donc a bien été supprimé) :



Vérification du fichier /etc/shadow (même remarque que pour le fichier /etc/passwd, l'utilisateur util1 a totalement disparu ce qui signifie qu'il a bien été supprimé) :



Vérification du fichier /etc/group (on remarque que le groupe de l'utilisateur util1 nommé lui aussi util1 a bien été supprimé, mais aussi le groupe gr1 qui lui aussi a été supprimé) :



IV. Les droits d'accès

1. Dans votre répertoire personnel créez un dossier test et mettez dedans deux fichiers vides f1 et f2 avec la commande *touch*. En restant dans votre répertoire personnel, avec la commande *ll*, affichez toutes les informations des fichiers de test.

```
[user@localhost ~]$ mkdir test
[user@localhost ~]$ cd test
[user@localhost test]$ touch f1 f2
[user@localhost test]$ ll
total 0
-rw-r--r-- 1 user user 0 avril  2 10:27 f1
-rw-r--r-- 1 user user 0 avril  2 10:27 f2
[user@localhost test]$ cd ..
[user@localhost ~]$ ll
total 40
drwxrwxr-x 2 user user 4096 avril  2 01:04 Desktop/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Documents/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Images/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Modèles/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Musique/
drwxr-xr-x 2 user user 4096 avril  2 08:29 rep1/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Téléchargements/
drwxr-xr-x 2 user user 4096 avril  2 10:27 test/
drwx------ 2 user user 4096 mars  19  2022 tmp/
drwxr-xr-x 2 user user 4096 avril  2 01:01 Vidéos/
[user@localhost ~]$
```

- Comment sont distingués les fichiers et les dossiers au niveau du premier caractère affiché ?

Si l'on regarde bien le premier caractère dans les permissions, on peut remarquer que les dossiers ont un « d » devant les permissions tandis que les fichiers possèdent simplement un "–" :

```
[user@localhost test]$ ll
total 4
-rw-r--r-- 1 user user    0 avril  2 10:27 f1
-rw-r--r-- 1 user user    0 avril  2 10:27 f2
drwxr-xr-x 2 user user 4096 avril  2 10:33 test/
[user@localhost test]$
```

- Que signifie le droit x pour un répertoire ? Pour un fichier ?

Le droit « x » correspond au droit d'exécution. Pour un fichier, cela signifie qu'il peut être lancé comme un programme ou un script. Pour un répertoire, ce droit permet de traverser le répertoire, c'est-à-dire d'accéder à ses fichiers ou sous-répertoires, même si le contenu ne peut pas être listé sans le droit de lecture.

- Quels sont les droits par défaut pour le propriétaire, le groupe et les autres ?

Pour un fichier :

- **Propriétaire :** lecture et écriture (rw-)
- **Groupe :** lecture (r--)
- **Autres :** lecture (r--)
- Ce qui donne en décimal : **644**.

Pour un répertoire :

- **Propriétaire :** lecture, écriture et exécution (rwx)
- **Groupe :** lecture et exécution (r-x)
- **Autres :** lecture et exécution (r-x)
- Ce qui donne en notation décimale **755**.

2. Pour le fichier f1, mettre le groupe users (commande *chgrp*). Controlez.

```
[root@localhost test]# chgrp users f1
[root@localhost test]# ll
total 4
-rw-r--r-- 1 user users    0 avril  2 10:27 f1
-rw-r--r-- 1 user user     0 avril  2 10:27 f2
drwxr-xr-x 2 user user  4096 avril  2 10:33 test/
[root@localhost test]#
```

3. Modifiez les droits sur le fichier f1 pour autoriser tout pour le groupe users : commande *chmod g+rwx f1*.

```
[root@localhost test]# chmod g+rwx f1
[root@localhost test]# ll
total 4
-rw-rwxr-- 1 user users    0 avril  2 10:27 f1*
-rw-r--r-- 1 user user     0 avril  2 10:27 f2
drwxr-xr-x 2 user user  4096 avril  2 10:33 test/
[root@localhost test]#
```

5. Changez le propriétaire du fichier f2 (commande *chown*) pour mettre nobody.

```
[root@localhost test]# chown nobody f2
[root@localhost test]# ll
total 4
-rw-rwxr-- 1 user   users    0 avril  2 10:27 f1*
-rw-r--r-- 1 nobody user     0 avril  2 10:27 f2
drwxr-xr-x 2 user   user  4096 avril  2 10:33 test/
[root@localhost test]#
```

5. Changez les droits sur le fichier f2 en utilisant la notation octale : propriétaire en lecture et écriture, et groupe en lecture seulement, les autres à aucun accès. Vérifiez le résultat.

Commande utilisée : « chmod 640 f2 »

Explication :

Tableau des permissions sous Linux :

Permission	Read (r)	Write (x)	Exécutable
Valeur octale	4	2	1

1. Pour que le propriétaire puisse avoir l'accès en lecture et écriture, il suffit de prendre la valeur read (4) et de l'additionner avec la valeur octale de write (2), ce qui nous donne 6. Cette valeur doit être définie en 1er pour le propriétaire du fichier.

2. Pour que le groupe puisse y avoir accès en lecture seule, il faut que la deuxième valeur soit égale à la valeur de read, et donc 4.

3. Et enfin, pour que les autres n'aient aucun accès, il suffit de mettre la dernière valeur à 0.

Vérification :

```
[root@localhost test]# chmod 640 f2
[root@localhost test]# ll
total 4
-rw-rwxr-- 1 user   users    0 avril  2 10:27 f1*
-rw-r----- 1 nobody user     0 avril  2 10:27 f2
drwxr-xr-x 2 user   user  4096 avril  2 10:33 test/
[root@localhost test]#
```

On remarque bien qu'ici, les permissions sur notre fichier sont défini sur "**rw-r------**" ce qui signifie bien que le propriétaire a un accès en lecture et écriture (premier triplet), que le groupe a un accès en lecture seul (deuxième triplet), et que les autres n'ont aucun accès (troisième triplet).

6. Supprimez le répertoire test et tout ce qu'il contient.

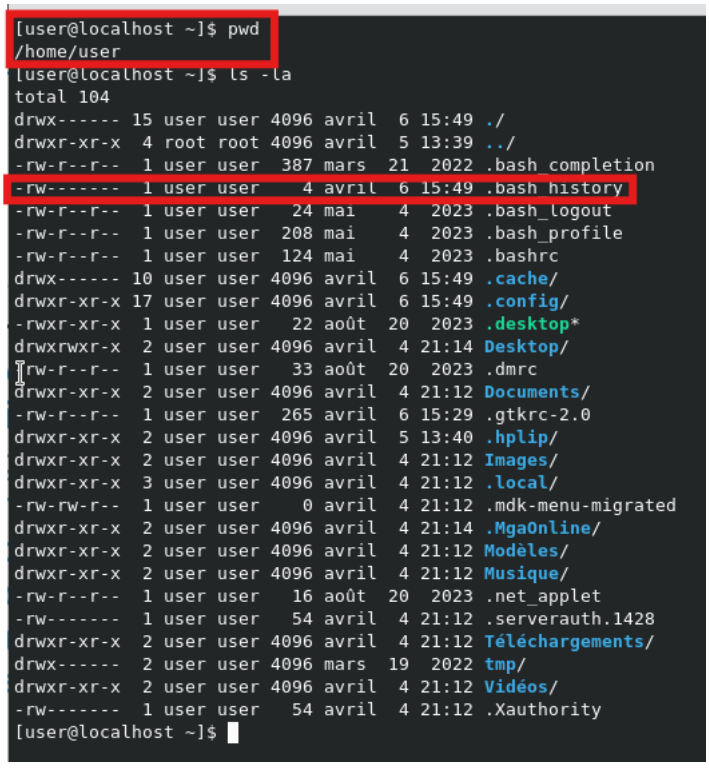
Pour supprimer le dossier test, il suffit de retourner au répertoire précèdent avec un « cd .. » et puis de faire un « rm -rf test » pour supprimer le dossier et tout ce qu'il contient :

V. Historique des commandes, complétion automatique et caractères génériques

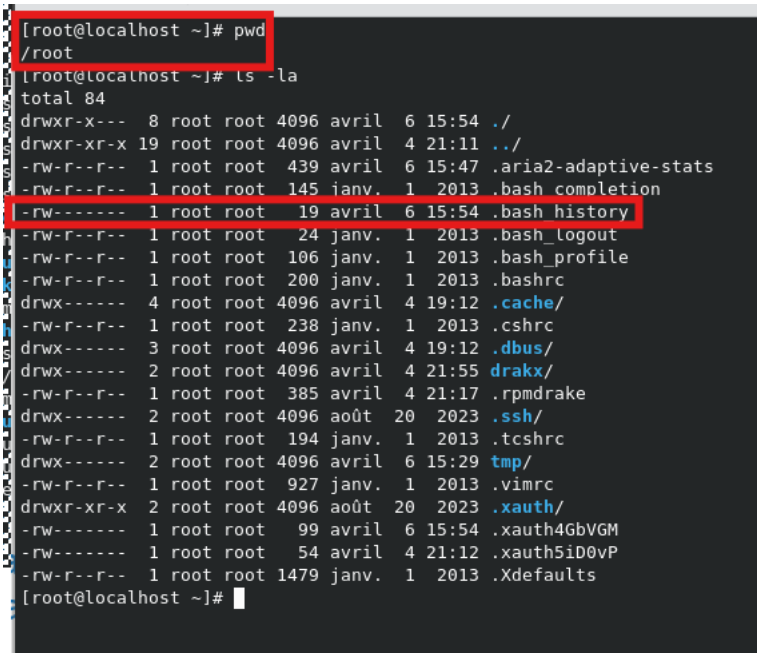
1. En appuyant sur les flèches du haut et du bas, on peut rappeler les commandes précédentes. Testez dans la console utilisateur et dans la console administrateur. Quelle est la différence et pourquoi ?

Vidéo montrant comment rappeler les commandes précédentes avec les flèches sur chaque utilisateur et montrant ou sont stockés ces informations (fichier .bash_history) : https://drive.proton.me/urls/7V94VM5JKR#jdlclcmUMFdC

En mode utilisateur, l'historique des commandes tapés dans bash est stockés dans le dossier suivant : /home/(nom d'utilisateur)/.bash_history qui est un fichier caché. Il est spécifique à l'utilisateur

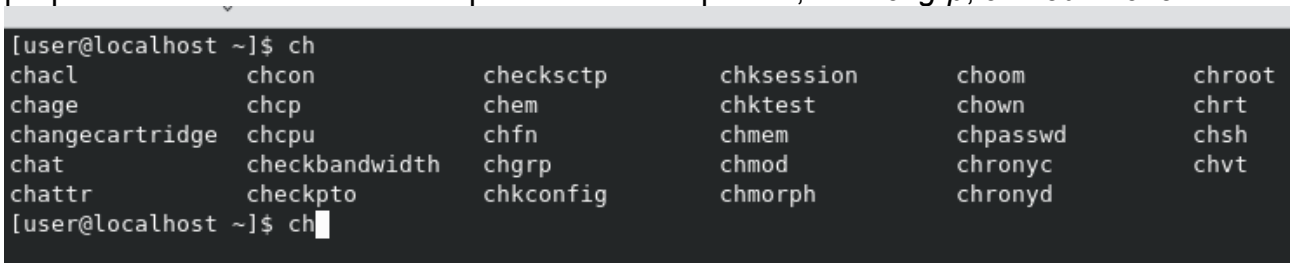


Tandis qu'en mode administrateur (root), l'historique des commandes bash sont stockés dans le dossier « /root/.bash_history »

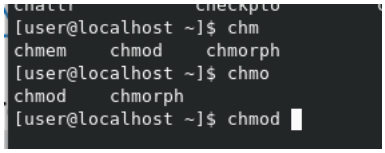


⚠ Il est important d'utiliser le paramètre -a pour afficher les fichiers masquer

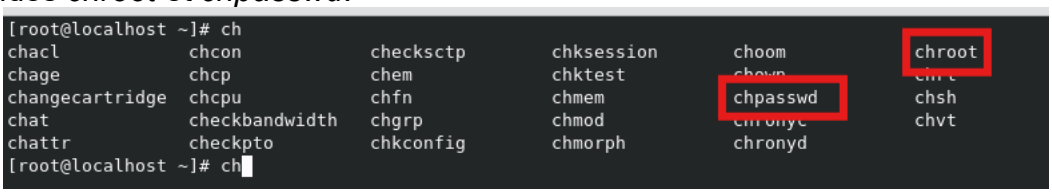
2. Dans la console utilisateur, tapez *ch* et appuyez sur la touche tabulation. Le système vous propose toutes les commandes qui commencent pas ch, dont *chgrp*, *chmod* et *chown*.



Rajoutez un m et appuyez sur la touche tabulation. Le système complète avec un o et vous donne les seules possibilités qui restent... Un outil très utile pour éviter les fautes de frappe et soutenir sa mémoire défaillante !



3. Dans la console administrateur, tapez *ch* et la touche tabulation. Vous voyez qu'il y a quelques commandes supplémentaires accessibles en super-utilisateur telles que les commandes *chroot* et *chpasswd*.



- **chroot** : Cette commande permet de changer le répertoire racine apparent pour un processus donné, créant ainsi un environnement isolé appelé "jail". C'est souvent utilisé pour des raisons de sécurité ou pour tester un système dans un espace restreint.
- **chpasswd** : Cette commande permet de changer les mots de passe des utilisateurs en batch à partir d'une entrée standard ou d'un fichier. Elle est utile pour automatiser la gestion des comptes sur un système Linux.

4. Autre exemple : on veut afficher le fichier des utilisateurs mais on ne se rappelle pas exactement le nom et l'emplacement. On tape *cat /e* et tabulation. Le système complète cat /etc/. Appuyez encore sur tabulation et il propose d'afficher tous les fichiers du répertoire. C'est trop long donc répondez non. Rajoutez un p et appuyez sur la touche tabulation, il vous propose la liste des fichiers du répertoire qui commencent par p. Vous rajoutez as, tabulation et entrée pour afficher ce que vous cherchiez. Facile !
Vidéo montrant les différentes actions effectuées pour afficher les différents éléments : https://drive.proton.me/urls/WMFMG89ZX0#nMJp9QlyTxLT

5. Dans votre répertoire personnel, créez les fichiers vides afile, bfile, cfile, file1, file2, file3, file1file, file2file, fileafile et filebfile.
Fichiers créer avec la commande « touch » :

```
[user@localhost ~]$ touch afile bfile cfile file1 file2 file3 file1file file2file fileafile filebfile
[user@localhost ~]$ ll
total 32
-rw-r--r-- 1 user user    0 avril  6 17:34 afile
-rw-r--r-- 1 user user    0 avril  6 17:34 bfile
-rw-r--r-- 1 user user    0 avril  6 17:34 cfile
drwxrwxr-x 2 user user 4096 avril  4 21:14 Desktop/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Documents/
-rw-r--r-- 1 user user    0 avril  6 17:34 file1
-rw-r--r-- 1 user user    0 avril  6 17:34 file1file
-rw-r--r-- 1 user user    0 avril  6 17:34 file2
-rw-r--r-- 1 user user    0 avril  6 17:34 file2file
-rw-r--r-- 1 user user    0 avril  6 17:34 file3
-rw-r--r-- 1 user user    0 avril  6 17:34 fileafile
-rw-r--r-- 1 user user    0 avril  6 17:34 filebfile
drwxr-xr-x 2 user user 4096 avril  4 21:12 Images/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Modèles/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Musique/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Téléchargements/
drwx------ 2 user user 4096 mars  19  2022 tmp/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Vidéos/
[user@localhost ~]$
```

6. Le point d'interrogation remplace n'importe quel caractère. L'astérisque remplace un groupe de caractères et les crochets permettent de spécifier une suite. Exemples : testez les commandes *ls file?*, *ls file** et *ls file[1-3]*.
La commande « ls file? » affiche tous les fichiers commençant par file et ayant un seul caractère inconnu à la fin :

```
[user@localhost ~]$ ls file?
file1  file2  file3
[user@localhost ~]$
```

La commande « ls file* » affiche tous les fichiers commençant par file (ayant un ou plusieurs caractères en plus) :

```
[user@localhost ~]$ ls file*
file1  file1file  file2  file2file  file3  fileafile  filebfile
[user@localhost ~]$
```

La commande « ls file[1-3] » affiche uniquement les fichiers commençant par file et dont les caractères finaux sont compris entre 1 et 3 :

```
[user@localhost ~]$ ls file[1-3]
file1  file2  file3
```

7. Trouvez la commande la plus compacte pour supprimer seulement les fichiers afile, bfile et cfile. Supprimez ensuite uniquement les fichiers file1file et file2file, puis tous ceux commençant par file.

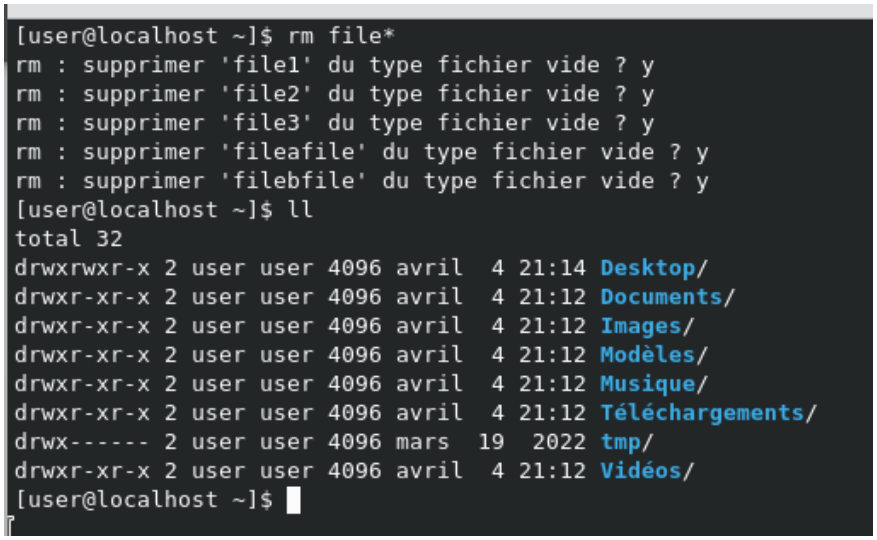
La commande la plus compacte pour supprimer seulement les fichiers afile, bfile et cfile serait la commande « rm ?file » qui supprimerait tous les fichiers commençant par un caractère inconnu et finissant par « file » :

```
[user@localhost ~]$ rm ?file
rm : supprimer 'afile' du type fichier vide ? y
rm : supprimer 'bfile' du type fichier vide ? y
rm : supprimer 'cfile' du type fichier vide ? y
[user@localhost ~]$ ll
total 32
drwxrwxr-x 2 user user 4096 avril  4 21:14 Desktop/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Documents/
-rw-r--r-- 1 user user    0 avril  6 17:34 file1
-rw-r--r-- 1 user user    0 avril  6 17:34 file1file
-rw-r--r-- 1 user user    0 avril  6 17:34 file2
-rw-r--r-- 1 user user    0 avril  6 17:34 file2file
-rw-r--r-- 1 user user    0 avril  6 17:34 file3
-rw-r--r-- 1 user user    0 avril  6 17:34 fileafile
-rw-r--r-- 1 user user    0 avril  6 17:34 filebfile
drwxr-xr-x 2 user user 4096 avril  4 21:12 Images/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Modèles/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Musique/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Téléchargements/
drwx------ 2 user user 4096 mars  19  2022 tmp/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Vidéos/
[user@localhost ~]$
```

De même pour file1file et file2file, mais cette fois ci en plaçant un « [1-2] au milieu (à la place du chiffre étant le seul caractère à être modifié, et en indiquant bien les caractères devant être supprimés pour ne pas que fileafile et filebfile sois supprimés) :

```
[user@localhost ~]$ rm file[1-2]file
rm : supprimer 'file1file' du type fichier vide ? y
rm : supprimer 'file2file' du type fichier vide ? y
[user@localhost ~]$ ll
total 32
drwxrwxr-x 2 user user 4096 avril  4 21:14 Desktop/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Documents/
-rw-r--r-- 1 user user    0 avril  6 17:34 file1
-rw-r--r-- 1 user user    0 avril  6 17:34 file2
-rw-r--r-- 1 user user    0 avril  6 17:34 file3
-rw-r--r-- 1 user user    0 avril  6 17:34 fileafile
-rw-r--r-- 1 user user    0 avril  6 17:34 filebfile
drwxr-xr-x 2 user user 4096 avril  4 21:12 Images/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Modèles/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Musique/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Téléchargements/
drwx------ 2 user user 4096 mars  19  2022 tmp/
drwxr-xr-x 2 user user 4096 avril  4 21:12 Vidéos/
[user@localhost ~]$
```

Puis enfin pour terminer, pour être sûr de supprimer tous les fichiers commençant par « file », on va utiliser la commande « rm file* » :



On remarque bien qu'à la fin, tous les fichiers sont bien supprimés du répertoire personnel.