

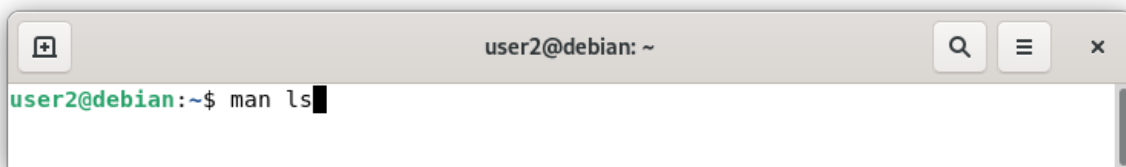
Shell

JOB 1

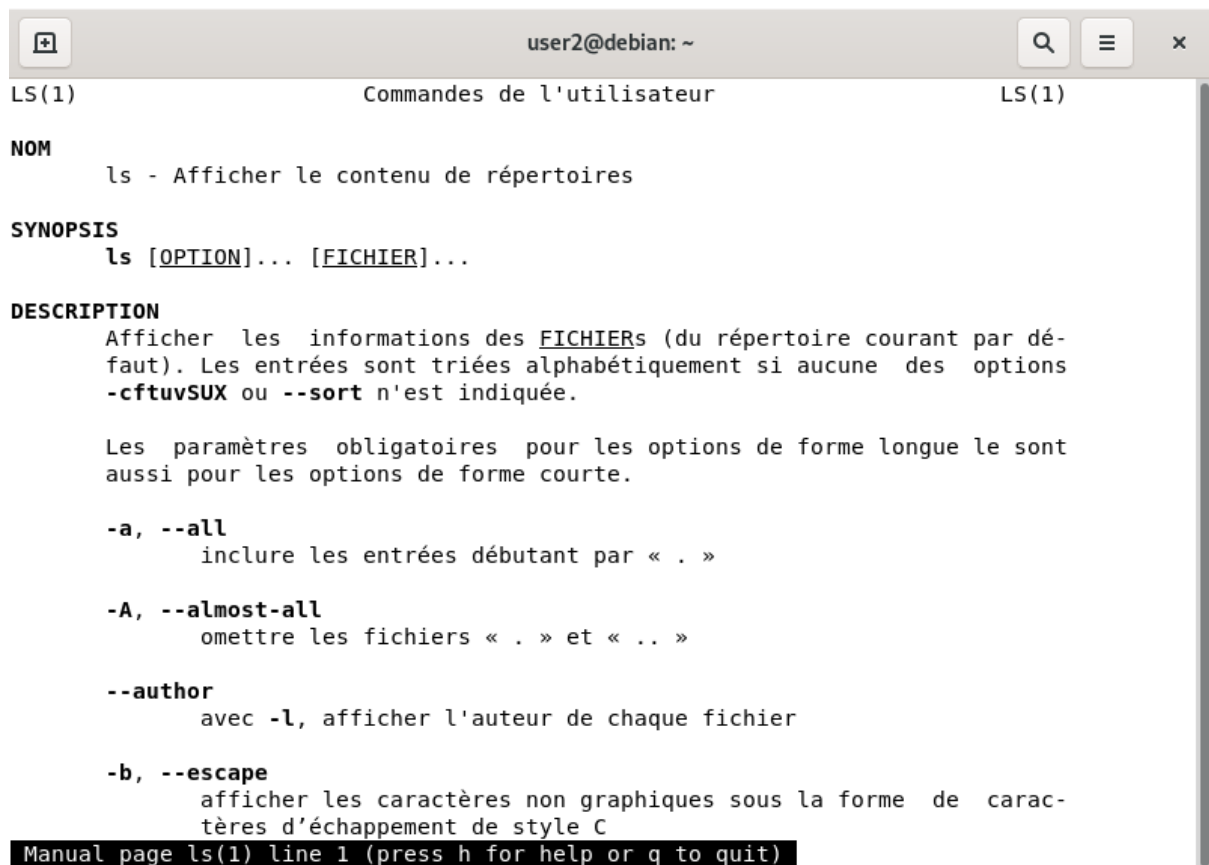
Afficher le manuel de la commande ls

man ls

man : La commande man est disponible sous Linux et permet de visionner le manuel d'une commande ou le manuel d'un fichier de configuration. Celle-ci est très utile pour obtenir des informations complètes.



```
user2@debian: ~  
user2@debian:~$ man ls
```



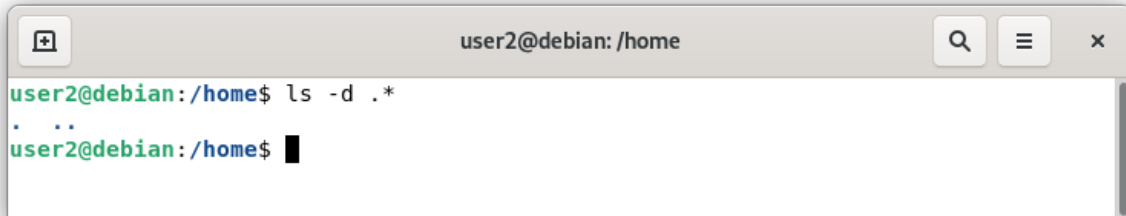
```
user2@debian: ~  
LS(1)                               Commandes de l'utilisateur                               LS(1)  
  
NOM  
    ls - Afficher le contenu de répertoires  
  
SYNOPSIS  
    ls [OPTION]... [FICHIER]...  
  
DESCRIPTION  
    Afficher les informations des FICHIERS (du répertoire courant par défaut). Les entrées sont triées alphabétiquement si aucune des options -cftuvSUX ou --sort n'est indiquée.  
  
    Les paramètres obligatoires pour les options de forme longue le sont aussi pour les options de forme courte.  
  
    -a, --all  
        inclure les entrées débutant par « . »  
  
    -A, --almost-all  
        omettre les fichiers « . » et « .. »  
  
    --author  
        avec -l, afficher l'auteur de chaque fichier  
  
    -b, --escape  
        afficher les caractères non graphiques sous la forme de caractères d'échappement de style C  
Manual page ls(1) line 1 (press h for help or q to quit)
```

Afficher les fichiers cachés du home de votre utilisateur

Se placer dans le dossier “home” puis écrire la commande `ls -d .*`

ls : Elle permet de lister le contenu du répertoire que vous souhaitez (le répertoire courant par défaut), y compris les fichiers et autres répertoires imbriqués.

-d .* : N'afficher que les fichiers et dossiers qui commencent par un point (cachés).

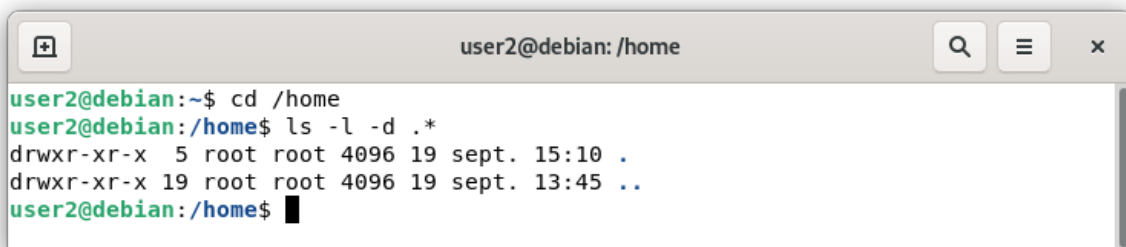


```
user2@debian: /home
user2@debian:/home$ ls -d .*
.  ..
user2@debian:/home$
```

Afficher les fichiers cachés plus les informations sur les droits sous forme de liste

ls -l -d .*

ls -l : Affiche des informations précises sur les fichiers dans un ou plusieurs répertoires.



```
user2@debian:~$ cd /home
user2@debian:/home$ ls -l -d .*
drwxr-xr-x  5 root root 4096 19 sept. 15:10 .
drwxr-xr-x 19 root root 4096 19 sept. 13:45 ..
user2@debian:/home$
```

Comment ajouter des options à une commande ?

Les paramètres qui commencent par « - » ou « -- » sont appelés options et contrôlent le comportement de la commande.

Quelles sont les deux syntaxes principales d'écriture des options pour une commande ?

commande -options paramètres

commande -options --options-longues paramètres

JOB 2

Lisez un fichier en utilisant une commande qui permet seulement de lire

Se placer dans le dossier qui contient le fichier puis écrire **cat <nom du fichier>**

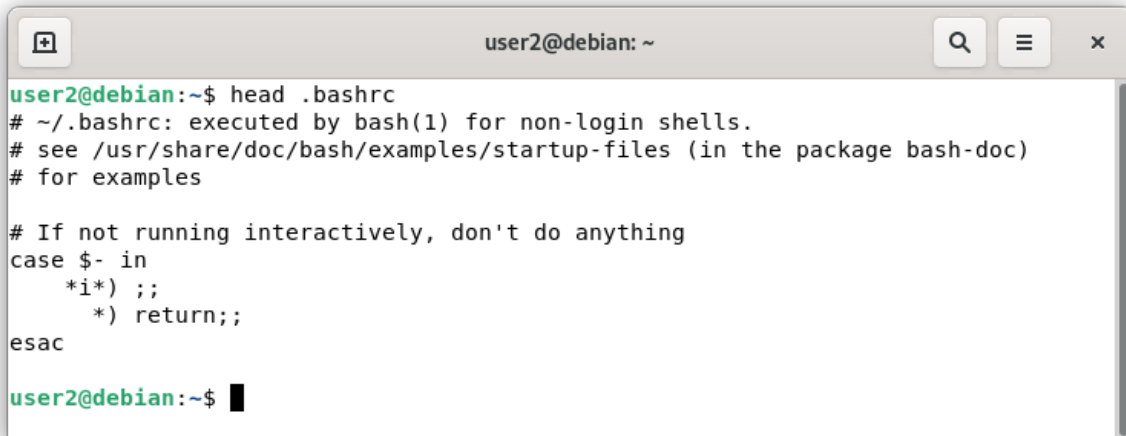
```
user2@debian: ~  
user2@debian:~$ cat .bashrc  
# ~/.bashrc: executed by bash(1) for non-login shells.  
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)  
# for examples  
  
# If not running interactively, don't do anything  
case $- in  
    *i*) ;;  
    *) return;;  
esac  
  
# don't put duplicate lines or lines starting with space in the history.  
# See bash(1) for more options  
HISTCONTROL=ignoreboth  
  
# append to the history file, don't overwrite it  
shopt -s histappend  
  
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)  
HISTSIZE=1000  
HISTFILESIZE=2000  
  
# check the window size after each command and, if necessary,  
# update the values of LINES and COLUMNS.  
shopt -s checkwinsize  
  
# If set, the pattern "*" used in a pathname expansion context will  
# match all files and zero or more directories and subdirectories.  
#shopt -s globstar
```

Pour les prochaines questions, il faut au préalable se placer dans le dossier mère de “.bashrc”.

- afficher les 10 premières lignes du fichier “.bashrc”

`head .bashrc`

Écrit comme `head <fichier>`, la commande head de Linux affiche les dix premières lignes du fichier spécifié.

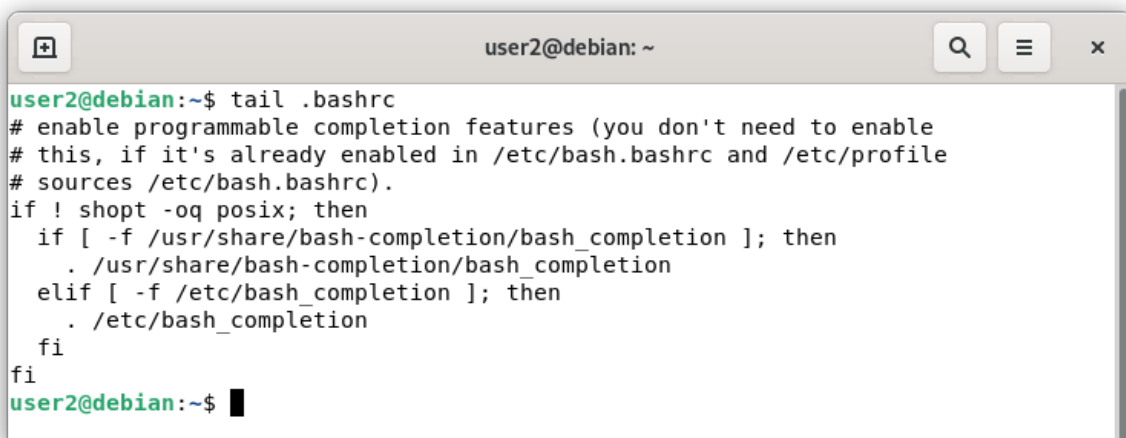


```
user2@debian: ~  
user2@debian:~$ head .bashrc  
# ~/.bashrc: executed by bash(1) for non-login shells.  
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)  
# for examples  
  
# If not running interactively, don't do anything  
case $- in  
  *i*) ;;  
  *) return;;  
esac  
  
user2@debian:~$
```

- afficher les 10 dernières lignes du fichier “.bashrc”

`tail .bashrc`

Écrit comme `tail <fichier>`, la commande tail de Linux affiche les dix dernières lignes du fichier spécifié.

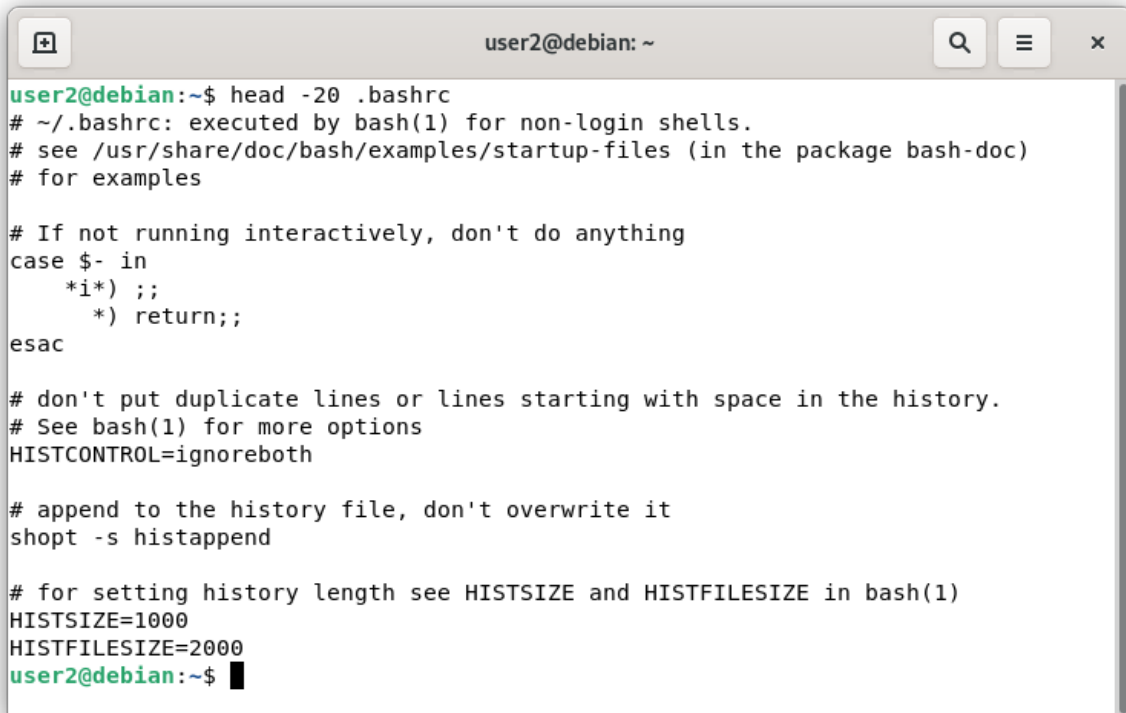


```
user2@debian: ~  
user2@debian:~$ tail .bashrc  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
  if [ -f /usr/share/bash-completion/bash_completion ]; then  
    . /usr/share/bash-completion/bash_completion  
  elif [ -f /etc/bash_completion ]; then  
    . /etc/bash_completion  
  fi  
fi  
user2@debian:~$
```

- afficher les 20 premières lignes du fichier “.bashrc”

`head -20 .bashrc`

On précise que l'on veut les 20 premières lignes avec `-20`. Si on voulait les 50 premières lignes, on changerait `-20` par `-50`

A terminal window titled 'user2@debian: ~' with search, menu, and close buttons. It displays the first 20 lines of the '.bashrc' file. The output shows various configuration comments and settings for the bash shell, including a case statement for non-interactive shells, history control settings, and history file size configurations.

```
user2@debian:~$ head -20 .bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

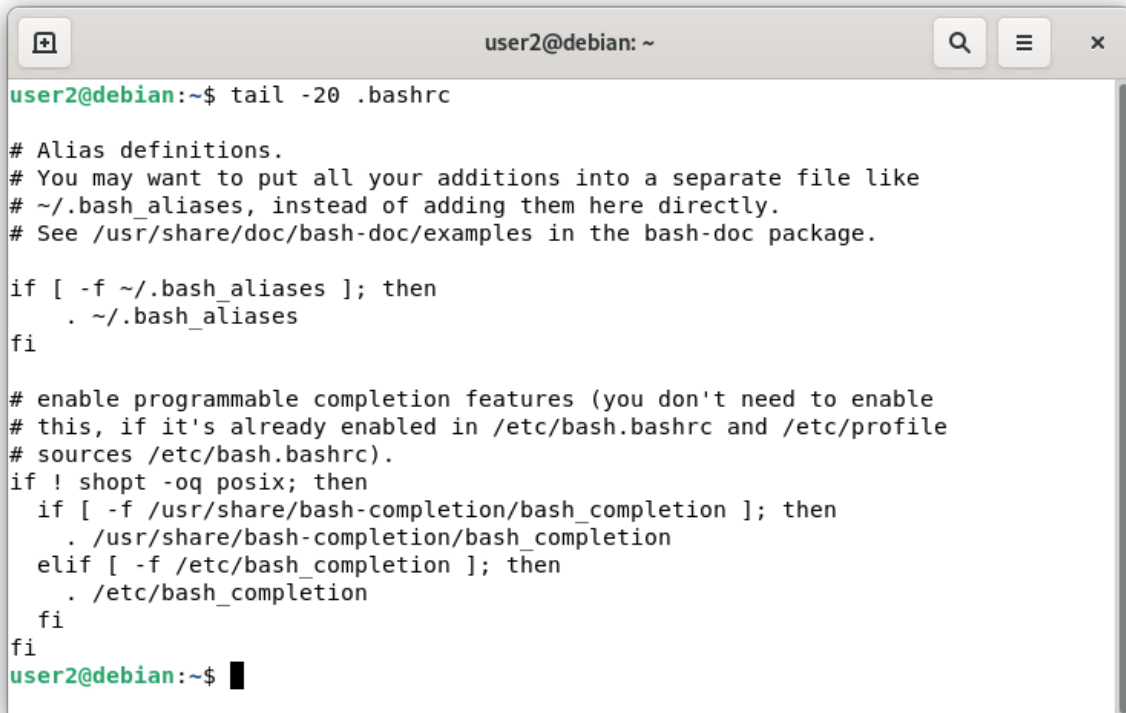
# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000
user2@debian:~$
```

- afficher les 20 dernières lignes du fichier “.bashrc”

tail -20 .bashrc

On précise que l'on veut les 20 dernières lignes avec **-20**. Si on voulait les 50 dernières lignes, on changerait **-20** par **-50**

A terminal window titled 'user2@debian: ~' with search, menu, and close buttons. The command 'tail -20 .bashrc' has been executed, displaying the last 20 lines of the .bashrc file. The output shows alias definitions, a section for sourcing ~/.bash_aliases, and a section for enabling bash completion features. The prompt 'user2@debian:~\$' is followed by a black cursor.

```
user2@debian:~$ tail -20 .bashrc

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi
user2@debian:~$ █
```

JOB 3

- Installer le paquet “cmatrix”

`sudo apt install cmatrix`

`sudo` : il faut être super-utilisateur pour pouvoir installer un paquet.

`apt install` : apt permet d'installer des logiciels sur les distributions à base de Debian depuis les dépôts.

A terminal window titled 'user2@debian: ~' with search, menu, and close buttons. The command 'sudo apt install cmatrix' has been executed. The output shows the password prompt, progress of package list reading, dependency tree construction, and state information reading. It confirms that cmatrix version 2.0-3 is already installed and up-to-date. The prompt 'user2@debian:~\$' is followed by a black cursor.

```
user2@debian:~$ sudo apt install cmatrix
[sudo] Mot de passe de user2 :
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
cmatrix est déjà la version la plus récente (2.0-3).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
user2@debian:~$ █
```

- lancer le paquet que vous venez d'installer

cmatrix

On a juste à écrire **cmatrix** pour la lancer (faire Ctrl+C pour en sortir de cmatrix)

```
user2@debian: ~  
user2@debian:~$ cmatrix
```



- Mettre à jour son gestionnaire de paquets et ses différents logiciels

sudo apt update && sudo apt upgrade

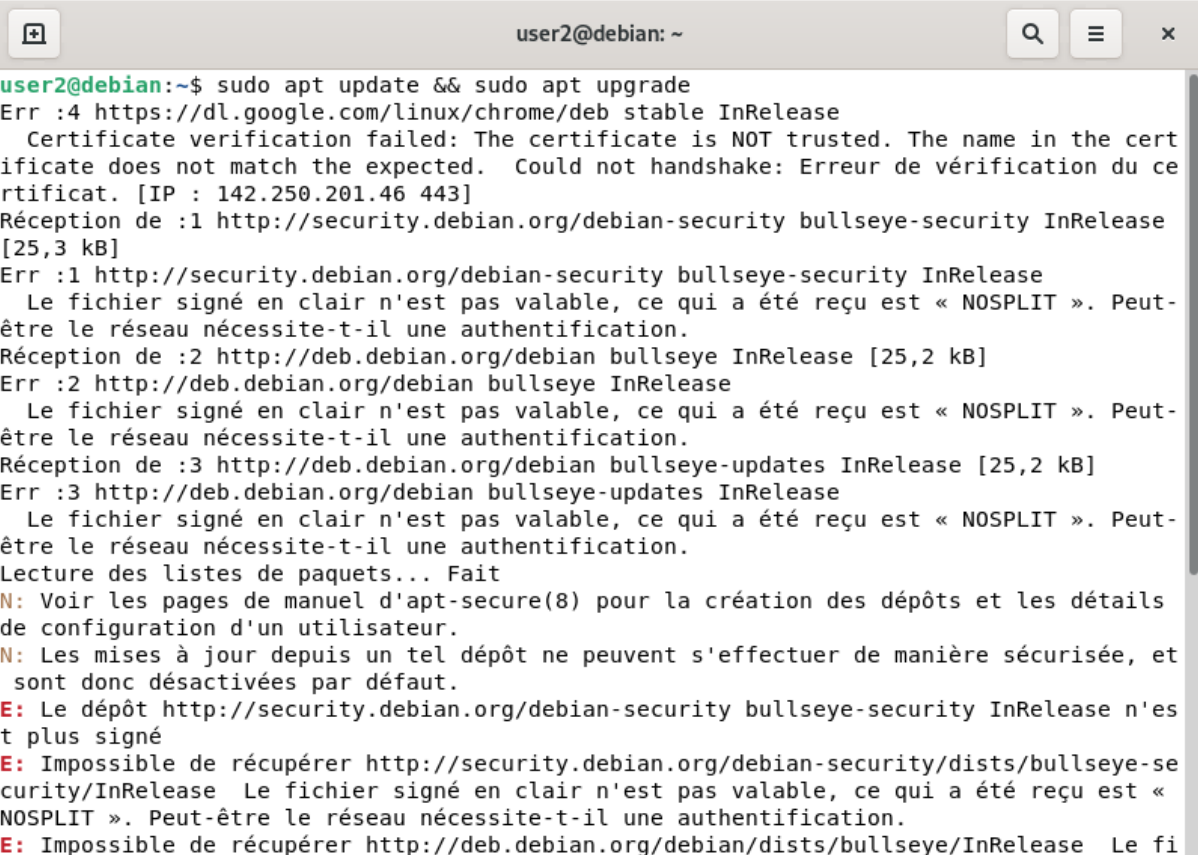
apt update : L'instruction update va rechercher les mises à jour disponibles pour votre système et vos programmes installés en se basant sur les sources définies dans /etc/apt/source.list. Un fichier d'index est créé pour lister les mises à jour disponibles.

apt upgrade : L'option upgrade installe les mises à jour identifiées avec apt update sans supprimer les paquets installés. S'il y a de nouvelles dépendances à installer, les paquets peuvent être installés ou non selon le type de commande utilisée apt, apt-get ou aptitude.

Il existe aussi 2 autres commandes pour mettre à jour son gestionnaire de paquets et ses différents logiciels: **apt dist-upgrade** et **apt full-upgrade** . Les options

dist-upgrade ou full-upgrade sont identiques, utiliser l'une ou l'autre revient donc au même. Ces deux options agissent plus « intelligemment » que la fonction upgrade. En plus de mettre à jour les paquets existants, elles vont également être en mesure de gérer les dépendances. Si de nouveaux paquets doivent être installés pour satisfaire des dépendances, ils le seront. Ceux qui ne sont plus utiles, sont supprimés et les paquets essentiels ou requis, sont installés. Les paquets les plus importants sont traités en priorité. Cette commande est moins conservatrice et peut causer des résultats inattendus en raison des possibilités de nouvelles installations ou de suppressions.

&& : Opérateur logique. La commande qui suit **&&** ne s'exécutera que si la première commande a été exécutée avec succès, c'est-à-dire que son état de sortie est zéro. Cet opérateur peut être utilisé pour vérifier si la première commande a été exécutée avec succès.



```
user2@debian: ~  
user2@debian:~$ sudo apt update && sudo apt upgrade  
Err :4 https://dl.google.com/linux/chrome/deb stable InRelease  
  Certificate verification failed: The certificate is NOT trusted. The name in the certificate does not match the expected. Could not handshake: Erreur de vérification du certificat. [IP : 142.250.201.46 443]  
Réception de :1 http://security.debian.org/debian-security bullseye-security InRelease [25,3 kB]  
Err :1 http://security.debian.org/debian-security bullseye-security InRelease  
  Le fichier signé en clair n'est pas valable, ce qui a été reçu est « NOSPLIT ». Peut-être le réseau nécessite-t-il une authentification.  
Réception de :2 http://deb.debian.org/debian bullseye InRelease [25,2 kB]  
Err :2 http://deb.debian.org/debian bullseye InRelease  
  Le fichier signé en clair n'est pas valable, ce qui a été reçu est « NOSPLIT ». Peut-être le réseau nécessite-t-il une authentification.  
Réception de :3 http://deb.debian.org/debian bullseye-updates InRelease [25,2 kB]  
Err :3 http://deb.debian.org/debian bullseye-updates InRelease  
  Le fichier signé en clair n'est pas valable, ce qui a été reçu est « NOSPLIT ». Peut-être le réseau nécessite-t-il une authentification.  
Lecture des listes de paquets... Fait  
N: Voir les pages de manuel d'apt-secure(8) pour la création des dépôts et les détails de configuration d'un utilisateur.  
N: Les mises à jour depuis un tel dépôt ne peuvent s'effectuer de manière sécurisée, et sont donc désactivées par défaut.  
E: Le dépôt http://security.debian.org/debian-security bullseye-security InRelease n'est plus signé  
E: Impossible de récupérer http://security.debian.org/debian-security/dists/bullseye-security/InRelease Le fichier signé en clair n'est pas valable, ce qui a été reçu est « NOSPLIT ». Peut-être le réseau nécessite-t-il une authentification.  
E: Impossible de récupérer http://deb.debian.org/debian/dists/bullseye/InRelease Le fi
```

- Télécharger les internets : Google

1. **sudo wget**

https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb

Wget est un programme en ligne de commande non interactif de téléchargement de fichiers depuis le Web. Il supporte les protocoles HTTP, HTTPS et FTP ainsi que le téléchargement au travers des proxies HTTP.


```
user2@debian: ~  
user2@debian:~$ sudo wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb  
--2022-09-21 12:48:07-- https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb  
Résolution de dl.google.com (dl.google.com)... 142.250.201.46, 2a00:1450:4006:80f:200e  
Connexion à dl.google.com (dl.google.com)|142.250.201.46|:443... connecté.  
requête HTTP transmise, en attente de la réponse... 200 OK  
Taille : 91239488 (87M) [application/x-debian-package]  
Sauvegarde en : « google-chrome-stable_current_amd64.deb »  
  
google-chrome-stable_ 100%[=====] 87,01M 21,2MB/s ds 4,1s  
  
2022-09-21 12:48:11 (21,2 MB/s) – « google-chrome-stable_current_amd64.deb » sauvegardé [91239488/91239488]  
user2@debian:~$
```

2. `sudo dpkg -i google-chrome-stable_current_amd64.deb`
`dpkg` (pour Debian package) est l'outil de bas niveau gérant les paquets des distributions basées sur Debian. Utilisable en ligne de commande, il est chargé de l'installation, la suppression et la gestion des paquets Debian (.deb), le type de paquets traités par Ubuntu. Il permet aussi la gestion de paquets Debian en provenance de sources extérieures aux dépôts APT.

```
user2@debian: ~  
user2@debian:~$ sudo dpkg -i google-chrome-stable_current_amd64.deb  
(Lecture de la base de données... 301044 fichiers et répertoires déjà installés.)  
Préparation du dépaquetage de google-chrome-stable_current_amd64.deb ...  
Dépaquetage de google-chrome-stable (105.0.5195.125-1) sur (105.0.5195.125-1) ..  
.  
Paramétrage de google-chrome-stable (105.0.5195.125-1) ...  
Traitement des actions différées (« triggers ») pour gnome-menus (3.36.0-1) ...  
Traitement des actions différées (« triggers ») pour desktop-file-utils (0.26-1) ...  
.  
Traitement des actions différées (« triggers ») pour man-db (2.9.4-2) ...  
user2@debian:~$  
user2@debian:~$
```

- Redémarrer votre machine

`sudo reboot` ou `sudo shutdown -r now`

`reboot` : Pour redémarrer la machine.

`shutdown` : La commande `shutdown` joue un rôle essentiel lorsque vous souhaitez arrêter ou redémarrer Linux via le terminal. Vous pouvez lui adjoindre une option

suivie par une heure et un message. La syntaxe de la commande shutdown de Linux ressemble à ce qui suit :

`shutdown [OPTION] [HEURE] [MESSAGE]`

si on veut redémarrer maintenant

`shutdown -r now`

si on veut redémarrer dans une minute

`shutdown -r 1`

A terminal window titled 'user2@debian: ~' showing the command 'sudo shutdown -r 1' being executed. The output is 'Reboot scheduled for Wed 2022-09-21 14:32:08 CEST, use 'shutdown -c' to cancel.' followed by two more prompts 'user2@debian:~\$'.

si on veut redémarrer à une heure précise, exemple 23h

`shutdown -r 23:00`

A terminal window titled 'user2@debian: ~' showing the command 'sudo shutdown -r 23:00' being executed. The output is 'Reboot scheduled for Wed 2022-09-21 23:00:00 CEST, use 'shutdown -c' to cancel.' followed by two prompts 'user2@debian:~\$'. The second prompt shows the command 'sudo shutdown -c' being entered.

et si on veut annuler un redémarrage

`shutdown -c`

Le système Linux peut également être redémarré à l'aide de la combinaison de touches Ctrl + Alt + Suppr.

- éteindre votre machine

`sudo poweroff` ou `sudo shutdown -h now`

poweroff : ferme les systèmes de fichier, éteint le système et met hors tension. Ne doit être utilisé qu'en mode mono utilisateur. Si le système n'est pas en niveau d'exécution 0 ou 6 appelle shutdown -h puis fait la mise hors tension.

shutdown : termine tout le traitement éventuellement à un horaire spécifié et diffuse un message à tous les utilisateurs leur demandant de se déconnecter. Exécute l'extinction. L'option -h arrête le système quand l'extinction est achevée.

si on veut éteindre maintenant

`shutdown -h now`

si on veut éteindre dans une minute

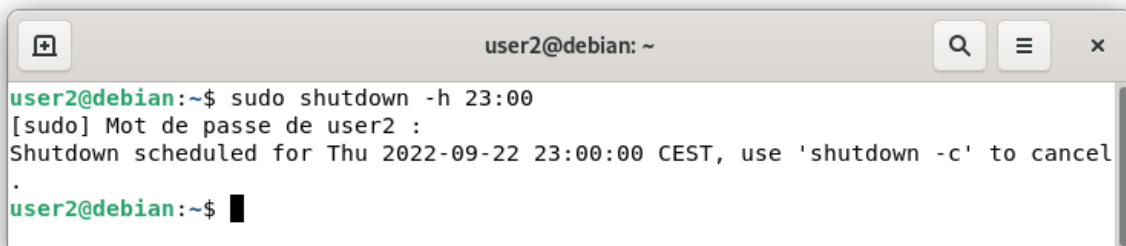
`shutdown -h 1`



```
user2@debian: ~  
user2@debian:~$ sudo shutdown -h 1  
Shutdown scheduled for Wed 2022-09-21 14:33:11 CEST, use 'shutdown -c' to cancel  
.  
user2@debian:~$ █
```

si on veut éteindre à une heure précise, exemple 23h

`shutdown -h 23:00`



```
user2@debian: ~  
user2@debian:~$ sudo shutdown -h 23:00  
[sudo] Mot de passe de user2 :  
Shutdown scheduled for Thu 2022-09-22 23:00:00 CEST, use 'shutdown -c' to cancel  
.  
user2@debian:~$ █
```

et si on veut annuler une extinction

`shutdown -c`

JOB 4

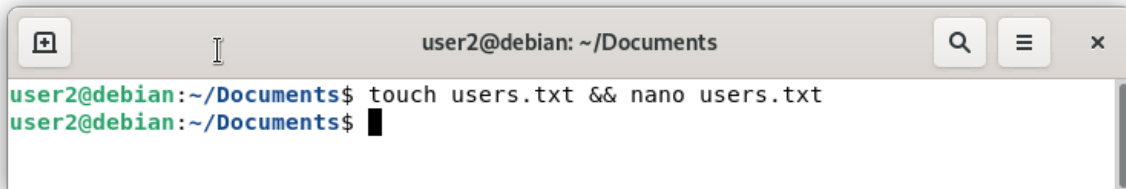
- Créer un fichier `users.txt` qui contiendra `User1` et `User2` séparé par un retour à la ligne

`touch users.txt && nano users.txt`

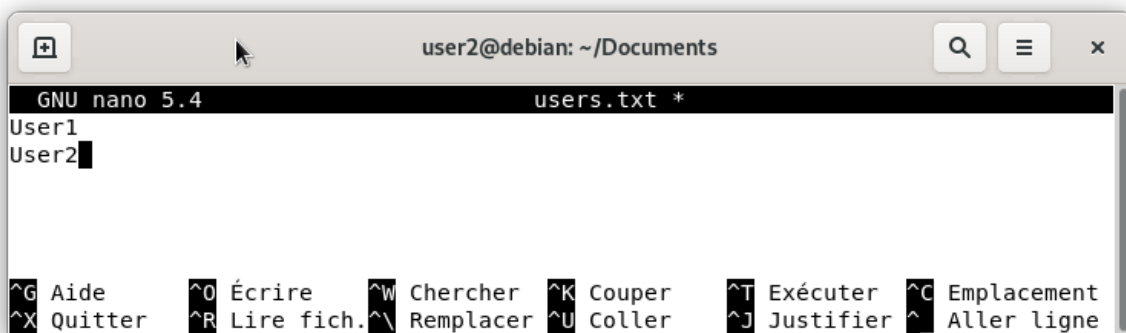
La commande `touch` de Linux est principalement utilisée pour créer des fichiers vides, et modifier le timestamp (horodatage) de fichiers ou de dossiers.

nano est un éditeur de texte pour les systèmes Unix et dérivés, basé sur la bibliothèque ncurses et publié sous licence GNU GPL.

Ecrire "User1" et "User2" puis faire ctrl+O pour enregistrer puis Entrée puis ctrl+X puis sortir de nano



```
user2@debian: ~/Documents
user2@debian:~/Documents$ touch users.txt && nano users.txt
user2@debian:~/Documents$
```

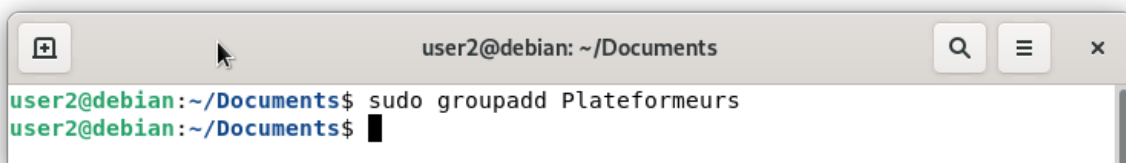


```
GNU nano 5.4 users.txt *
User1
User2
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

- Créer un groupe appelé "Plateformeurs"

sudo groupadd Plateformeurs

La commande **groupadd** crée un nouveau compte de groupe en utilisant les valeurs spécifiées sur la ligne de commande et les valeurs par défaut du système. Le nouveau groupe sera inséré dans les fichiers du système selon les besoins.



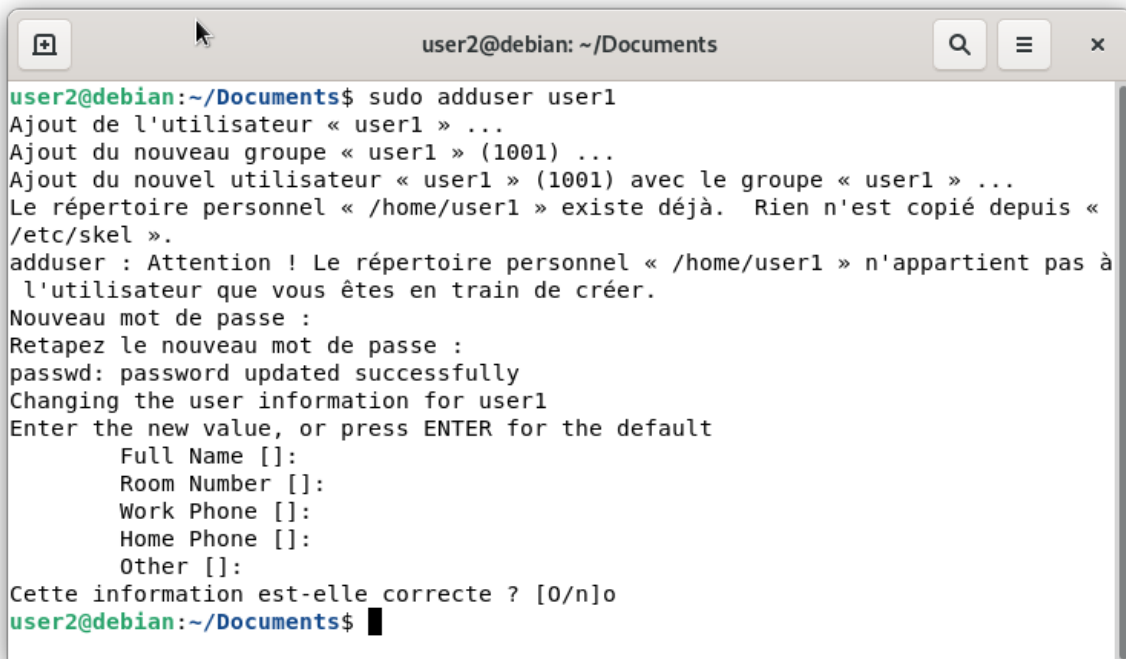
```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo groupadd Plateformeurs
user2@debian:~/Documents$
```

- Créer un utilisateur appelé "User1"

sudo adduser user1

On utilise la commande **/usr/sbin/useradd** pour créer les nouveaux comptes utilisateurs. La commande **adduser** est interactive et demande successivement : le mot de passe du nouvelle utilisateur, une confirmation de ce mot de passe, le nom complet de l'utilisateur, son numéro de chambre/bureau, son numéro de téléphone professionnel, son numéro de téléphone personnel et propose un champ d'entre d'autres informations. Hormis le mot de passe et la confirmation du mot de passe,

tous les autres champs sont optionnels. De plus ces informations sont modifiables via la commande `chfn`



```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo adduser user1
Ajout de l'utilisateur « user1 » ...
Ajout du nouveau groupe « user1 » (1001) ...
Ajout du nouvel utilisateur « user1 » (1001) avec le groupe « user1 » ...
Le répertoire personnel « /home/user1 » existe déjà. Rien n'est copié depuis « /etc/skel ».
adduser : Attention ! Le répertoire personnel « /home/user1 » n'appartient pas à l'utilisateur que vous êtes en train de créer.
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for user1
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Cette information est-elle correcte ? [0/n]o
user2@debian:~/Documents$
```

- Créer un utilisateur appelé “User2”

`sudo adduser user2`

Pareil que pour la question précédente.

- Ajouter “User2” au groupe Plateformeurs

`sudo adduser user2 Plateformeurs`

La commande `adduser` peut s'utiliser comme `adduser nom_utilisateur nom_groupe`. Utilisée de cette façon, elle permet d'ajouter un utilisateur à un groupe.



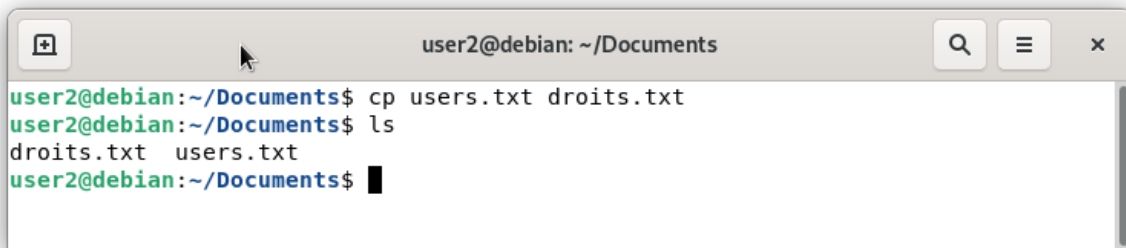
```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo adduser user2 Plateformeurs
Ajout de l'utilisateur « user2 » au groupe « Plateformeurs »...
Adding user user2 to group Plateformeurs
Fait.
user2@debian:~/Documents$
```

- Copier votre “users.txt” dans un fichier “droits.txt”

`cp users.txt droits.txt`

`cp` signifie copie. Cette commande est utilisée pour copier des fichiers ou un groupe de fichiers ou de répertoires.

Il crée une image exacte d'un fichier sur un disque avec un nom de fichier différent. La commande `cp` nécessite au moins deux noms de fichiers dans ses arguments.



```
user2@debian: ~/Documents
user2@debian:~/Documents$ cp users.txt droits.txt
user2@debian:~/Documents$ ls
droits.txt  users.txt
user2@debian:~/Documents$
```

- Copier votre “users.txt” dans un fichier “groupes.txt”

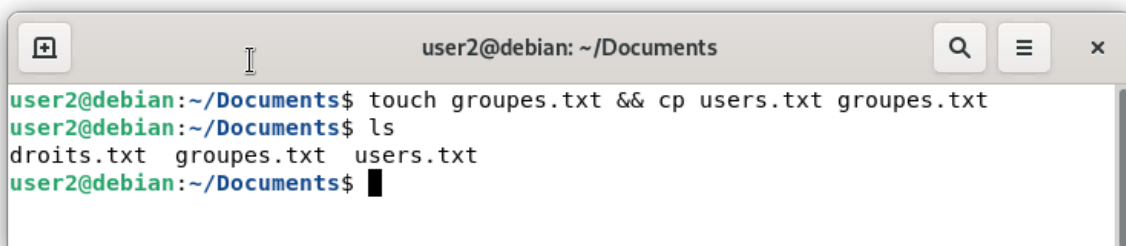
`touch groupes.txt && cp users.txt groupes.txt`

`touch` est une commande Unix standard permettant de créer un fichier vide. Cette commande permet également de modifier le timestamp de dernier accès et de dernière modification d'un fichier.

La commande `&&` ne s'exécutera que si la première commande a été exécutée avec succès, c'est-à-dire que son état de sortie est zéro. Cet opérateur peut être utilisé pour vérifier si la première commande a été exécutée avec succès.

La commande `cp` est utilisée pour copier des fichiers ou un groupe de fichiers ou de répertoires.

Il crée une image exacte d'un fichier sur un disque avec un nom de fichier différent. La commande `cp` nécessite au moins deux noms de fichiers dans ses arguments.



```
user2@debian: ~/Documents
user2@debian:~/Documents$ touch groupes.txt && cp users.txt groupes.txt
user2@debian:~/Documents$ ls
droits.txt  groupes.txt  users.txt
user2@debian:~/Documents$
```

- Changer le propriétaire du fichier “droits.txt” pour mettre “User1”

`sudo chown user1 droits.txt`

La commande `chown` permet de changer le propriétaire et/ou le groupe. La commande `chgrp` permet de changer le groupe.

Syntaxe de la commande `chown` : `chown [-option] [utilisateur][:groupe] cible1 [cible2 ..]` Par défaut lorsqu'un utilisateur crée un fichier ou un dossier, il en devient propriétaire et fait partie du groupe portant le même nom.

```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo chown user1 droits.txt
user2@debian:~/Documents$ la
bash: la : commande introuvable
user2@debian:~/Documents$ ls -la
total 20
drwxr-xr-x  2 user2 user2 4096 21 sept. 14:41 .
drwxr-xr-x 16 user2 user2 4096 21 sept. 12:50 ..
-rw-r--r--  1 user1 user2  12 21 sept. 14:40 droits.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:41 groupes.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:34 users.txt
user2@debian:~/Documents$
```

- Changer les droits du fichier “droits.txt” pour que “User2” ai accès seulement en lecture

`sudo chmod 740 droits.txt`

La commande `chmod` permet de modifier les droits et permissions. On utilise cette dernière en indiquant l'utilisateur (u, g, o ou a), les actions "+, -, =" et le droit concerné "r, w, x" ou la notation décimale.

740 : Ces chiffres définissent les droits donnés sur le fichier. Le premier chiffre désigne les droits du propriétaire du fichier, le deuxième ceux du groupe du propriétaire et le troisième ceux des autres utilisateurs. 0 : aucun droit. 1 : droit d'exécution. 2 : droit d'écriture. 3 : droit d'écriture et d'exécution. 4 : droit de lecture. 5 : droit de lecture et d'exécution. 6 : droit de lecture et d'écriture. 7 : droit de lecture, d'écriture et d'exécution.

```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo chmod 740 droits.txt
user2@debian:~/Documents$ ls -la
total 20
drwxr-xr-x  2 user2 user2 4096 21 sept. 14:41 .
drwxr-xr-x 16 user2 user2 4096 21 sept. 12:50 ..
-rwxr-----  1 user1 user2  12 21 sept. 14:40 droits.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:41 groupes.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:34 users.txt
user2@debian:~/Documents$
```

- Changer les droits du fichier “groupes.txt” pour que les utilisateurs puissent accéder au fichier en lecture uniquement

`sudo chmod 744 groupes.txt`

Même explication que la question précédente.

```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo chmod 744 groupes.txt
[sudo] Mot de passe de user2 :
user2@debian:~/Documents$ la
total 24
drwxr-xr-x  2 user2 user2 4096 21 sept. 15:25 .
drwxr-xr-x 17 user2 user2 4096 21 sept. 15:57 ..
-rwxr----- 1 user1 user2  12 21 sept. 14:40 droits.txt
-rwxr--r--  1 user2 user2  12 21 sept. 14:41 groupes.txt
-rw-r--r--  1 user2 user2  16 21 sept. 15:26 nb_lignes.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:34 users.txt
user2@debian:~/Documents$
```

- Changer les droits du fichier pour que le groupe "Plateformeurs" puisse y accéder en lecture/écriture.

`sudo chmod 760 groupes.txt`

Même explication que la question précédente.

```
user2@debian: ~/Documents
user2@debian:~/Documents$ sudo chmod 760 groupes.txt
user2@debian:~/Documents$ ls -la
total 20
drwxr-xr-x  2 user2 user2 4096 21 sept. 14:41 .
drwxr-xr-x 16 user2 user2 4096 21 sept. 12:50 ..
-rwxr----- 1 user1 user2  12 21 sept. 14:40 droits.txt
-rwxrw----  1 user2 user2  12 21 sept. 14:41 groupes.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:34 users.txt
user2@debian:~/Documents$
```

JOB 5

- Ajouter un alias qui permettra de lancer la commande "ls -la" en tapant "la"

`alias la='ls -la'`

`alias` : s'emploie principalement pour abréger une commande avec ses options ou rajouter par défaut des options à une commande régulièrement utilisée.

Syntaxe de la commande `alias` : `alias [NomAlias]='[commande avec ses options ou arguments]'`


```
user2@debian: ~/Documents
user2@debian:~/Documents$ alias la='ls -la'
user2@debian:~/Documents$ la
total 20
drwxr-xr-x  2 user2 user2 4096 21 sept. 14:41 .
drwxr-xr-x 16 user2 user2 4096 21 sept. 12:50 ..
-rwxr----- 1 user1 user2  12 21 sept. 14:40 droits.txt
-rwxrw----  1 user2 user2  12 21 sept. 14:41 groupes.txt
-rw-r--r--  1 user2 user2  12 21 sept. 14:34 users.txt
user2@debian:~/Documents$
```

Attention : les alias créés par cette commande disparaîtront quand vous fermerez le terminal.

- Ajouter un alias qui permettra de lancer la commande “apt-get update” en tapant “update”

`alias update='sudo apt-get update'`

Même explication que celle de la question précédente.

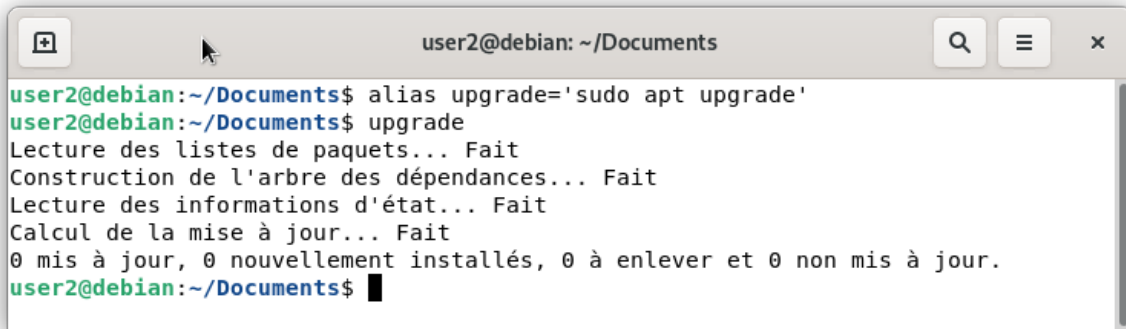
```
user2@debian: ~/Documents
user2@debian:~/Documents$ alias update='sudo apt update'
user2@debian:~/Documents$ update
Réception de :1 http://security.debian.org/debian-security bullseye-security InRelease [48,4 kB]
Atteint :2 http://deb.debian.org/debian bullseye InRelease
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease
48,4 ko réceptionnés en 1s (81,9 ko/s)
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Tous les paquets sont à jour.
user2@debian:~/Documents$
```

Attention : les alias créés par cette commande disparaîtront quand vous fermerez le terminal.

- Ajouter un alias qui permettra de lancer la commande “apt-get upgrade” en tapant “upgrade”

`alias upgrade='sudo apt-get upgrade'`

Même explication que celle de la question précédente.



```
user2@debian: ~/Documents
user2@debian:~/Documents$ alias upgrade='sudo apt upgrade'
user2@debian:~/Documents$ upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.
user2@debian:~/Documents$
```

Attention : les alias créés par cette commande disparaîtront quand vous fermerez le terminal.

- Ajouter une variable d'environnement qui se nommera "USER" et qui sera égale à votre nom d'utilisateur

Une variable est l'assignation d'une étiquette à un contenu ; ce contenu, comme l'indique le mot « variable », peut changer autant que l'on veut ; l'assignation de l'étiquette à ce contenu, elle, est fixe, aussi longtemps que l'on ne dissout pas la variable.

Il existe deux types de variables: les variables d'environnement et les variables de shell.

Une variable d'environnement sont des variables dynamiques utilisées par les différents processus d'un système d'exploitation. Elles servent à communiquer des informations entre les programmes qui ne se trouvent pas sur la même ligne hiérarchique, et qui ont donc besoin d'une convention pour se communiquer mutuellement leurs choix. Elles constituent un moyen d'influencer le comportement des logiciels sur votre système. Par exemple, la variable d'environnement « LANG » détermine la langue que les logiciels utilisent pour communiquer avec l'utilisateur.

Les variables de shell sont des variables qui sont exclusivement contenues dans le shell dans lequel elles ont été configurées ou définies. Elles sont couramment utilisées pour garder un suivi des données éphémères, comme le répertoire actuellement utilisé.

Par convention, ces types de variables sont généralement définis par des majuscules. Cela aide les utilisateurs à distinguer les variables d'environnement dans d'autres contextes.

`export USER=lea`

Une variable de shell peut être exportée pour devenir une variable d'environnement grâce à la commande export.

Syntaxe : `export [nom de la variable]=[valeur]`

```
user2@debian: ~/Documents
TERM=xterm-256color
USER=user2
GNOME_TERMINAL_SERVICE=:1.81
DISPLAY=:0
SHLV=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1002
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
GDMSESSION=gnome
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1002/bus
OLDPWD=/home/user2
_=/usr/bin/env
user2@debian:~/Documents$
```

```
user2@debian: ~/Documents
user2@debian:~/Documents$ export USER=lea
user2@debian:~/Documents$ env
SHELL=/bin/bash
SESSION_MANAGER=local/debian:@/tmp/.ICE-unix/1380,unix/debian:/tmp/.ICE-unix/1380
QT_ACCESSIBILITY=1
COLORTERM=truecolor
SSH_AGENT_LAUNCHER=openssh
XDG_MENU_PREFIX=gnome-
GNOME_DESKTOP_SESSION_ID=this-is-deprecated
SSH_AUTH_SOCK=/run/user/1002/keyring/ssh
XMODIFIERS=@im=ibus
DESKTOP_SESSION=gnome
```

```
user2@debian: ~/Documents
TERM=xterm-256color
USER=lea
GNOME_TERMINAL_SERVICE=:1.81
DISPLAY=:0
SHLV=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1002
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
GDMSESSION=gnome
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1002/bus
OLDPWD=/home/user2
_=/usr/bin/env
user2@debian:~/Documents$
```

Attention : les variables créées par cette commande disparaîtront quand vous fermerez le terminal.

- Mettre à jour les modifications de votre bashrc dans votre shell actuel
Se placer dans le dossier parent de .bashrc.

`nano ~/.bashrc` (`nano` est un éditeur de texte pour les systèmes Unix et dérivés, basé sur la bibliothèque ncurses et publié sous licence GNU GPL.)

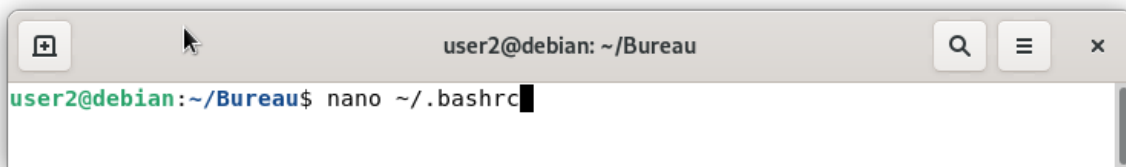
Écrire les commandes voulues dans le fichier (par exemple les alias précédents).

Taper Ctrl+O pour enregistrer les modifications puis Ctrl+X pour sortir de nano.

`source ~/.bashrc`

La commande de la source peut être utilisée pour recharger/rafraîchir un fichier de fonctions dans le script shell courant ou d'une invite de commande. Une fois le fichier `/home/user/.bashrc` modifié, pour le faire prendre en compte par la session courante, il faut lancer la commande source avec en paramètre le fichier.

Syntaxe : `source FILENAME <arguments>`



```
user2@debian: ~/Bureau
user2@debian:~/Bureau$ nano ~/.bashrc
```



```
GNU nano 5.4 /home/user2/.bashrc
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors ->
alias ls='ls --color=auto'
#alias dir='dir --color=auto'
#alias vdir='vdir --color=auto'

#alias grep='grep --color=auto'
#alias fgrep='fgrep --color=auto'
#alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^_ Aller ligne
```

```
user2@debian: ~/Bureau
GNU nano 5.4 /home/user2/.bashrc *
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@h: \w\a\]$PS1"
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors ->
    alias ls='ls --color=auto'
    alias la='ls -la'
    alias update='sudo apt-get update'
    alias upgrade='sudo apt-get upgrade'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    #alias grep='grep --color=auto'
    #alias fgrep='fgrep --color=auto'
    #alias egrep='egrep --color=auto'

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter    ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne
```

```
user2@debian: ~
user2@debian:~/Bureau$ nano ~/.bashrc
user2@debian:~/Bureau$ source ~/.bashrc
bash: ~/.bashrc: Aucun fichier ou dossier de ce type
user2@debian:~/Bureau$ cd ~
user2@debian:~$ source ~/.bashrc
user2@debian:~$ update
Atteint :1 http://deb.debian.org/debian bullseye InRelease
Réception de :2 http://security.debian.org/debian-security bullseye-security InR
elease [48,4 kB]
Atteint :3 https://dl.google.com/linux/chrome/deb stable InRelease

Atteint :4 http://deb.debian.org/debian bullseye-updates InRelease

48,4 ko réceptionnés en 1s (84,4 ko/s)
Lecture des listes de paquets... Fait
user2@debian:~$
```

- Afficher les variables d'environnement

env

```
user2@debian: ~  
user2@debian:~$ env  
SHELL=/bin/bash  
SESSION_MANAGER=local/debian:@/tmp/.ICE-unix/1380,unix/debian:/tmp/.ICE-unix/1380  
QT_ACCESSIBILITY=1  
COLORTERM=truecolor  
SSH_AGENT_LAUNCHER=openssh  
XDG_MENU_PREFIX=gnome-  
GNOME_DESKTOP_SESSION_ID=this-is-deprecated  
SSH_AUTH_SOCK=/run/user/1002/keyring/ssh  
XMODIFIERS=@im=ibus  
DESKTOP_SESSION=gnome  
GTK_MODULES=gail:atk-bridge  
PWD=/home/user2  
LOGNAME=user2  
XDG_SESSION_DESKTOP=gnome  
XDG_SESSION_TYPE=wayland  
XAUTORITY=/run/user/1002/.mutter-Xwaylandauth.A95LS1  
GDM_LANG=fr_FR.UTF-8  
HOME=/home/user2  
USERNAME=user2  
IM_CONFIG_PHASE=1  
LANG=fr_FR.UTF-8  
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzt=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31
```

- Ajouter à votre Path le chemin **"/home/"votre utilisateur"/Bureau"**

env

Copier le PATH

nano ~/.bashrc

Aller tout en bas du fichier et écrire **export PATH=(coller le PATH copié):/home/"votre utilisateur"/Bureau**

Taper Ctrl+O pour enregistrer les modifications puis Ctrl+X pour sortir de nano.

source ~/.bashrc

```
user2@debian: ~
GNOME_TERMINAL_SCREEN=/org/gnome/Terminal/screen/aac0b591_e30d_4b41_bb78_fcd9294b0e0e
GNOME_SETUP_DISPLAY=:1
XDG_SESSION_CLASS=user
TERM=xterm-256color
USER=user2
GNOME_TERMINAL_SERVICE=:1.81
DISPLAY=:0
SHLVL=1
QT_IM_MODULE=ibus
XDG_RUNTIME_DIR=/run/user/1002
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
GDMSESSION=gnome
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1002/bus
OLDPWD=/home/user2/Bureau
_=/usr/bin/env
user2@debian:~$ nano ~/.bashrc
```

```
GNU nano 5.4 /home/user2/.bashrc *
#alias la='ls -A'
#alias l='ls -CF'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier  ^_ Aller ligne

U
U
```

```
user2@debian: ~  
QT_IM_MODULE=ibus  
XDG_RUNTIME_DIR=/run/user/1002  
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games  
GDMSESSION=gnome  
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1002/bus  
OLDPWD=/home/user2/Bureau  
_=/usr/bin/env  
user2@debian:~$ nano ~/.bashrc  
user2@debian:~$ source .bashrc  
user2@debian:~$
```

```
user2@debian: ~  
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/user2/Bureau  
GDMSESSION=gnome  
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/1002/bus  
OLDPWD=/home/user2/Bureau  
_=/usr/bin/env  
user2@debian:~$
```

JOB 6

Vous devez télécharger l'archive suivante et la désarchiver seulement avec le terminal.

Je n'ai pas pu télécharger l'archive avec le terminal mais j'ai pu la désarchiver:

`tar -xzvf 'Copie de Ghost in the Shell.tar.gz'`

tar est un outil d'archivage qui fut porté sur les premières versions d'UNIX.

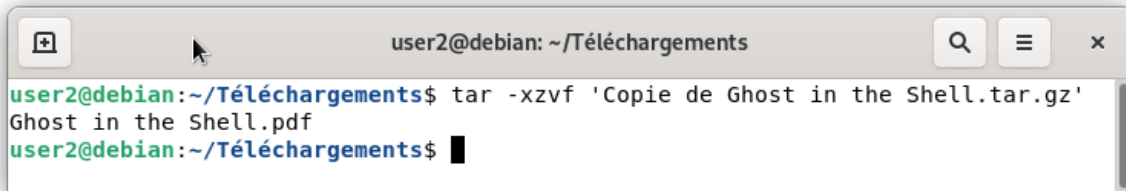
La commande tar permet donc d'archiver ou de désarchiver des répertoires et des fichiers de façon optimale et ce, sans interface graphique, au contraire, par exemple, de file-roller !

x : Pour extraire l'archive

z : Zip, indique la commande tar qui crée un fichier de goudron à l'aide de GZIP

v : Pour activer le mode "verbeux" (c'est à dire "bavard", en affichant ce qu'il fait)

f : Créer des archives avec le nom de fichier donné



```
user2@debian: ~/Téléchargements
user2@debian:~/Téléchargements$ tar -xzvf 'Copie de Ghost in the Shell.tar.gz'
Ghost in the Shell.pdf
user2@debian:~/Téléchargements$
```

JOB 7

- Créer un fichier “une_commande.txt” avec le texte suivant “Je suis votre fichier texte”

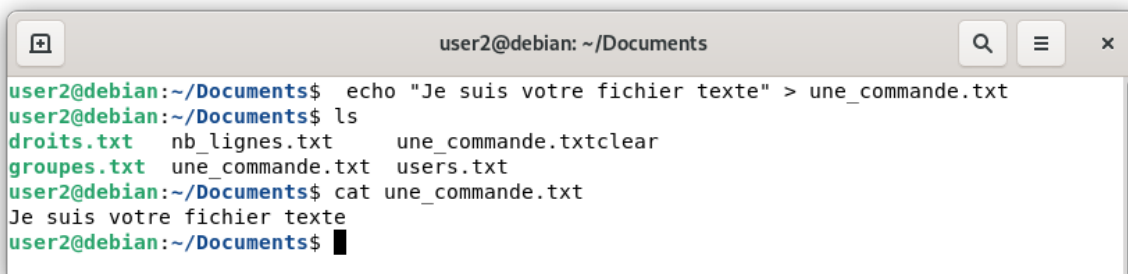
`echo "Je suis votre fichier texte" > une_commande.txt`

`echo` : L'utilisation basique est de donc de saisir le texte qui va ensuite s'afficher dans le terminal ou via un script shell.

La syntaxe : `echo <OPTIONS> <TEXTE>`

`>` : sert à copier le résultat de la commande précédente

`une_commande.txt` : dossier dans lequel le résultat sera copié



```
user2@debian: ~/Documents
user2@debian:~/Documents$ echo "Je suis votre fichier texte" > une_commande.txt
user2@debian:~/Documents$ ls
droits.txt  nb_lignes.txt  une_commande.txt  clear
groupes.txt une_commande.txt users.txt
user2@debian:~/Documents$ cat une_commande.txt
Je suis votre fichier texte
user2@debian:~/Documents$
```

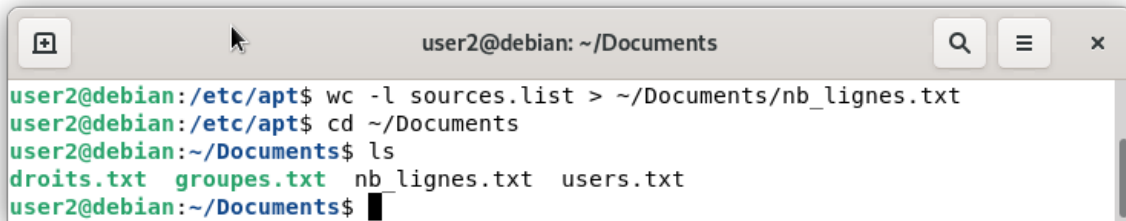
- Compter le nombre de lignes présentes dans votre fichier de source apt et les enregistrer dans un fichier nommé “nb_lignes.txt”

`wc -l sources.list > ~/Documents/nb_lignes.txt`

`wc -l sources.list` : pour compter le nombre de lignes dans le dossier sources.list

`>` : sert à copier le résultat de la commande

`~/Documents/nb_lignes.txt` : chemin du fichier dans lequel les résultats vont être collés

A terminal window titled 'user2@debian: ~/Documents'. The user runs 'wc -l sources.list > ~/Documents/nb_lignes.txt', then 'cd ~/Documents', and finally 'ls'. The output of 'ls' shows 'droits.txt', 'groupes.txt', 'nb_lignes.txt', and 'users.txt'.

```
user2@debian:~/Documents$ wc -l sources.list > ~/Documents/nb_lignes.txt
user2@debian:~/Documents$ cd ~/Documents
user2@debian:~/Documents$ ls
droits.txt  groupes.txt  nb_lignes.txt  users.txt
user2@debian:~/Documents$
```

- Afficher le contenu du fichier source apt et l'enregistrer dans un autre fichier appelé "save_sources"

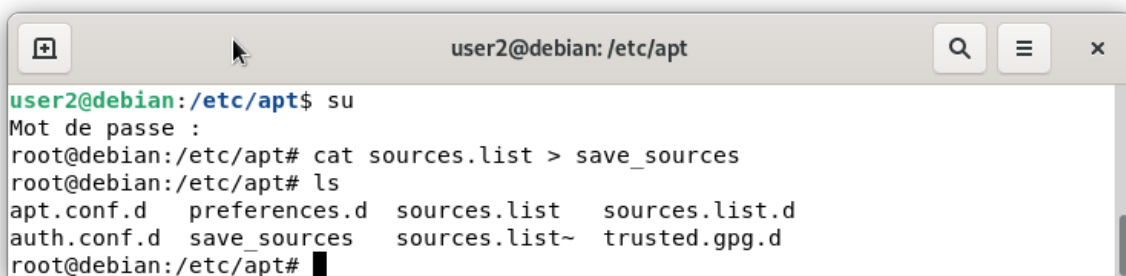
Se mettre en root avant d'écrire cette commande

`cat sources.list > save_sources`

`cat` : permet d'afficher le contenu du fichier

`>` : sert à copier le résultat de la commande

`save_sources` : fichier dans lequel les résultats vont être collés

A terminal window titled 'user2@debian: /etc/apt'. The user runs 'su' to become root, then 'cat sources.list > save_sources', and finally 'ls'. The output of 'ls' shows 'apt.conf.d', 'preferences.d', 'sources.list', 'sources.list.d', 'auth.conf.d', 'save_sources', 'sources.list~', and 'trusted.gpg.d'.

```
user2@debian:/etc/apt$ su
Mot de passe :
root@debian:/etc/apt# cat sources.list > save_sources
root@debian:/etc/apt# ls
apt.conf.d  preferences.d  sources.list  sources.list.d
auth.conf.d  save_sources  sources.list~  trusted.gpg.d
root@debian:/etc/apt#
```

- Faites une recherche des fichiers commençant par "." tout en cherchant le mot alias qui sera utilisé depuis un fichier

Se déplacer dans le dossier /home/(nom de l'utilisateur) et taper cette commande

`grep alias /-d .*`

`grep` est une commande qui permet de faire des recherches de chaînes de caractères dans un flux de texte (fichier ou sortie d'une autre commande, par exemple) et elle accepte les expressions régulières. Il en existe plusieurs variantes, par exemple : `agrep`, qui permet la recherche de chaînes approximatives.

`alias /-d .*` : chaîne de caractères à rechercher

```
user2@debian: ~  
user2@debian:~$ grep alias /-d .*  
grep: /-d: Aucun fichier ou dossier de ce type  
grep: .: est un dossier  
grep: ..: est un dossier  
.bashrc:# enable color support of ls and also add handy aliases  
.bashrc: alias ls='ls --color=auto'  
.bashrc: alias la='ls -la'  
.bashrc: alias update='sudo apt-get update'  
.bashrc: alias upgrade='sudo apt-get upgrade'  
.bashrc: #alias dir='dir --color=auto'  
.bashrc: #alias vdir='vdir --color=auto'  
.bashrc: #alias grep='grep --color=auto'  
.bashrc: #alias fgrep='fgrep --color=auto'  
.bashrc: #alias egrep='egrep --color=auto'  
.bashrc:# some more ls aliases  
.bashrc:#alias ll='ls -l'  
.bashrc:#alias la='ls -A'  
.bashrc:#alias l='ls -CF'  
.bashrc:# ~/.bash_aliases, instead of adding them here directly.  
.bashrc:if [ -f ~/.bash_aliases ]; then  
.bashrc: . ~/.bash_aliases  
grep: .cache: est un dossier  
grep: .config: est un dossier  
grep: .gnupg: est un dossier  
grep: .local: est un dossier  
grep: .mozilla: est un dossier  
grep: .pki: est un dossier  
grep: .ssh: est un dossier  
user2@debian:~$
```

POUR ALLER PLUS LOIN

- Installer la commande tree

`sudo apt install tree`

`apt install` : apt permet d'installer des logiciels sur les distributions à base de Debian depuis les dépôts.

`tree` : nom du logiciel à installer

```
user2@debian: ~  
user2@debian:~$ sudo apt install tree  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
tree est déjà la version la plus récente (1.8.0-1+b1).  
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.  
user2@debian:~$
```

- Lancer la commande **tree** en arrière-plan qui aura pour but d'afficher toute l'arborescence en de votre / en enregistrant le résultat dans un fichier "tree.save"

```
cp tree > tree.save |cat tree.save &
```

cp : copier des fichiers ou un groupe de fichiers ou de répertoires.

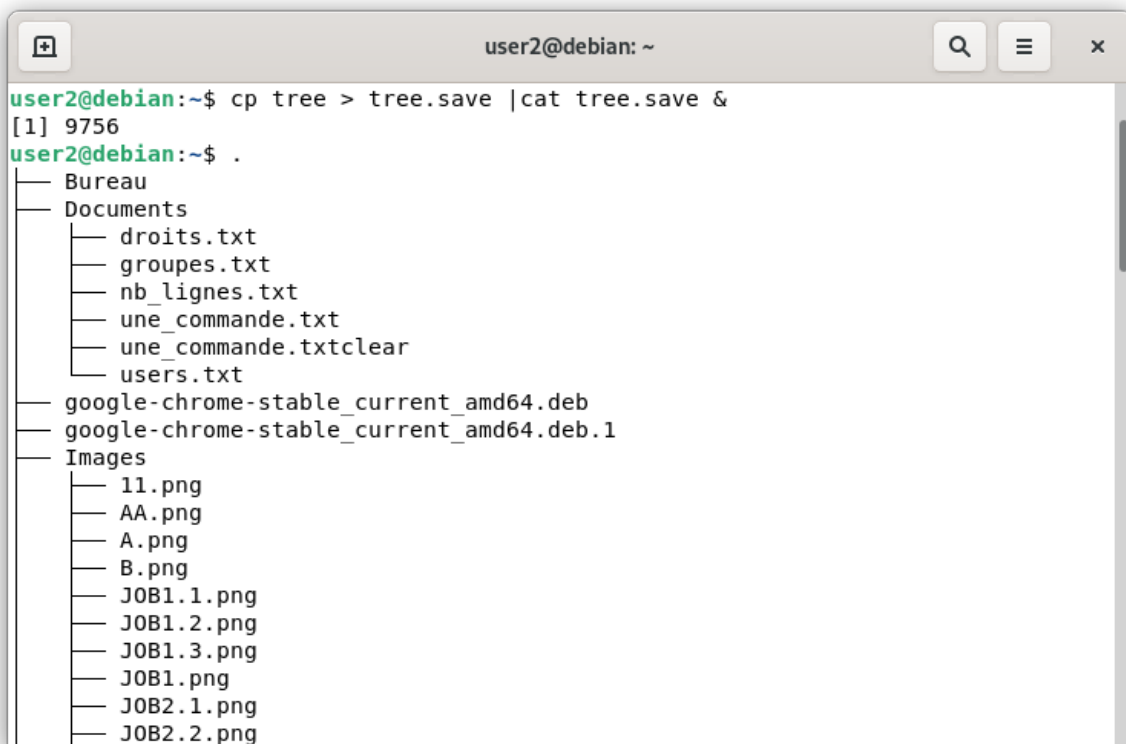
tree : pour lancer la commande tree et donc afficher l'arborescence à partir du fichier dans lequel on se trouve.

> : sert à copier le résultat de la commande

tree.save : fichier dans lequel les résultats vont être collés

cat : afficher le contenu d'un fichier sur le terminal.

Le **&** sert à lancer une commande en arrière plan. Ça veut dire que le shell lance la commande et n'attend pas qu'elle se termine.

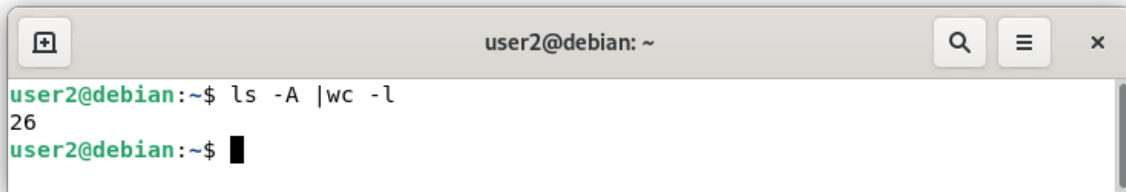


```
user2@debian: ~  
user2@debian:~$ cp tree > tree.save |cat tree.save &  
[1] 9756  
user2@debian:~$ .  
— Bureau  
— Documents  
  — droits.txt  
  — groupes.txt  
  — nb_lignes.txt  
  — une_commande.txt  
  — une_commande.txtclear  
  — users.txt  
— google-chrome-stable_current_amd64.deb  
— google-chrome-stable_current_amd64.deb.1  
— Images  
  — 11.png  
  — AA.png  
  — A.png  
  — B.png  
  — JOB1.1.png  
  — JOB1.2.png  
  — JOB1.3.png  
  — JOB1.png  
  — JOB2.1.png  
  — JOB2.2.png
```

- lister les éléments présents dans le dossier courant et utiliser directement le résultat de votre première commande pour compter le nombre d'éléments trouvés

```
ls -A |wc -l
```

```
ls -a |wc -l
```

A terminal window titled 'user2@debian: ~' with search, menu, and close buttons. The prompt is 'user2@debian:~\$'. The command 'ls -A | wc -l' has been entered and executed, resulting in the output '26'. The prompt is now 'user2@debian:~\$' with a cursor.

```
user2@debian:~$ ls -A | wc -l
26
user2@debian:~$
```

Quelques explications nécessaires :

- Dans un premier temps on liste les fichiers et répertoires contenus dans le répertoire en cours en utilisant la commande `ls` en affichant un fichier par ligne (option `"-l"`) et en affichant tous les fichiers y compris les fichiers cachés (option `"-A"`). L'option `-A` permet d'afficher tous les éléments présents dans le dossier mais en omettant les dossiers `"."` et `".."`, qui sont le dossier dans lequel on se trouve et son dossier parent. Si on veut aussi compter ces dossiers, on remplacera l'option `-A` par l'option `-a`.
- Dans un second temps on utilise le résultat de cette commande que l'on va traiter par une autre, c'est le rôle du pipe (`"|"`) qui signifie que l'on lance une seconde commande sur le résultat de la première
- Dans un troisième temps, on utilise la commande `wc` (qui compte les résultats) sur le résultat du `ls` en ne comptant que les lignes (option `"-l"`).

- Lancer une commande pour update vos paquets, si l'update réussit alors, vous devrez lancer un upgrade de vos paquets. Si l'update échoue, votre upgrade ne se lancera pas

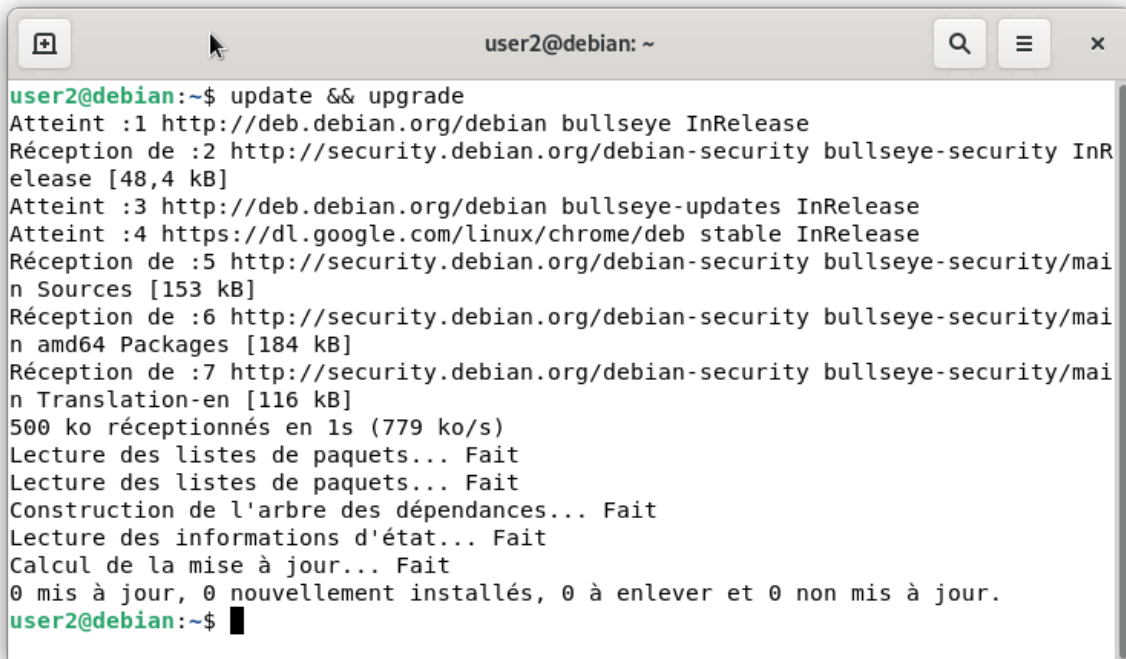
`update && upgrade`

La commande `&&` ne s'exécutera que si la première commande a été exécutée avec succès, c'est-à-dire que son état de sortie est zéro. Cet opérateur peut être utilisé pour vérifier si la première commande a été exécutée avec succès.

La commande `;` est utilisée pour exécuter plusieurs commandes en une seule fois. Plusieurs commandes peuvent être enchaînées à l'aide de cet opérateur. L'exécution de la commande qui succède à cet opérateur s'exécutera toujours après l'exécution de la commande qui le précède, quel que soit l'état de sortie de la commande précédente. Les commandes s'exécutent toujours de manière séquentielle. Les commandes séparées par un opérateur point-virgule sont exécutées séquentiellement, le shell attend que chaque commande se termine à son tour. L'état de retour est l'état de sortie de la dernière commande exécutée.

La commande `|` permet de lancer une seconde commande sur le résultat de la première.

`update` et `upgrade` sont les alias de `sudo apt update` et de `sudo apt upgrade` que nous avons créés au début de ce projet.



```
user2@debian: ~  
user2@debian:~$ update && upgrade  
Atteint :1 http://deb.debian.org/debian bullseye InRelease  
Réception de :2 http://security.debian.org/debian-security bullseye-security InRelease [48,4 kB]  
Atteint :3 http://deb.debian.org/debian bullseye-updates InRelease  
Atteint :4 https://dl.google.com/linux/chrome/deb stable InRelease  
Réception de :5 http://security.debian.org/debian-security bullseye-security/main Sources [153 kB]  
Réception de :6 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [184 kB]  
Réception de :7 http://security.debian.org/debian-security bullseye-security/main Translation-en [116 kB]  
500 ko réceptionnés en 1s (779 ko/s)  
Lecture des listes de paquets... Fait  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Calcul de la mise à jour... Fait  
0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.  
user2@debian:~$
```