



Job 01

1. Comment ajouter des options à une commande ?

Pour ajouter des options à une **commande**, il faut choisir le nom du programme que l'on souhaite exécuter (par exemple: ls, man, cat etc.). Cela indiquera ce que nous voulons faire. On doit choisir l'**option** qui est souvent précédée d'un tiret court (par exemple -l) ou long (par exemple: -- long). Il faut également déterminer l'**argument** qui est l'élément sur lequel la commande doit agir. (Par exemple cat /home/vanny/.bashrc).

Les autres éléments (les **redirections**, les **tuyaux**, les **variables** etc.) peuvent être combinés afin d'effectuer des tâches plus avancées dans Unix.

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

Les deux syntaxes principales d'écriture des options pour une commande sont les suivantes:

Syntaxe courte: dans cette syntaxe, les options sont principalement représentées par des lettres ou des caractères courts, précédés d'un tiret court (tiret du 6, donc -).

Chaque lettre ou caractère court représente une option spécifique.

Syntaxe longue : dans cette syntaxe, les options sont souvent représentées par des noms descriptifs complets, précédés de deux tirets (tiret du 6, donc --).

3. Afficher le manuel de la commande ls

Pour afficher le manuel de la commande ls, il faut ouvrir le cmd, puis écrire **man ls**

- *man* est l'abréviation de manual en anglais. Cette commande fait référence aux pages de manuel, qui sont des documents de référence décrivant l'utilisation, la syntaxe et les options des commandes système et des programmes.

- *ls* est l'abréviation de list en anglais. C'est une commande qui permet de lister le contenu d'un répertoire.

```
vanny@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
Manual page ls(1) line 1 (press h for help or q to quit)
```

4. Afficher les fichiers cachés du home de votre utilisateur

La commande **ls -a** permet d'afficher une liste détaillée de tous les fichiers et répertoires présents dans le répertoire /home.

- **ls** est une commande pour lister le contenu d'un répertoire.
- **-a** est une option qui demande à la commande **ls** d'afficher tous les fichiers, y compris les fichiers cachés.
- **~** le tilde est utilisé pour représenter le répertoire personnel de l'utilisateur (équivalent à "/home/nom_utilisateur"). Il n'est donc pas nécessaire de rajouter */home* après **ls -a**

```
vanny@debian:~$ ls -a
.          .bashrc  Documents Images      Musique    Téléchargements
..         Bureau  .face    .lessht   .profile   Vidéos
.bash_history .cache  .face.icon .local    Public
.bash_logout .config .gnupg    Modèles   .ssh
```

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

Pour afficher les fichiers cachés ainsi que les informations sur les droits sous forme de liste dans un terminal Linux, on utilise la commande **ls -al**

- **ls** est une commande pour lister le contenu d'un répertoire.
- **-a** est une option qui demande à la commande **ls** d'afficher tous les fichiers, y compris les fichiers cachés.
- **-l** est une option qui demande à la commande **ls** d'afficher les résultats sous forme de liste détaillée. Cela signifie qu'elle affiche des informations telles que les autorisations, le propriétaire, le groupe, la taille, la date de modification et le nom des fichiers.

vanny@debian:~\$ ls -al

```
total 128
drwx----- 15 vanny vanny 4096 21 sept. 10:49 .
drwxr-xr-x  3 root  root  4096 18 sept. 17:10 ..
-rw-----  1 vanny vanny 1307 20 sept. 14:59 .bash_history
-rw-r--r--  1 vanny vanny  220 18 sept. 17:10 .bash_logout
-rw-r--r--  1 vanny vanny 3526 18 sept. 17:10 .bashrc
drwxr-xr-x  2 vanny vanny 4096 18 sept. 17:13 Bureau
drwx-----  8 vanny vanny 4096 19 sept. 10:51 .cache
drwx----- 10 vanny vanny 4096 19 sept. 10:51 .config
drwxr-xr-x  2 vanny vanny 4096 18 sept. 17:13 Documents
-rw-r--r--  1 User1 vanny   14 20 sept. 16:38 droits.txt
-rw-r--r--  1 vanny vanny 5290 18 sept. 17:10 .face
lrwxrwxrwx  1 vanny vanny    5 18 sept. 17:10 .face.icon -> .face
drwx-----  2 vanny vanny 4096 20 sept. 15:17 .gnupg
-r--r--r--  1 vanny vanny   15 20 sept. 16:44 groupes.txt
drwxr-xr-x  2 vanny vanny 4096 18 sept. 17:13 Images
-rw-r--r--  1 vanny vanny 22832 21 sept. 09:41 index.html
-rw-----  1 vanny vanny   20 21 sept. 10:49 .lessht
drwx-----  4 vanny vanny 4096 18 sept. 17:13 .local
drwxr-xr-x  2 vanny vanny 4096 18 sept. 17:13 Modèles
drwxr-xr-x  2 vanny vanny 4096 18 sept. 17:13 Musique
-rw-r--r--  1 vanny vanny  807 18 sept. 17:10 .profile
```

Job 02

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

Pour lire le contenu d'un fichier il faut utiliser la commande **cat nom_du_fichier**.

Par exemple pour lire le `bashrc`, il faut écrire **cat /home/vanny/.bashrc**

- `cat` est une commande permettant de concaténer des fichiers ainsi que d'afficher leur contenu sur la sortie standard.

```
vanny@debian:~$ cat /home/vanny/.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;;
```

2. Afficher les 10 premières lignes du fichier “.bashrc”

La commande utilisée est **head /home/vanny/.bashrc**

- `head` est la commande pour afficher les premières lignes d'un fichier.
- `/home/vanny/.bashrc` est le chemin du fichier `.bashrc` dans notre répertoire personnel.

```
vanny@debian:~$ head /home/vanny/.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
```

3. Afficher les 10 dernières lignes du fichier “.bashrc”

La commande utilisée est **tail /home/vanny/.bashrc**

- `tail` est une commande pour afficher les dernières lignes d'un fichier.
- `/home/vanny/.bashrc` est le chemin du fichier `.bashrc` dans notre répertoire personnel.

Pour l'exercice 1 & 2, `tail` et `head` affichent par défaut les 10 premières lignes de chaque fichier, il n'est donc pas nécessaire de rajouter `-n 10` (ce qui revient à dire `head -n 10 /home/vanny/.bashrc` ou `tail -n 10 /home/vanny/.bashrc`)

4. Afficher les 20 premières lignes du fichier “.bashrc”

La commande utilisée est **head -n 20 /home/vanny/.bashrc**

- *head* est une commande pour afficher les premières lignes d'un fichier.
- *-n 20* est utilisée pour spécifier un nombre ou une limite, dans notre cas 20 lignes.
- */home/vanny/.bashrc* est le chemin du fichier *.bashrc* dans notre répertoire personnel.

```
vanny@debian:~$ head -n 20 /home/vanny/.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
```

5. Afficher les 20 dernières lignes du fichier “.bashrc”

La commande utilisée est **tail -n 20 /home/vanny/.bashrc**

- *head* est une commande pour afficher les premières lignes d'un fichier.
- *-n 20* est utilisée pour spécifier un nombre ou une limite. Dans notre cas 20 lignes.
- */home/vanny/.bashrc* est le chemin du fichier *.bashrc* dans notre répertoire personnel.
- *-n 20*

Job 03

1. Installer le paquet "cmatrix"

Il faut utiliser la commande suivante: **sudo apt install cmatrix** .Puis mettre le mot de passe.

- *sudo* signifie *SuperUser DO*. Cette commande est utilisée pour exécuter la commande en tant qu'administrateur (superutilisateur) ou avec des privilèges élevés. Cela nous permet d'effectuer des opérations qui nécessitent des droits d'administration.

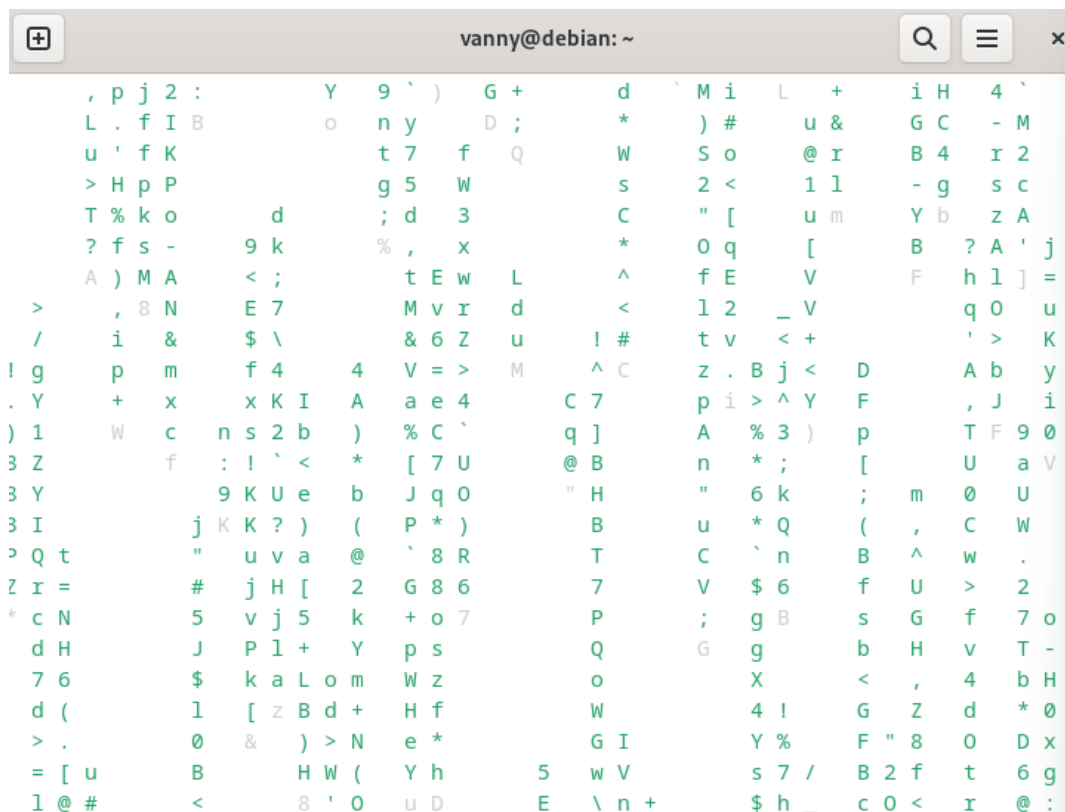
- *apt* correspond au gestionnaire de paquets avancé pour les systèmes basés sur Debian.

Il est utilisé pour gérer l'installation, la mise à jour et la suppression de logiciels.

```
vanny@debian:~$ sudo apt install cmatrix
[sudo] Mot de passe de vanny :
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.
```

2. Lancer le paquet que vous venez d'installer

Pour ouvrir le paquet il faut utiliser **cmatrix** C'est un petit utilitaire qui permet d'afficher la Matrix dans notre terminal. Pour quitter la Matrix, on utilisera les touches *Ctrl + C*



```
vanny@debian: ~
, p j 2 :      Y 9 ` )  G +      d ` M i  L  +      i H 4 `
L . f I B      o n y      D ;      *      ) #      u &      G C  - M
u ' f K      t 7  f  Q      W      S o      @ r      B 4  r 2
> H p P      g 5  W      s      2 <      1 l      - g  s c
T % k o      d      ; d 3      C      " [      u m      Y b  z A
? f s -      9 k      % ,  x      *      0 q      [      B  ? A ' j
A ) M A      < ;      t E w  L      ^      f E      V      F  h l ] =
>      , 8 N      E 7      M v r  d      <      l 2  _ V      q 0  u
/      i  &      $ \      & 6 Z  u      ! #      t v      < +      ' >  K
! g      p  m      f 4      4  V = >  M      ^ C      z . B j <  D      A b  y
. Y      +  x      x K I  A  a e 4      C 7      p i > ^ Y  F      , J  i
) 1      W      c      n s 2 b  )  % C `      q ]      A  % 3 )  p      T F 9 0
3 Z      f      : ! ` <      *      [ 7 U      @ B      n      * ;      [      U  a V
3 Y      9 K U e  b  J q 0      " H      "      6 k      ;      m  0  U
3 I      j K K ? )  (  P * )      B      u      * Q      (      ,  C  W
> Q t      "      u v a  @ ` 8 R      T      C      ` n      B  ^  w  .
Z r =      #      j H [  2  G 8 6      7      V      $ 6      f  U  >  2
* c N      5  v j 5  k  + o 7      P      ;      g B      s  G  f  7 o
d H      J  P l +  Y  p s      Q      G      g      b  H  v  T -
7 6      $  k a L o m  W z      o      X      <      ,  4  b  H
d (      l  [ z B d +  H f      W      4 !      G  Z  d  * 0
> .      0  &      ) > N  e *      G I      Y %      F " 8  0  D x
= [ u      B      H W (  Y h      5  w V      s 7 /  B 2 f  t  6 g
l @ #      <      8 ' 0  u D      E  \ n +      $ h _  c 0 <  r  @ :
```

3. Mettre à jour son gestionnaire de paquets

J'utilise la commande **sudo apt update** pour mettre à jour la liste des paquets disponibles dans les dépôts de logiciels configurés sur notre système.

- *sudo* est utilisé pour exécuter la commande en tant qu'administrateur ou superutilisateur.
- *apt* est la gestionnaire de paquets permet d'installer, de mettre à jour et de gérer les paquets logiciels sur notre système.
- *update* est l'argument de la commande qui indique à *apt* de mettre à jour la liste des paquets disponibles.

```
vanny@debian:~$ sudo apt update
[sudo] Mot de passe de vanny :
Atteint :1 http://deb.debian.org/debian bookworm InRelease
Atteint :2 http://security.debian.org/debian-security bookworm-security InRelease
Réception de :3 http://deb.debian.org/debian bookworm-updates InRelease [52,1 kB]
52,1 ko réceptionnés en 1s (85,2 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.
```

4. Mettre à jour ses différents logiciels

Pour mettre à jour les différents logiciels il faut utiliser la commande: **sudo apt-get upgrade**

- *sudo* est utilisé pour exécuter la commande en tant qu'administrateur ou superutilisateur.
- *apt-get* est un outil de gestion de paquets permettant d'installer, de mettre à jour et de supprimer des logiciels sur notre système.
- *upgrade* est l'argument de la commande qui indique que l'on souhaite effectuer une mise à jour des logiciels déjà installés. *upgrade* met à jour toutes les applications et les packages du système vers leurs dernières versions disponibles dans les dépôts de logiciels configurés sur notre système.

```
vanny@debian:~$ sudo apt-get 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.
```

5. Télécharger les internets : Google

Pour effectuer des recherches Google à partir de la ligne de commande, on peut utiliser la commande `wget` pour envoyer des requêtes à Google suivi de l'URL: **wget**

<https://www.google.com>

- *wget* est une contraction de World Wide Web et *get* (obtenir), ce qui signifie littéralement "obtenir des fichiers depuis le Web"

```
vanny@debian:~$ wget https://www.google.com
--2023-09-21 09:41:01-- https://www.google.com/
Résolution de www.google.com (www.google.com)... 172.217.171.196, 2a00:1450:4006:808::2004
Connexion à www.google.com (www.google.com)|172.217.171.196|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : non indiqué [text/html]
Sauvegarde en : « index.html »

index.html          [ <=>          ] 22,30K  --.-KB/s    ds 0,002s

2023-09-21 09:41:02 (14,2 MB/s) - « index.html » sauvegardé [22832]
```

6. Redémarrer votre machine

Pour redémarrer notre machine il faut utiliser **sudo reboot**

- *reboot* est la commande qui indique au système d'exploitation de redémarrer l'ordinateur. Toutes les opérations et programmes en cours seront alors interrompus et fermés, et le système sera relancé depuis le début.

7. Éteindre votre machine

Pour éteindre notre machine, on peut choisir de le faire avec délai ou non:

poweroff permet d'éteindre la machine immédiatement.

sudo shutdown -h now permet d'éteindre la machine avec un délai.

Job 04

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

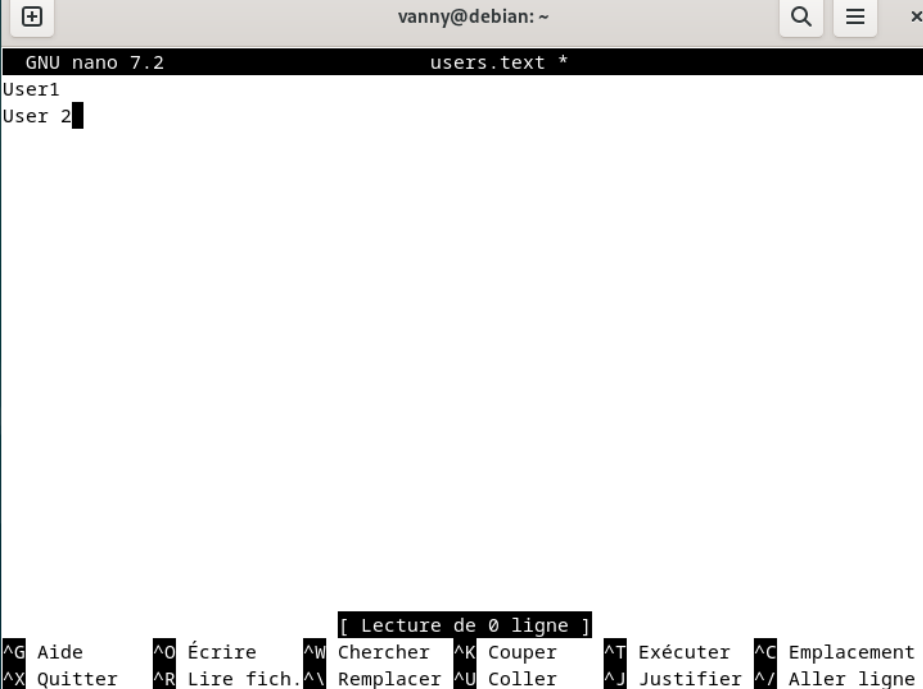
La commande **nano users.txt** permet de créer un fichier users.txt

Il faut ensuite entrer User1, saut à la ligne et User 2.

Puis ctrl+X pour quitter, O pour Oui et la touche.

- *nano* est un éditeur de texte utilisé sur les systèmes Unix et Linux pour éditer des fichiers texte directement dans un terminal.

```
vanny@debian:~$ nano users.txt
```



The screenshot shows a terminal window with the nano text editor open. The title bar indicates 'vanny@debian: ~'. The editor's status bar shows 'GNU nano 7.2' and 'users.txt *'. The content of the file is 'User1' followed by a newline and 'User 2'. The bottom of the screen displays a list of keyboard shortcuts for nano, such as ^G Aide, ^O Écrire, ^W Chercher, etc.

2. Créer un groupe appelé “Plateformeurs”

Pour créer un groupe il faut utiliser la commande **sudo groupadd Plateformeurs**

- *sudo* est une commande qui permet à un utilisateur d'exécuter des commandes avec les privilèges d'administration ou superutilisateur.

- *groupadd* est une commande pour ajouter un nouveau groupe d'utilisateurs au système d'exploitation.

```
vanny@debian:~$ sudo groupadd Plateformeurs
```

```
[sudo] Mot de passe de vanny :
```

3. Créer un utilisateur appelé “User1”

4. Créer un utilisateur appelé “User2”

La commande **sudo useradd User1** permet de créer un nouvel utilisateur appelé “User1”

Pour créer un “User2”, il faudra donc utiliser la commande **sudo useradd User2**

- *useradd* est une commande pour ajouter un nouvel utilisateur à un système.

```
vanny@debian:~$ sudo useradd User1
```

```
vanny@debian:~$ sudo useradd User2
```

5. Ajouter “User2” au groupe Plateformeurs

Pour ajouter User2 dans le groupe Plateformeurs il faut utiliser la commande **sudo usermod -aG Plateformeurs User2**

- *sudo* est une commande qui permet à un utilisateur d'exécuter des commandes avec les privilèges d'administration ou superutilisateur.
- *usermod* : est une commande pour modifier les paramètres d'un utilisateur existant.
- *-a* ajoute l'utilisateur au groupe sans le supprimer de ses groupes actuels.
- *-G* Spécifie le groupe auquel ajouter l'utilisateur.

```
vanny@debian:~$ sudo usermod -aG Plateformeurs User2
```

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

Il faut utiliser la commande **cp users.txt droits.txt**

- *cp* est la syntaxe de base pour copier un fichier texte.
- *users.txt* est le nom du fichier que l'on souhaite copier.
- *droits.txt* est le nom que l'on souhaite donner au nouveau fichier copié.

```
vanny@debian:~$ cp users.txt droits.txt
```

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

Il faut utiliser la commande **cp users.txt groupes.txt**

- *cp* est la syntaxe de base pour copier un fichier texte.
- *users.txt* est le nom du fichier que l'on souhaite copier.
- *groupes.txt* est le nom que l'on souhaite donner au nouveau fichier copié.

```
vanny@debian:~$ cp users.txt groupes.txt
```

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

Pour changer de propriétaire il faut utiliser **sudo chown User1 droits.txt**

- *sudo* est une commande qui permet à un utilisateur d'exécuter des commandes avec les privilèges d'administration ou superutilisateur.
- *chown* est une commande pour changer le propriétaire d'un fichier ou d'un répertoire. Le terme *chown* est une contraction de "change owner".

```
vanny@debian:~$ sudo chown User1 droits.txt
```

9. Changer les droits du fichier "droits.txt" pour que "User2" ai accès seulement en lecture

```
vanny@debian:~$ ls -ltr
total 92
drwxr-xr-x 2 vanny vanny 4096 18 sept. 17:13 Vidéos
drwxr-xr-x 2 vanny vanny 4096 18 sept. 17:13 Public
drwxr-xr-x 2 vanny vanny 4096 18 sept. 17:13 Modèles
drwxr-xr-x 2 vanny vanny 4096 18 sept. 17:13 Images
-rwxr-xr-x 1 vanny vanny 38 20 sept. 10:43 script.sh
-rw-r--r-- 1 vanny vanny 0 20 sept. 15:18 vanny.txt
-rw-r--r 1 User1 vanny 14 20 sept. 16:38 droits.txt
```

"-rw-r--r--" signifie qu'il s'agit d'un fichier avec les droits suivants :

Le propriétaire peut le lire et le modifier.

Les membres du groupe peuvent le lire mais ne peuvent pas le modifier.

Les autres utilisateurs peuvent le lire mais ne peuvent pas le modifier.

En conclusion le User2 a déjà seulement accès à la lecture du fichier "droits.txt".

10. Changer les droits du fichier "groupes.txt" pour que les utilisateurs puissent accéder au fichier en lecture uniquement

J'ai utilisé la commande suivante: **sudo chmod u=r,go=r groupes.txt**

- *sudo* est une commande qui permet à un utilisateur d'exécuter des commandes avec les privilèges d'administration ou superutilisateur.
- *chmod* est la commande pour changer les autorisations d'un fichier.
- *u=r* définit les autorisations pour le propriétaire du fichier (*u* pour utilisateur) en lecture seule (*r* pour lecture).
- *go=r* définit les autorisations pour le groupe et les autres utilisateurs (le *g* pour groupe et le *o* pour autres) en lecture seule (*r* pour lecture).
- *groupes.txt*: C'est le nom du fichier dont vous souhaitez changer les autorisations.

```
vanny@debian:~$ sudo chmod u=r,go=r groupes.txt
```

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

J'ai utilisé la commande suivante: **sudo chmod g+rw groupes.txt**

```
vanny@debian:~$ sudo chmod -R g+rw groupes.txt
```

- *sudo* est une commande qui permet à un utilisateur d'exécuter des commandes avec les privilèges d'administration ou superutilisateur.
- *chmod* est la commande pour changer les autorisations d'un fichier.
- *g* désigne le groupe associé au fichier.
- *+rw* indique que les permissions du groupe doivent être modifiées en ajoutant la possibilité de lire (*r*) et d'écrire (*w*) le fichier. Cela signifie que les membres du groupe auront désormais la permission de lire et de modifier le fichier "groupes.txt".
- *groupes.txt* 'est le nom du fichier sur lequel on souhaite appliquer les modifications de permissions.

1. Ajouter un alias qui permettra de lancer la commande “ls -la” en tapant “la”

Les alias sont des substitutions abrégées de commandes répétitives et/ou longues à taper dans la console.

On peut définir nos alias dans le fichier `bashrc` qui se trouve dans notre Dossier Personnel. Pour se faire il faut taper la commande **nano .bashrc**. Ensuite juste après la ligne "some more ls aliases", on ajoutera **alias la="ls -la"**

```
# some more ls aliases
#alias ll='ls -l'
#alias la='ls -A'
#alias l='ls -CF'
alias la="ls -la"
alias update="apt-get update"
alias upgrade="apt-get upgrade"
```

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

On utilisera la commande **alias update="apt-get update"**

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

On utilisera la commande **alias upgrade="apt-get upgrade"**

4. Ajouter une variable d'environnement qui se nommera “USER” et qui sera égale à votre nom d'utilisateur

Nous allons écrire dans terminal: **export USER=vanny**

Pour vérifier que la variable a été définie correctement on utilise **echo \$USER**

```
vanny@debian:~$ export USER=vanny
vanny@debian:~$ echo $USER
vanny
```

- *export* est une commande pour rendre une variable d'environnement accessible aux processus lancés depuis le shell courant. Elle permet à la variable *USER* d'être utilisée par d'autres programmes qu'on exécute à partir du même terminal.
- *USER* est le nom de la variable d'environnement. Dans Unix, la variable "USER" est généralement utilisée pour stocker le nom de l'utilisateur actuellement connecté.
- *vanny* est la valeur qu'on attribue à la variable *USER*. Donc la valeur *vanny* est définie comme le nom d'utilisateur actuel. Cela signifie que, tant que cette variable est définie, lorsqu'on fait référence à *\$USER* dans des scripts ou des commandes, cela renverra *vanny* en tant que nom d'utilisateur.

5. Mettre à jour les modifications de votre bashrc dans votre shell actuel

Pour mettre à jour les modifications de notre bashrc on utilisera la commande : **source ~/.bashrc**

- *source* permet d'exécuter un fichier de script dans le contexte actuel du shell. Cela signifie que toutes les commandes et les variables définies dans le fichier seront interprétées et exécutées comme si elles avaient été saisies directement dans le terminal.

- **~/.bashrc** représente le répertoire personnel de l'utilisateur et *.bashrc* est le nom du fichier.

```
vanny@debian:~$ source ~/.bashrc
```

6. Afficher les variables d'environnement

Pour afficher les variables d'environnement il faut écrire dans la commande: **echo \$NOM_DE_LA_VARIABLE**. Nous allons prendre l'exemple de "LANG"

- *echo* est une commande pour afficher du texte à la sortie standard
- \$LANG est une variable d'environnement qui détermine la langue par défaut du système.

```
vanny@debian:~$ echo $LANG
fr FR.UTF-8
```

7. Ajouter à votre Path le chemin "/home/votre utilisateur/Bureau"

Il faut écrire dans la commande: **export PATH="/home/vanny/Bureau: \$PATH"**

- *export* permet de changer temporairement le chemin de PATH.

```
vanny@debian:~$ export PATH="/home/vanny/Bureau: $PATH"
```

```
vanny@debian:~$ echo $PATH
```

```
/home/vanny/Bureau: /home/'vanny'/Bureau: /home/'votre utilisateur'/Bureau: /usr
/local/bin: /usr/bin: /bin: /usr/local/games: /usr/games
```

Job 06

Nous avons d'une part téléchargé le fichier envoyé par Luc dans notre dossier Bureau.

Ensuite nous avons copier-coller le fichier "Ghost in the Shell" dans notre machine virtuelle, dans le dossier "Documents".

Pour désarchiver, nous avons utilisé **tar -xvf Documents/"Ghost in the Shell.tar"** pour extraire une archive .tar

```
vanny@debian:~$ tar -xvf Documents/"Ghost in the Shell.tar"
```

```
._Ghost in the Shell.pdf
```

```
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.quarantine » pour l'en-tête étendu a été ignoré
```

```
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.provenance » pour l'en-tête étendu a été ignoré
```

```
tar: Le mot clé inconnu « LIBARCHIVE.xattr.com.apple.macl » pour l'en-tête étendu a été ignoré
```

```
Ghost in the Shell.pdf
```

Job 07

echo "Je suis votre fichier texte" > une_commande.txt | wc -l < /etc/apt/sources.list > nb_lignes.txt | cat /etc/apt/sources.list > save_sources | grep -r "alias" .* | tee fichiers_alias.txt

```
vanny@debian:~$ echo "Je suis votre fichier texte" > une_commande.txt | wc -l < /etc/apt/sources.list > nb_lignes.txt | cat /etc/apt/sources.list > save_sources | grep -r "alias" .* | tee fichiers_alias.txt
.bash_history:alias la="ls -la"
.bashrc:# enable color support of ls and also add handy aliases
.bashrc:  alias ls='ls --color=auto'
.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:alias la="ls -la"
.bashrc:alias update="apt-get update"
.bashrc:alias upgrade="apt-get upgrade"
.bashrc:# ~/.bash_aliases, instead of adding them here directly.
.bashrc:if [ -f ~/.bash_aliases ]; then
.bashrc:  . ~/.bash_aliases
grep: .cache/mozilla/firefox/jl7pcba9.default-esr/startupCache/scriptCache.bin :
```

1. 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 permet de créer un fichier *une_commande.txt* avec le texte suivant “*Je suis votre fichier texte*”

> redirige la sortie de cette commande vers le fichier nommé *une_commande.txt*.

| le symbole pipe permet de chaîner des commandes ensemble pour effectuer des opérations plus complexes.

Donc le contenu “Je suis votre fichier texte” est écrit dans le fichier “une_commande.txt”.

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

wc -l < /chemin/vers/votre/fichier/source/apt > nb_lignes.txt

wc signifie *word count* et permet de compter le nombre de lignes dans le fichier spécifié par le chemin */chemin/vers/votre/fichier/source/apt*.

-l est une option de **wc** qui indique de compter les lignes. Cette commande est redirigée > vers un fichier appelé *nb_lignes.txt*.

Donc, *nb_lignes.txt* contiendra le nombre de lignes dans le fichier */chemin/vers/votre/fichier/source/apt*

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

cat /chemin/vers/votre/fichier/source/apt > save_sources

cat permet d'afficher le contenu du fichier spécifié par le chemin

/chemin/vers/votre/fichier/source/apt et **>** redirige ce contenu vers un fichier nommé **"save_sources"**.

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

grep -r "alias" .* | tee fichiers_alias.txt

grep une commande pour rechercher des motifs dans des fichiers texte.

- **r** est un option indique à **grep** de rechercher récursivement dans les sous-répertoires. Cela signifie que la recherche s'étendra à tous les fichiers et sous-répertoires à partir du répertoire actuel.

"alias" 'est le motif de recherche. Ici, on recherche toutes les occurrences du mot "alias" dans les fichiers texte.

.* est une partie de la commande pour spécifier les fichiers dans lesquels **grep** doit effectuer la recherche. **.*** est une expression régulière qui fait correspondre tous les fichiers et répertoires dans le répertoire actuel.

| est un pipe qui permet de rediriger la sortie standard de la commande précédente (dans ce cas, la sortie de **grep**) vers l'entrée standard d'une autre commande. Cela permet de chaîner des commandes ensemble.

tee est une commande qui prend l'entrée standard et la copie à la fois sur la sortie standard et dans un fichier spécifié.

fichiers_alias.txt est le nom du fichier dans lequel on souhaite enregistrer les résultats de la recherche.

Donc, lorsqu'on exécute cette commande **grep** va rechercher le mot "alias" dans tous les fichiers texte du répertoire actuel et de ses sous-répertoires. Les résultats de cette recherche sont affichés à l'écran. En même temps, grâce au **tee**, les résultats sont également écrits dans un fichier appelé "fichiers_alias.txt".