

Shell

Jules JEAN-LOUIS

JOB 1

Afficher le manuel de la commande `ls`

`man ls`

On peut aussi utiliser la commande `help`

Fichier cache du home

`ls -a`

“-a”, “all” tous, afficher les fichiers cachés plus les informations sur les droits sous forme de liste. les fichier marqué d’un (.) sont dans dans le répertoire actuel, alors que (..) indique le répertoire parent.

`ls -la`

-l affiche la liste des fichiers organise verticalement avec les permissions.

La ligne commence avec le nom du fichier ou l’authorisation du répertoire, le propriétaire ou le nom du groupe, la taille du fichier, la date de création ou modification.

Comment ajouter des options à une commande ?

Pour ajouter des options a un commande il suffit de rajouter un espace puis un moins (-) suivie de lettre correspondant au option, on peut en ajouter plusieurs option sur une même commande, par exemple :

`ls -lrtha`

commande en liste vertical, fichier caché, taille de fichier en kilo, mega, giga, ordre inverse, ranger par ordre, les nouveau en premiers.

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

Une commande linux doit suivre une certaine syntaxes :

`$ commande [options [argument1 [argument2 [...]]]]`

Entre la commande et l’option qui vont définir certaines actions, il faut un espace et un - devant l’option. Le nombre d’argument peut être limité, il existe plusieurs option de commande :

Option courte forme d'un tiré est d'une lettre **-s**
Option long avec deux tiré est un mots **-size**
Option courte et long qui exige des arguments **-speed 50** ou **-s 50**

JOB 2

La commande **cat [nom du fichier]** permet d'afficher le contenu du fichier directement dans le terminal sans le modifier.

nl [nom du fichier] affiche aussi le contenu du fichier avec la numérotation de chaque ligne.

vi. [nom du fichier] permet d'afficher dans l'éditeur

Pour afficher les 10 premières ligne : **head -n 10 [nom du fichier]**

-n [chiffre/nombre] définit le nombre de lignes qui sont affichés à partir du début du fichier.

Pour afficher les 10 dernières lignes : **tail -n 10 .bashrc**

-n [chiffre/nombre] définit le nombre de lignes qui sont affichées à partir de la fin du fichier.

Pour afficher les 20 premières ligne : **head -n 20 .bashrc**

Pour afficher les 20 dernières lignes : **tail -n 20 .bashrc**

JOB 3

Installation "**Cmatrix**"

D'abord utiliser la commande **sudo apt-get update** et **sudo apt-get upgrade**, afin d'être sûr que tous les package soit à jour. Cela va automatiquement rechercher tous les package installés est vérifié s' ils sont à jour. **apt-get update**, va rechercher les derniers package pour la distribution ainsi que les logiciels, mais ne va pas télécharger ni installer les mises à jour disponible. **apt-get upgrade** va téléchargé et installé les nouveaux package disponible.

Puis installé cmatrix avec :

```
$ sudo apt install cmatrix
```

Et le lancé avec la commande :

```
$ cmatrix
```

Telecharger google :

Utiliser la commande `wget` :

```
wget
```

```
https://dl.google.com/linux/direct/google-chrome-stable\_current\_amd64.deb
```

Puis procédez à l'installation :

```
sudo apt install ./google-chrome-stable_current_amd64.deb
```

Pour redémarrer l'ordinateur il faut disposer des privilèges super-utilisateur ou root.

On peut utiliser `sudo reboot` ou `sudo systemctl reboot`.

Lors du redémarrage, le système va afficher un message, le système va clôturer toutes les sessions en cours. On peut ne pas envoyer de message avec

```
sudo systemctl --no-wall reboot
```

On peut aussi personnaliser le message avec

```
sudo systemctl --message="Hardware upgrade" reboot
```

Pour éteindre la machine il faut utiliser la commande `shutdown`, avec l'option `-r` le système va redémarrer par défaut après 1 minute, le délai peut être modifié en ajustant la durée par exemple :

```
sudo shutdown -r 10:00
```

On redémarre immédiatement avec l'option `-r now`. L'option permet d'annuler le redémarrage planifié.

JOB 4

Créer un document `user.txt`

utiliser la commande `touch user.txt` est pour l'éditer utiliser `nano user.txt` pour y ajouter `User1` et `User2`.

Création du nouveau groupe "`Plateformes`", il faut les droits sudo ou root, la création du groupe va modifier le fichier `/etc/login.defs` et `/etc/group` contiendra le nom de chaque groupe avec les différents utilisateurs.

Utiliser la commande `groupadd` :

```
sudo groupadd Plateformeurs
```

Ensuite créé les 2 utilisateurs avec la commande **adduser** :

```
adduser User1
```

```
adduser User2
```

cela ne fonctionne pas car linux n'autorise pas les majuscules dans les nom d'utilisateur, il faut donc utiliser :

```
sudo adduser --force-badname User1
```

refaire la manipulation pour l'utilisateur 2, que l'on va ajouté au groupe Plateformeurs précédents créé avec :

```
sudo usermod -a -G Plateformeurs User2
```

-a qui s'utilise avec -G pour ajouter un utilisateur à un groupe supplémentaire.

Création d'un fichier droits.txt en modifiant le fichier user.txt

```
mv user.txt droits.txt
```

et copier le fichier droits.txt en renommant groupes.txt

```
cp droits.txt groupes.txt
```

On peut aussi utiliser la commande **>** :

```
cat droits.txt > groupes.txt
```

On doit maintenant changer le propriétaire du fichier droits.txt pour User1 grace a la commande chown:

```
chown User1 droits.txt
```

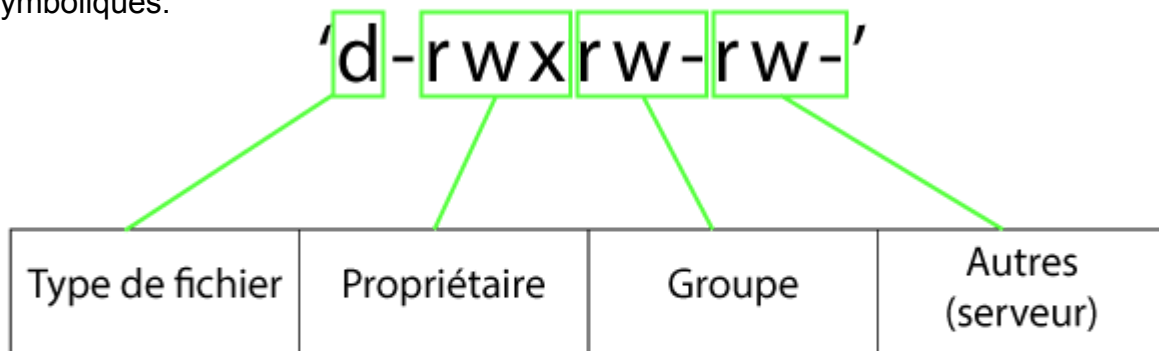
 ou avec la reference ID de l'User1

La syntaxe de **chown nom_utilisateur chemin_vers_fichier**, on peut aussi passer en root est utilisé la commande chmod u+rw pour donner les droits à un utilisateur.

Le fichier **droits.txt** appartenant maintenant au User1 il est en lecture seulement pour l'User2.

Utiliser la commande **ls -l** pour afficher les paramètres des fichiers contenu dans un dossier. Les permissions sont indiquées dans la colonne de gauche, suivant un format bien particulier :

Le premier caractère du bloc de permissions indique le type du fichier : - pour un fichier normal, d pour un répertoire. On trouve également parfois l pour les liens symboliques.



Droits du propriétaire

- **r** ou - : droit de lire (r pour *read*) le fichier (r pour oui, - pour non)
- **w** ou - : droit d'écrire (w pour *write*) dans le fichier

- **x** ou **-** : droit d'exécuter (x pour *execute*) le fichier.

Droits du groupe

Comme les droits du propriétaire, mais s'applique aux gens qui sont dans le groupe propriétaire.

Droits des autres

Comme les droits du propriétaire, mais s'applique aux gens qui sont ni le propriétaire, ni dans le groupe propriétaire.

Modifier les droits d'accès

Il existe 4 commandes liées aux droits d'accès :

- **chmod** (*change mode*), la plus utile, qui modifie les permissions
- **chgrp** (*change group*) qui modifie le groupe d'un fichier ou répertoire
- **chown** (*change owner*) qui modifie le propriétaire d'un fichier ou répertoire
- **umask**, qui définit les protections par défaut

Pour que le fichier soit en lecture seulement il faut utiliser la commande **chmod**, qui va déterminer les permissions avec une suite de chiffres.

La lecture (**r**)(read) - 4

L'écriture (**w**)(write) - 2

L'exécution (**x**)(eXecute) - 1

Toutes les permission (**rwX**) - 7

Aucune permission - 0

Les permissions s'ajoutent si on veut toutes les permissions on ajoute (**r**) - 4 (**w**) - 2 (**x**) - 1. **rwX** = 4+2+1=7.

Les permissions du groupe et des autres sont autorisées qu'à lire donc 4.

chmod 0444 groupes.txt

re-donné la permission en lecture/écriture

chmod 666 groupes.txt pour avoir **-rw-rw-rw**

On peut aussi utiliser **chmod u+rw**

JOB 5

Création d'alias

Les alias sont des raccourcis que l'on peut créer dans le terminal temporaire, on peut les ajouter dans le fichier `.bashrc` ou `.bash_aliases` pour qu'il soit permanent:

```
alias la='ls -la'
alias update='sudo apt-get update'
alias upgrade='sudo apt-get upgrade'
```

Les variables d'environnement sont des raccourcis pour des applications qui peuvent contenir des texte, nombres, ou des autres données. La variable User existe mais on peut la créer en utilisant :

```
$export UTILISATEUR=Jules
$echo $UTILISATEUR
$Jules
```

Pour mettre à jour de manière permanente les alias il faut modifier le fichier `.bashrc` ou `.bash_aliases`.

```
$ export $PATH=/home/jules/desktop
```

Télécharger et désarchivé :

Utiliser la commande `wget` :

```
wget "https://drive.google.com/file/d/1wrjp6bto9ni5yyC3ej3zxDtqELazHPHz/view?usp=sharing&export=download" -O"CopiedeGhostintheShell.tar"
```

La commande `-O` permet de directement renommer le fichier télécharger.

Utiliser la commande `tar` pour désarchiver les fichier `.tar` :

```
tar -xf CopiedeGhostintheshell.tar
```

Les options sont les suivante :

```
-x pour extraire, -v pour verbose et -f pour fichier
```

Liste de plusieurs commandes réaliser en une ligne :

```
touch une_commande.txt && echo "Je suis votre fichier texte" >
une_commande.txt
cat /etc/apt/sources.list | wc -l > nb_lignes.txt
cat /etc/apt/sources.list > save_sources
grep alias .*
```

Pour aller plus loin ...

Installation de tree :

```
$ sudo apt install tree
```

```
$ tree
```

Les chevrons permettent de créer un fichier vide ou bien pour vider un fichier existant de son contenu dans un répertoire. La commande > et touch sont des équivalents, sauf que touch permet de mettre à jour la date et l'heure.

La redirection est une fonction des chevrons, on peut mettre le résultat d'une première commande dans un fichier créer, les messages d'erreur ou un périphérique. Défini par les chiffres 0, 1 et 2.

L'utilisation du simple chevron (>) va supprimer le contenu d'un fichier déjà présent alors que le double chevron (>>) ajoutera le contenu à la suite.

```
tree > tree.save
```

ls -la | wc -l dans un répertoire, tree -a | wc -l pour le nombre d'éléments totaux

```
sudo apt update && sudo apt upgrade
```

Le double ampersand permet d'indiquer au terminal bash d'exécuter la première commande, si elle réussit, l'opérateur exécute la deuxième.

Enchaînements de commandes:

Exécution simultanée : `com1 & com2`

Les commandes s'exécutent en simultané.

Exécution successive : `com1 ; com2`

Les commandes s'exécutent l'une après les autres.

Enchaînements conditionnels :

`com1 && com2` La commande deux ne s'exécute que si la première s'est soldée par un succès.

Alternative : `com1 || com2 || ... || comN`

La commande com1 jusqu'à comN sont exécutées successivement tant qu'aucune ne se termine correctement. Dès qu'une des commandes se soldée par un succès, la commande alternative est terminée.