

Lexique des commandes

addgroup	3
adduser	4
alias	5
apt.....	6
cat.....	8
cd	9
chmod	10
chown	11
cp	12
crontab	13
delgroup.....	14
deluser.....	15
df.....	16
diff.....	17
du.....	18
echo	19
find.....	20
free	21
grep.....	22
iostat.....	23
ip address.....	24
kill.....	25
locate.....	26
ls.....	27
man	28
mkdir.....	29
mpstat.....	30
mv.....	31

nano.....	32
netplan.....	33
pkill.....	34
ps	35
pwd.....	36
read.....	37
rm	38
rmdir	39
sed	40
ssh-copy-id	41
ssh-keygen.....	43
sudo	45
top	47
touch.....	48
uptime.....	49
usermod.....	50
vim	51
watch.....	53

addgroup

Description

La commande "addgroup" permet de créer un nouveau groupe sur votre système Linux. Cette commande est généralement utilisée pour créer un groupe pour gérer les permissions et les accès aux fichiers pour différents utilisateurs.

Options

-g GID : Spécifie le numéro d'identifiant de groupe (GID) pour le nouveau groupe.
--system : Crée un groupe système, qui est généralement utilisé pour les services et les applications du système d'exploitation.

Exemple

Créer un nouveau groupe nommé "mygroup" :

```
sudo addgroup mygroup
```

Créer un groupe système nommé "mygroup" avec un GID spécifique (par exemple, 5000) :

```
sudo addgroup --system -g 5000 mygroup
```

Vidéo

Section 4 - Administration de base de son système Linux

- Quelques manipulations pour créer et supprimer des groupes
- Exercice en ligne - Apprendre à manipuler les groupes
- [Correction] Exercice en ligne - Apprendre à manipuler les groupes

adduser

Description

La commande "adduser" permet de créer un nouvel utilisateur sur votre système Linux. Elle crée un répertoire personnel pour l'utilisateur dans /home, copie les fichiers de configuration par défaut et attribue un groupe principal pour l'utilisateur.

Options

--system : Crée un utilisateur système sans répertoire personnel.
--ingroup GROUP : Ajoute le nouvel utilisateur au groupe spécifié (GROUP).
--shell SHELL : Définit le shell par défaut pour le nouvel utilisateur (par exemple, /bin/bash).
--disabled-password : Crée un compte sans mot de passe. L'utilisateur ne pourra pas se connecter avec un mot de passe, mais pourra se connecter avec d'autres méthodes, comme les clés SSH.
--gecos GECOS : Configure les informations complémentaires pour l'utilisateur, telles que le nom complet, le numéro de téléphone, etc.

Exemple

Créer un nouvel utilisateur nommé "john" :
`sudo adduser john`

Créer un nouvel utilisateur nommé "john" et l'ajouter au groupe "sudo" :
`sudo adduser john --ingroup sudo`

Vidéo

Section 4 - Administration de base de son système Linux

- Quelques manipulations pour créer et supprimer des utilisateurs
- Exercice en ligne - Apprendre à manipuler les utilisateurs
- [Correction] Exercice en ligne - Apprendre à manipuler les utilisateurs

alias

Description

La commande "alias" permet de créer des raccourcis pour des commandes ou des séquences de commandes.

Options

Exemple

Créer un alias "ll" pour la commande "ls -l" :

```
$ alias ll="ls -l"
```

```
$ ll
```

```
total 12
```

```
drwxr-xr-x 2 jordan jordan 4096 Apr 28 12:30 Documents
```

```
drwxr-xr-x 2 jordan jordan 4096 Apr 28 12:30 Pictures
```

```
-rw-r--r-- 1 jordan jordan  45 Apr 28 12:30 test.txt
```

Pour en savoir plus sur la commande "alias", consultez le manuel en utilisant la commande "man alias" ou recherchez des informations sur la gestion des alias dans le shell que vous utilisez (par exemple, Bash, Zsh, etc.).

Vidéo

- Section 2 – Premières utilisations d'Ubuntu et commandes de base
Utilisation des Alias pour administrer son système

apt

Description

La commande apt (Advanced Package Tool) est un outil de gestion de paquets pour les systèmes d'exploitation basés sur Debian, comme Ubuntu. Elle permet d'installer, mettre à jour, supprimer et gérer les paquets logiciels.

Options

update : Met à jour la liste des paquets disponibles à partir des dépôts configurés.

upgrade : Met à jour les paquets installés avec les dernières versions disponibles.

install : Installe un ou plusieurs paquets logiciels.

remove : Supprime un ou plusieurs paquets logiciels.

autoremove : Supprime les paquets inutilisés (installés automatiquement et devenus inutiles).

search : Recherche des paquets correspondant au terme spécifié.

show : Affiche des informations détaillées sur un paquet spécifique.

Exemple

```
sudo apt install vim
```

```
Reading package lists... Done
```

```
Building dependency tree
```

```
Reading state information... Done
```

```
The following additional packages will be installed:
```

```
vim-runtime
```

```
Suggested packages:
```

```
ctags vim-doc vim-scripts
```

```
The following NEW packages will be installed:
```

```
vim vim-runtime
```

```
0 upgraded, 2 newly installed, 0 to remove and 0 not  
upgraded.
```

```
Need to get 6,724 kB of archives.
```

```
After this operation, 32.2 MB of additional disk space will  
be used.
```

```
Do you want to continue? [Y/n] Y
```

Dans cet exemple, la commande `sudo apt install vim` installe le paquet "vim" et ses dépendances.

Vidéo

Section 4 - Administration de base de son système Linux

- Installer et supprimer des paquets sur Ubuntu
- Exercice en ligne - Apprendre à manipuler des paquets
- [Correction] Exercice en ligne - Apprendre à manipuler des paquets

cat

Description

Permet de concaténer plusieurs fichiers et d'afficher le contenu de fichiers existants.

Options

Exemple

```
jordan@ubuntu:~$ ls  
monfichier1 monfichier2 monfichier3 monfichier4
```

```
jordan@ubuntu:~$ cat monfichier1  
Voici le contenu du fichier monfichier1  
Ici nous sommes à la deuxième ligne du fichier
```

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Exploration de mes premières commandes Linux
- Exercice en ligne – commandes Linux de base

cd

Description

Permet de changer de répertoire courant au niveau du terminal.

Options

Exemple

```
jordan@ubuntu:~$ pwd  
/home/jordan
```

```
jordan@ubuntu:~/Documents$ cd Documents  
jordan@ubuntu:~/Documents$ pwd  
/home/jordan/Documents
```

Dans l'exemple ci-dessus, on se trouve initialement dans le répertoire /home/jordan. Puis grâce à la commande cd, nous nous déplaçons dans le répertoire /home/jordan/Documents.

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Exploration de mes premières commandes Linux
- Exercice en ligne – commandes Linux de base

chmod

Description

La commande "chmod" permet de modifier les permissions d'un fichier ou d'un répertoire sous Linux. Les permissions sont divisées en trois catégories : propriétaire (u), groupe (g) et autres (o).

Options

-R : Applique les modifications récursivement aux fichiers et répertoires à l'intérieur du répertoire spécifié.

--reference=RFILE : Utilise les permissions du fichier RFILE comme modèle pour définir les permissions du fichier ou du répertoire cible.

Exemple

Ajouter la permission d'exécution pour le propriétaire d'un fichier nommé "script.sh" :

```
chmod u+x script.sh
```

Attribuer les permissions de lecture, écriture et exécution au propriétaire et lecture et exécution au groupe et aux autres pour un fichier nommé "script.sh" :

```
chmod 755 script.sh
```

Cela équivaut à :

- Propriétaire : 4 (lecture) + 2 (écriture) + 1 (exécution) = 7
- Groupe : 4 (lecture) + 1 (exécution) = 5
- Autres : 4 (lecture) + 1 (exécution) = 5

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Droits d'accès aux fichiers et répertoires sur Linux
- Exercice en ligne - Manipuler les permissions des fichiers
- [Correction] Exercice en ligne - Manipuler les permissions des fichiers

chown

Description

La commande "chown" permet de modifier le propriétaire et le groupe d'un fichier ou d'un répertoire sous Linux. Elle est généralement utilisée pour changer les permissions d'accès aux fichiers et répertoires.

Options

-R : Change le propriétaire et le groupe de manière récursive pour les répertoires et leurs contenus.
-c : Affiche uniquement les modifications effectuées.
-v : Affiche un message détaillé pour chaque fichier ou répertoire traité.
--preserve-root : Empêche la modification du propriétaire et du groupe du répertoire racine lors de l'utilisation de l'option -R.

Exemple

Changer le propriétaire du fichier "example.txt" en "jordan" et le groupe en "staff":

```
sudo chown jordan:staff example.txt
```

Exemple d'utilisation avec l'option -R pour changer le propriétaire et le groupe de manière récursive sur un répertoire et son contenu :

```
sudo chown -R jordan:staff /path/to/directory
```

Vidéo

Section 4 - Administration de base de son système Linux

- Quelques manipulations pour créer et supprimer des groupes
- Exercice en ligne - Apprendre à manipuler les groupes
- [Correction] Exercice en ligne - Apprendre à manipuler les groupes

cp

Description

La commande "cp" est utilisée pour copier des fichiers et des répertoires d'un emplacement à un autre. Elle crée une copie indépendante de l'élément source.

Options

- r : Copie récursive des répertoires et de leur contenu.
- p : Préserve les attributs du fichier (permissions, date de modification, etc.).
- i : Demande confirmation avant d'écraser un fichier existant.
- u : Copie uniquement si la source est plus récente que la destination ou si la destination n'existe pas.

Exemple

Copier un fichier "file.txt" dans un répertoire "backup" :

```
$ cp file.txt backup/
```

Copier récursivement un répertoire "folder" et son contenu dans un autre répertoire "backup" :

```
$ cp -r folder backup/
```

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice 2 en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice 2 en ligne - Manipulation des fichiers et des dossiers

crontab

Description

La commande crontab permet de créer, modifier, lister et supprimer des tâches planifiées (cron jobs) pour un utilisateur. Les tâches planifiées sont des commandes ou des scripts qui sont exécutés automatiquement à des intervalles réguliers ou à des heures spécifiques.

Options

- e : Ouvre le fichier crontab de l'utilisateur pour l'édition avec l'éditeur de texte par défaut.
- l : Liste les tâches planifiées actuelles pour l'utilisateur.
- r : Supprime le fichier crontab de l'utilisateur, supprimant ainsi toutes les tâches planifiées.
- u : Spécifie l'utilisateur pour lequel afficher, éditer ou supprimer le fichier crontab (nécessite les privilèges sudo).D31

Exemple

Pour ajouter une tâche planifiée qui exécute un script toutes les heures :

- Tapez crontab -e pour ouvrir le fichier crontab de l'utilisateur en édition.

- Ajoutez la ligne suivante :

```
0 * * * * /chemin/vers/mon-script.sh
```

- Sauvegardez et fermez le fichier.

Exemple de sortie de la commande crontab -l :

```
# m h dom mon dow    command
0 * * * * /chemin/vers/mon-script.sh
```

Dans cet exemple, la tâche planifiée exécute le script /chemin/vers/mon-script.sh toutes les heures à la minute 0 (par exemple, 12:00, 13:00, 14:00, etc.).

Vidéo

Section 4 - Administration de base de son système Linux

- Mise en place de crontabs sur votre système
- Exercice en ligne - Apprendre à utiliser les crontabs
- [Correction] Exercice en ligne - Apprendre à utiliser les crontabs

delgroup

Description

La commande "delgroup" permet de supprimer un groupe de votre système Linux. Cette commande retire également le groupe de la liste des groupes d'appartenance de tous les utilisateurs qui en font partie.

Options

--only-if-empty : Supprime le groupe seulement s'il n'a pas de membres.

Exemple

Supprimer le groupe "mygroup" :

```
sudo delgroup mygroup
```

Supprimer le groupe "mygroup" uniquement s'il est vide (sans membres) :

```
sudo delgroup --only-if-empty mygroup
```

Vidéo

Section 4 - Administration de base de son système Linux

- Quelques manipulations pour créer et supprimer des groupes
- Exercice en ligne - Apprendre à manipuler les groupes
- [Correction] Exercice en ligne - Apprendre à manipuler les groupes

deluser

Description

La commande "deluser" permet de supprimer un utilisateur de votre système Linux. Elle supprime l'utilisateur du système, mais ne supprime généralement pas les fichiers personnels de l'utilisateur, qui se trouvent dans le répertoire /home de l'utilisateur.

Options

--remove-home : Supprime également le répertoire personnel de l'utilisateur.
--remove-all-files : Supprime tous les fichiers du système qui appartiennent à l'utilisateur.
--backup-to DIR : Effectue une sauvegarde des fichiers de l'utilisateur dans le répertoire spécifié (DIR) avant de les supprimer.
--group GROUP : Supprime l'utilisateur du groupe spécifié (GROUP) sans supprimer l'utilisateur lui-même.

Exemple

Supprimer un utilisateur nommé "john" :

```
sudo deluser john
```

Supprimer un utilisateur nommé "john" et son répertoire personnel :

```
sudo deluser --remove-home john
```

Vidéo

Section 4 - Administration de base de son système Linux

- Quelques manipulations pour créer et supprimer des utilisateurs
- Exercice en ligne - Apprendre à manipuler les utilisateurs
- [Correction] Exercice en ligne - Apprendre à manipuler les utilisateurs

df

Description

Permet d'afficher l'espace disque utilisé et disponible sur les systèmes de fichiers montés.

Options

- h : Affiche les tailles en format lisible par l'homme (K, M, G, etc.).
- T : Affiche le type de système de fichiers.
- a : Inclut les systèmes de fichiers ayant 0 blocs.
- total : Affiche un total général à la fin.
- i : Affiche les inodes au lieu des blocs.

Exemple

```
jordan@ubuntu:~$ df -h
```

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
udev	1,9G	0	1,9G	0%	/dev
tmpfs	388M	1,6M	386M	1%	/run
/dev/sda1	19G	11G	7,4G	59%	/
tmpfs	1,9G	17M	1,9G	1%	/dev/shm
tmpfs	5,0M	4,0K	5,0M	1%	/run/lock
tmpfs	1,9G	0	1,9G	0%	/sys/fs/cgroup
/dev/sda6	5,5G	1,5G	3,8G	29%	/home

Cet exemple permet d'afficher l'espace disque utilisé et disponible sur les systèmes de fichiers montés en format lisible par l'homme.

Vidéo

Section 6 - Administration avancée d'un système Linux

- Gestion de l'espace disque avec les commandes "df", "du", et "iostat"

diff

Description

La commande "diff" permet de comparer le contenu de deux fichiers et d'afficher les différences ligne par ligne. Elle est particulièrement utile pour identifier les modifications apportées à des fichiers texte.

Options

- u : Utilise le format unifié pour afficher les différences.
- r : Compare récursivement les répertoires.
- i : Ignore la casse.
- b : Ignore les changements dans la quantité d'espaces.
- N : Traite les fichiers inexistants comme vides.

Exemple

Comparaison de deux fichiers texte "file1.txt" et "file2.txt" :

```
$ diff file1.txt file2.txt

1c1
< Ceci est la première ligne du fichier 1.
---
> Ceci est la première ligne du fichier 2.
3c3
< Ceci est la troisième ligne du fichier 1.
---
> Ceci est la troisième ligne du fichier 2.
```

La sortie indique que les lignes 1 et 3 des deux fichiers sont différentes. Les signes "<" et ">" indiquent les lignes spécifiques de chaque fichier. Pour en savoir plus sur les options de la commande "diff", consultez le manuel en utilisant la commande "man diff".

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice 2 en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice 2 en ligne - Manipulation des fichiers et des dossiers

du

Description

Permet d'estimer l'espace disque utilisé par les fichiers et les répertoires.

Options

- h : Affiche les tailles en format lisible par l'homme (K, M, G, etc.).
- s : Affiche uniquement un total pour chaque argument (résumé).
- max-depth=N : Affiche l'utilisation du disque pour les N premiers niveaux de profondeur (N est un nombre entier).
- a : Inclut les fichiers, pas seulement les répertoires.
- c : Affiche un total général à la fin.

Exemple

```
jordan@ubuntu:~$ du -h Documents
```

```
24K      Documents/project1
12K      Documents/project2
4.0K     Documents/notes
40K      Documents
```

Cet exemple permet d'afficher l'espace disque utilisé par le répertoire "Documents" et ses sous-répertoires en format lisible par l'homme.

Vidéo

Section 6 - Administration avancée d'un système Linux

- Gestion de l'espace disque avec les commandes "df", "du", et "iostat"

echo

Description

La commande echo permet d'afficher une chaîne de caractères ou le contenu d'une variable à l'écran (sur la sortie standard).

Options

-n : Ne pas ajouter de saut de ligne à la fin de l'affichage.

-e : Permet l'interprétation des caractères d'échappement, tels que \n pour un saut de ligne, ou \t pour une tabulation.

Exemple

```
echo "Hello, World!"  
echo -e "Première ligne\nSeconde ligne"  
NOM="Jordan"  
echo "Bonjour, $NOM !"
```

```
Hello, World!  
Première ligne  
Seconde ligne  
Bonjour, Jordan !
```

Utilisable dans un script Shell pour afficher du texte.

Vidéo

Section 5 - Le scripting Bash

- Afficher du texte à l'écran grâce à un Script et au "Shebang"

find

Description

La commande "find" permet de rechercher des fichiers et des répertoires sur un système Linux en fonction de différents critères, tels que le nom, la taille ou la date de modification. Elle effectue une recherche récursive dans les répertoires spécifiés.

Options

- name : Recherche par nom de fichier.
- type : Recherche par type de fichier (f pour fichier, d pour répertoire).
- mtime : Recherche par date de modification (en jours).
- size : Recherche par taille de fichier.
- exec : Exécute une commande sur chaque fichier trouvé.

Exemple

Rechercher tous les fichiers avec l'extension ".txt" dans le répertoire "/home/user" et ses sous-répertoires :

```
$ find /home/user -type f -name "*.txt"
```

```
/home/user/document1.txt  
/home/user/folder/document2.txt  
/home/user/folder/subfolder/document3.txt
```

La commande "find" est très puissante et offre de nombreuses autres options pour affiner les recherches. Pour en savoir plus, consultez le manuel en utilisant la commande "man find".

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Trouver des fichiers et des dossiers sur Linux
- Exercice en ligne - Apprendre à utiliser la commande find
- [Correction] Exercice en ligne - Apprendre à utiliser la commande find

free

Description

Permet d'afficher des informations sur l'utilisation de la mémoire et du swap sur le système.

Options

- b : Affiche la sortie en octets.
- k : Affiche la sortie en kilo-octets (par défaut).
- m : Affiche la sortie en méga-octets.
- g : Affiche la sortie en giga-octets.
- tera : Affiche la sortie en téra-octets.
- h : Affiche la sortie dans un format lisible par l'homme.
- t : Affiche une ligne supplémentaire avec les totaux.

Exemple

```
jordan@ubuntu:~$ free -m
```

	total	used	free	shared
buff/cache	available			
Mem:	3830	1543	1002	124
1285	2005			
Swap:	4095	0	4095	

Cet exemple permet d'afficher les informations sur l'utilisation de la mémoire et du swap en méga-octets.

Vidéo

Section 6 - Administration avancée d'un système Linux

- Gestion de la mémoire RAM et commande "free"

grep

Description

Permet de rechercher des chaînes de caractères dans des fichiers en utilisant des expressions régulières.

Options

- i : Recherche insensible à la casse.
- r : Recherche récursive dans les sous-répertoires.
- n : Affiche les numéros de ligne correspondants.

Exemple

```
jordan@ubuntu:~$ cat test.txt
Je suis une première ligne
Voici la seconde ligne
Ici se situe le mot "jordan" que je recherche

jordan@ubuntu:~$ grep -i "jordan" test.txt
Ici se situe le mot "jordan" que je recherche
```

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Découverte du Man sur Linux
- Exercice en ligne – Utiliser le man et découvrir de nouvelles commandes

iostat

Description

Permet d'afficher des statistiques sur l'utilisation des dispositifs d'entrée/sortie (E/S) du système, comme les disques durs, les SSD et les dispositifs de stockage similaires.

Options

- c : Affiche uniquement les statistiques du processeur.
- d : Affiche uniquement les statistiques des dispositifs.
- k : Affiche les statistiques en kilooctets par seconde.
- m : Affiche les statistiques en mégaoctets par seconde.
- p [DEVICE] : Affiche les statistiques pour un dispositif spécifique ou pour tous les dispositifs.
- t : Affiche le temps écoulé depuis le dernier redémarrage du système.
- x : Affiche les statistiques étendues pour les dispositifs.
- y : Omet les premiers rapports avec une période de temps égale à zéro.

Exemple

```
jordan@ubuntu:~$ iostat
```

```
Linux 4.15.0-29-generic (ubuntu)    03/11/2019    _x86_64_
(4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           6.85    0.01    2.43    0.38    0.00   90.33

Device:            tps    kB_read/s    kB_wrtn/s    kB_read
kB_wrtn
sda                 1.99         23.97         20.42     2447012
2084620
```

Cet exemple permet d'afficher les statistiques d'utilisation des dispositifs d'entrée/sortie et du processeur.

Vidéo

Section 6 - Administration avancée d'un système Linux

- Visualiser la charge des processeurs avec : uptime, iostat, et mpstat
- Gestion de l'espace disque avec les commandes "df", "du", et "iostat"

ip address

Description

La commande "ip address" est utilisée pour afficher et gérer les adresses IP des interfaces réseau sur un système Linux. Elle permet de configurer et d'afficher les adresses IPv4 et IPv6, ainsi que de vérifier les interfaces réseau actives.

Options

show : Affiche les adresses IP et d'autres informations sur toutes les interfaces réseau.

add : Ajoute une nouvelle adresse IP à une interface réseau.

del : Supprime une adresse IP d'une interface réseau.

flush : Supprime toutes les adresses IP d'une interface réseau.

Exemple

```
ip address show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
mq state UP group default qlen 1000
    link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.2/16 brd 172.17.255.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::42:acff:fe11:2/64 scope link
        valid_lft forever preferred_lft forever
```

Affiche les adresses IP de toutes les interfaces réseau.

Vidéo

Section 4 - Administration de base de son système Linux

- Modifier les paramètres réseaux sur Ubuntu 20.04

kill

Description

Permet de terminer un processus en envoyant un signal spécifique à son PID (Identifiant de processus).

Options

signal : Envoyer un signal spécifique au processus (par exemple, -9 pour envoyer un signal SIGKILL). Les signaux les plus courants sont -9 (SIGKILL), -15 (SIGTERM) et -2 (SIGINT).

Exemple

```
jordan@ubuntu:~$ kill -9 12345
```

Permet d'envoyer un signal SIGKILL (forcer l'arrêt) au processus ayant le PID 12345

Vidéo

Section 6 - Administration avancée d'un système Linux

- Arrêter les processus grâce aux commandes "kill" et "pkill"

locate

Description

La commande "locate" permet de rechercher rapidement des fichiers et des répertoires sur un système Linux. Elle utilise une base de données interne pour effectuer la recherche, qui est généralement mise à jour régulièrement avec la commande "updatedb".

Options

- i : Recherche insensible à la casse.
- l LIMIT : Limite le nombre de résultats retournés.
- S : Affiche des statistiques sur la base de données utilisée pour la recherche.

Exemple

Rechercher tous les fichiers et répertoires contenant le mot "example" :

```
$ locate example
```

```
/usr/share/doc/python3.8/examples/scripts/example.py  
/usr/share/doc/python3.9/examples/scripts/example.py  
/home/user/example.txt
```

Notez que la base de données de "locate" peut ne pas être à jour avec les fichiers récemment ajoutés ou supprimés. Pour mettre à jour la base de données, vous pouvez utiliser la commande "sudo updatedb".

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Trouver des fichiers et des dossiers sur Linux

ls

Description

La commande "ls" (list) est utilisée pour lister le contenu d'un répertoire.

Options

- a : Affiche les fichiers et dossiers cachés (dont le nom commence par un point).
- l : Affiche les détails des fichiers et dossiers, tels que les permissions, la taille, la date de modification et le propriétaire.
- R : Liste les fichiers et dossiers récursivement.
- t : Trie les fichiers et dossiers par date de modification.
- r : Inverse l'ordre de tri.

Exemple

Lister le contenu d'un répertoire en affichant les détails des fichiers et dossiers :

```
$ ls -l
total 12
drwxr-xr-x 2 jordan jordan 4096 Apr 28 12:30 Documents
drwxr-xr-x 2 jordan jordan 4096 Apr 28 12:30 Pictures
-rw-r--r-- 1 jordan jordan   45 Apr 28 12:30 test.txt
```

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Détails sur la commande "ls" et les notions de chemins relatifs et absolus
- Exercice en ligne - Utiliser la commande ls

man

Description

Affiche le manuel de l'utilisateur pour une commande donnée.

Options

Exemple

```
jordan@ubuntu:~$ man cat
```

NAME

cat - concatenate files and print on the standard output

SYNOPSIS

```
cat [OPTION]... [FILE]...
```

DESCRIPTION

Concatenate FILE(s) to standard output.

With no FILE, or when FILE is -, read standard input.

-A, --show-all

equivalent to -vET

-b, --number-nonblank

number nonempty output lines, overrides -n

-e equivalent to -vE

-E, --show-ends

display \$ at end of each line

-n, --number

number all output lines

-s, --squeeze-blank

suppress repeated empty output lines

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Découverte du Man sur Linux
- Exercice en ligne – Utiliser le man et découvrir de nouvelles commandes

mkdir

Description

La commande "mkdir" est utilisée pour créer un nouveau répertoire. Elle permet de spécifier le nom et l'emplacement du répertoire à créer.

Options

-p : Crée les répertoires parents si nécessaire. Si les répertoires parents existent déjà, aucune erreur ne sera générée.
-m : Permet de définir les permissions du répertoire créé, en utilisant la notation octale.

Exemple

Créer un répertoire nommé "new_directory" :

```
$ mkdir new_directory
```

Créer une structure de répertoires imbriqués "dir1/dir2/dir3" en une seule commande :

```
$ mkdir -p dir1/dir2/dir3
```

Créer un répertoire avec des permissions spécifiques (ici, en lecture et écriture pour le propriétaire, et en lecture seulement pour le groupe et les autres) :

```
$ mkdir -m 744 new_directory
```

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice en ligne - Manipulation des fichiers et des dossiers

mpstat

Description

Permet d'afficher des statistiques sur l'utilisation du processeur par les différents cœurs ou l'ensemble du système.

Options

- P ALL : Affiche les statistiques pour tous les cœurs individuellement.
- P ON : Affiche les statistiques pour un cœur spécifique (remplacez ON par le numéro du cœur souhaité).
- u : Affiche l'utilisation globale du processeur (par défaut).
- I {SUM|CPU|SCPU|ALL} : Affiche les statistiques pour l'interrupteur, soit globalement, par processeur ou pour chaque processeur et chaque interrupteur matériel.
- dec=DEC : Fixe le nombre de décimales affichées à DEC (valeur par défaut : 2).

Exemple

```
jordan@ubuntu:~$ mpstat -P ALL
```

```
Linux 4.15.0-29-generic (ubuntu)    03/11/2019    _x86_64_  
(4 CPU)
```

03:20:02 PM	CPU	%usr	%nice	%sys	%iowait	%irq
%soft	%steal	%guest	%gnice	%idle		
03:20:02 PM	all	6.85	0.01	2.43	0.38	0.00
0.07	0.00	0.00	0.00	90.27		
03:20:02 PM	0	6.82	0.01	2.51	0.37	0.00
0.09	0.00	0.00	0.00	90.21		
03:20:02 PM	1	6.83	0.01	2.39	0.37	0.00
0.07	0.00	0.00	0.00	90.33		
03:20:02 PM	2	6.87	0.01	2.40	0.38	0.00
0.06	0.00	0.00	0.00	90.29		
03:20:02 PM	3	6.83	0.01	2.42	0.39	0.00
0.05	0.00	0.00	0.00	90.29		

Cet exemple permet d'afficher les statistiques sur l'utilisation du processeur pour tous les cœurs.

Vidéo

Section 6 - Administration avancée d'un système Linux

- Visualiser la charge des processeurs avec : uptime, iostat, et mpstat

mv

Description

La commande "mv" est utilisée pour déplacer ou renommer des fichiers et des répertoires. Elle permet de déplacer des éléments d'un emplacement à un autre ou de simplement changer le nom d'un fichier ou d'un répertoire sans changer sa position.

Options

- f : Force l'écrasement des fichiers de destination existants.
- i : Invite l'utilisateur avant d'écraser les fichiers de destination existants.
- u : Ne déplace que les fichiers source plus récents que les fichiers de destination ou lorsque le fichier de destination n'existe pas.

Exemple

Déplacer un fichier "file1.txt" dans un autre répertoire appelé "directory":

```
$ mv file1.txt directory/
```

Renommer un fichier "file1.txt" en "file2.txt":

```
$ mv file1.txt file2.txt
```

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice 2 en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice 2 en ligne - Manipulation des fichiers et des dossiers

nano

Description

Permet d'éditer des fichiers texte dans un terminal à l'aide d'un éditeur de texte simple et convivial.

Options

- B : Crée automatiquement une sauvegarde du fichier avant d'enregistrer les modifications.
- I : Active le mode "ignorer la casse" pour la recherche.
- M : Active le mode "mouse" qui permet l'utilisation de la souris.
- N : Active le mode "no convert" qui empêche la conversion automatique des fichiers entre les formats de fin de ligne DOS et Unix.
- Y : Spécifie la coloration syntaxique à utiliser pour le fichier.

Raccourcis de l'éditeur "nano"

- Ctrl + G : affiche l'aide de nano
- Ctrl + K : coupe la ligne sur laquelle se situe le curseur et la place de le presse-papier
- Ctrl + U : colle le contenu du presse-papier
- Ctrl + C : affiche la position du curseur (numéro de ligne et numéro de colonne)
- Ctrl + W : permet d'effectuer une recherche de texte dans le fichier
- Ctrl + O : permet d'enregistrer les modifications effectuées sur le fichier
- Ctrl + X : permet de quitter l'éditeur de texte nano

Exemple

Ouvrir et éditer un fichier nommé "example.txt" avec nano:

```
jordan@ubuntu:~$ nano example.txt
```

Modifier le fichier comme souhaité, puis enregistrer et quitter en utilisant Ctrl + X, puis Y pour confirmer les modifications, et enfin Enter.

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Prendre en main l'éditeur de texte nano

netplan

Description

La commande netplan est un outil de gestion de la configuration réseau pour les systèmes d'exploitation basés sur Ubuntu. Elle permet de configurer facilement les interfaces réseau, les adresses IP, les passerelles, les serveurs DNS et d'autres paramètres de réseau.

Options

apply : Applique les changements de configuration réseau immédiatement.

generate : Génère des fichiers de configuration réseau à partir des fichiers de configuration YAML de netplan.

debug : Vérifie les fichiers de configuration pour déceler les erreurs éventuelles et affiche des informations de débogage.

info : Affiche des informations sur la version de netplan installée.

Exemple

```
sudo netplan apply
```

Aucune sortie n'est généralement attendue lors de l'exécution de la commande `sudo netplan apply`. La commande applique simplement les changements de configuration réseau et retourne à l'invite de commande.

Vidéo

Section 4 - Administration de base de son système Linux

- Modifier les paramètres réseaux sur Ubuntu 20.04

pgill

Description

Permet de terminer des processus basés sur le nom, le groupe de processus ou le groupe d'utilisateurs.

Options

- signal : Envoyer un signal spécifique au processus (par exemple, -9 pour envoyer un signal SIGKILL).
- f : Faire correspondre l'ensemble de la ligne de commande, pas seulement le nom du processus.
- u : Tuer les processus appartenant à un utilisateur spécifique.
- G : Tuer les processus appartenant à un groupe spécifique.
- g : Tuer les processus appartenant à un groupe de processus spécifique.
- n : Tuer les processus avec un PID plus élevé que celui spécifié.
- P : Tuer les processus avec un PPID spécifique (parent process ID).

Exemple

```
jordan@ubuntu:~$ pgill firefox
```

Permet de tuer tous les processus dont le nom contient "firefox" :

Vidéo

Section 6 - Administration avancée d'un système Linux

- Arrêter les processus grâce aux commandes "kill" et "pgill"

ps

Description

Permet d'afficher les processus en cours d'exécution sur le système.

Options

- e : Affiche tous les processus en cours d'exécution.
- f : Affiche une sortie formatée avec des informations supplémentaires sur les processus.
- u utilisateur : Affiche les processus pour l'utilisateur spécifié.
- x : Affiche les processus sans terminal de contrôle (par exemple, les processus système et les démons).

Exemple

```
jordan@ubuntu:~$ ps -ef
```

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	09:00	?	00:00:02	/sbin/init
root	2	0	0	09:00	?	00:00:00	[kthreadd]
root	3	2	0	09:00	?	00:00:00	[rcu_gp]
root	4	2	0	09:00	?	00:00:00	[rcu_par_gp]
root	6	2	0	09:00	?	00:00:00	[kworker/0:0H-kblockd]
jordan	2890	2876	0	09:01	?	00:00:00	/usr/lib/gvfs/gvfsd-http

Permet d'afficher tous les processus en cours d'exécution avec une sortie formatée

Vidéo

Section 6 - Administration avancée d'un système Linux

- Lister les processus du système avec "watch" et "ps"

pwd

Description

Permet d'afficher le répertoire dans lequel l'utilisateur est positionné avec son terminal.

Options

Exemple

```
jordan@ubuntu:~$ pwd  
/home/jordan
```

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Exploration de mes premières commandes Linux
- Exercice en ligne – commandes Linux de base

read

Description

La commande read permet de lire une ligne de texte depuis l'entrée standard (habituellement le clavier) et de l'affecter à une ou plusieurs variables.

Options

- p PROMPT : Affiche un message de prompt avant de lire les données.
- t TIMEOUT : Spécifie un délai d'attente (en secondes) pour lire les données. Si le délai est écoulé, la commande se termine sans lire de données.
- s : Rend l'entrée silencieuse, utile lors de la saisie de mots de passe.

Exemple

```
#!/bin/bash
echo "Veuillez entrer votre nom : "
read NOM
echo "Bonjour, $NOM !"
```

```
Veuillez entrer votre nom :
Jordan
Bonjour, Jordan !
```

Dans un script shell, vous pouvez utiliser read pour demander à l'utilisateur de saisir son nom et l'afficher.

Vidéo

Section 5 - Le scripting Bash

- Interagir avec l'utilisateur et effectuer des opérations mathématiques

rm

Description

Permet de supprimer un fichier ou un ensemble de fichiers et répertoires.

Options

- r : Supprime récursivement les répertoires et leurs contenus.
- f : Force la suppression sans demander de confirmation.

Exemple

```
jordan@ubuntu:~$ ls  
monfichier1 monfichier2 monfichier3 monfichier4
```

```
jordan@ubuntu:~$ rm monfichier2
```

```
jordan@ubuntu:~$ ls  
monfichier1 monfichier3 monfichier4
```

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Exploration de mes premières commandes Linux
- Exercice en ligne – commandes Linux de base

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice en ligne - Manipulation des fichiers et des dossiers

rmdir

Description

La commande "rmdir" est utilisée pour supprimer un ou plusieurs répertoires vides. Si le répertoire n'est pas vide, la commande échouera.

Options

-p : Supprime les répertoires parents spécifiés s'ils sont également vides. Si l'un des répertoires parents n'est pas vide, la suppression s'arrête.

--ignore-fail-on-non-empty : Ignore les échecs lorsqu'un répertoire n'est pas vide.

Exemple

Supprimer un répertoire vide nommé "empty_directory" :

```
$ rmdir empty_directory
```

Supprimer une structure de répertoires imbriqués "dir1/dir2/dir3" s'ils sont tous vides :

```
$ rmdir -p dir1/dir2/dir3
```

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice en ligne - Manipulation des fichiers et des dossiers

sed

Description

La commande sed (stream editor) permet de manipuler du texte en appliquant des transformations sur chaque ligne d'un fichier ou d'une entrée standard en utilisant des expressions régulières.

Options

- i : Modifier directement le fichier d'entrée, sans afficher le résultat sur la sortie standard.
- e : Appliquer plusieurs commandes de transformation en une seule fois.
- n : Ne pas afficher les lignes modifiées, sauf si l'option "p" est utilisée dans le script.

Exemple

```
echo "Hello, World!" > input.txt
sed 's/World/Monde/g' input.txt
sed -i 's/World/Monde/g' input.txt
sed -e 's/World/Monde/g' -e 's/Hello/Bonjour/g' input.txt
```

Hello, Monde!

Après l'utilisation de l'option -i, le contenu du fichier input.txt est modifié pour remplacer "World" par "Monde".

Vidéo

Section 5 - Le scripting Bash

- Utilisation de la commande sed

ssh-copy-id

Description

La commande ssh-copy-id permet de copier une clé publique SSH locale sur un serveur distant, en ajoutant la clé au fichier ~/.ssh/authorized_keys sur le serveur. Cela facilite l'authentification par clé et évite d'avoir à saisir un mot de passe lors de la connexion SSH au serveur.

Options

- i : Spécifie un fichier de clé différent à copier (par défaut, la clé publique dans ~/.ssh/id_rsa.pub est utilisée).
- p : Spécifie un port différent pour la connexion SSH au serveur distant (le port par défaut est 22).
- f : Force l'ajout de la clé, même si elle existe déjà dans le fichier authorized_keys sur le serveur distant.

Exemple

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@remote-server.com
```

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be
installed: "/home/jordan/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the
new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed
-- if you are prompted now it is to install the new keys
user@remote-server.com's password:
```

```
Number of key(s) added: 1
```

```
Now try logging into the machine, with:  "ssh
'user@remote-server.com'"
and check to make sure that only the key(s) you wanted were
added.
```

La clé publique ~/.ssh/id_rsa.pub de l'utilisateur local est copiée sur le serveur distant dans le fichier authorized_keys de l'utilisateur spécifié.

Vidéo

Section 4 - Administration de base de son système Linux

- Le protocole SSH et la génération de clés
- Echanger les clés SSH entre les deux machines Debian 11

ssh-keygen

Description

La commande `ssh-keygen` génère, gère et convertit les clés d'authentification pour la connexion SSH. Elle est couramment utilisée pour créer des paires de clés publique/privée pour l'authentification sans mot de passe lors des connexions SSH.

Options

- t : Spécifie le type de clé à générer (par exemple, rsa, dsa, ecdsa, ed25519).
- b : Spécifie la taille de la clé en bits (par exemple, 2048 ou 4096 pour une clé RSA).
- f : Spécifie le nom de fichier dans lequel stocker la clé (par défaut, la clé privée est stockée dans ~/.ssh/id_rsa et la clé publique dans ~/.ssh/id_rsa.pub).
- C : Ajoute un commentaire à la clé générée (utile pour identifier à qui appartient la clé).
- N : Fournit une nouvelle passphrase pour la clé.

Example

```
ssh-keygen -t rsa -b 4096 -f ~/.ssh/my_key -C
"jordan@example.com"
```

```
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in
/home/jordan/.ssh/my_key.
Your public key has been saved in
/home/jordan/.ssh/my_key.pub.
The key fingerprint is:
SHA256:XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
jordan@example.com
The key's randomart image is:
+---[RSA 4096]---+
|           o=.o+o..      |
|          . o=oB.o       |
|          +.=*Bo        |
|          . =oO=o .      |
|          o S=oE .       |
|          . o .         |
|                       |
|                       |
|                       |
```

```
|  
+-----[SHA256]-----+
```

Cet exemple génère une paire de clés RSA de 4096 bits, stockée dans les fichiers `~/.ssh/my_key` (clé privée) et `~/.ssh/my_key.pub` (clé publique), avec un commentaire "jordan@example.com".

Vidéo

Section 4 - Administration de base de son système Linux

- Le protocole SSH et la génération de clés
- Echanger les clés SSH entre les deux machines Debian 11

sudo

Description

sudo (super user do) permet d'exécuter des commandes en tant qu'un autre utilisateur, généralement en tant que superutilisateur (root).

Options

- u <utilisateur> : Exécute la commande en tant que l'utilisateur spécifié.
- k : Révoque les droits sudo temporaires.
- l : Liste les commandes autorisées pour l'utilisateur actuel.

Exemple

```
jordan@ubuntu:~$ apt-get update
Reading package lists... Done
E: Could not open lock file /var/lib/apt/lists/lock - open
(13: Permission denied)
E: Unable to lock directory /var/lib/apt/lists/
W: Problem unlinking the file /var/cache/apt/pkgcache.bin -
RemoveCaches (13: Permission denied)
W: Problem unlinking the file
/var/cache/apt/srcpkgcache.bin - RemoveCaches (13:
Permission denied)
```

```
jordan@ubuntu:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security
InRelease [88.7 kB]
Hit:2 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:3 http://archive.ubuntu.com/ubuntu bionic-updates
InRelease [88.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic-backports
InRelease [74.6 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic-
updates/universe amd64 Packages [1718 kB]
Fetched 1970 kB in 4s (451 kB/s)
Reading package lists... Done
```

Dans l'exemple ci-dessus, on voit ainsi que la commande "apt-get update" n'est pas disponible à l'utilisateur jordan. Mais que grâce à la commande sudo, celui-ci peut obtenir temporairement les droits root lui permettant d'exécuter correctement la commande.

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Exploration de mes premières commandes Linux
- Exercice en ligne – commandes Linux de base

Section 4 - Administration de base de son système Linux

- Exercice en ligne - Apprendre à manipuler les utilisateurs
- [Correction] Exercice en ligne - Apprendre à manipuler les utilisateurs

top

Description

Permet de surveiller en temps réel les processus en cours d'exécution sur le système, ainsi que leur utilisation des ressources (CPU, mémoire, etc.).

Options

(commandes interactives) :

t : Basculer entre l'affichage du temps d'utilisation du processeur et l'affichage du résumé

m : Basculer entre l'affichage de l'utilisation de la mémoire et l'affichage du résumé

A : Trier les processus par utilisation du processeur

M : Trier les processus par utilisation de la mémoire

P : Trier les processus par priorité

u : Afficher les processus pour un utilisateur spécifique

k : Tuer un processus en entrant son PID

r : Modifier la priorité (nice value) d'un processus en entrant son PID

q : Quitter top

Exemple

```
jordan@ubuntu:~$ top
```

```
top - 12:19:23 up 3 days, 1:15, 1 user, load average:
0.00, 0.01, 0.05
Tasks: 125 total, 1 running, 124 sleeping, 0 stopped,
0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0
hi, 0.0 si, 0.0 st
KiB Mem : 2046764 total, 605924 free, 781472 used,
555368 buff/cache
KiB Swap: 1048572 total, 1048572 free, 0 used
```

Permet d'afficher l'état des processus en cours d'exécution et l'utilisation des ressources en temps réel

Vidéo

Section 6 - Administration avancée d'un système Linux

- La commande top - compréhension et interaction

touch

Description

Permet de créer un fichier ou un ensemble de fichier. Peut également être utilisé pour mettre à jour la date de la dernière modification du fichier.

Options

-a : Modifie uniquement l'heure d'accès du fichier.

-m : Modifie uniquement l'heure de modification du fichier.

-t : Permet de définir une date et une heure spécifiques pour la modification et l'accès.

Exemple

```
jordan@ubuntu:~$ ls
```

```
monfichier1 monfichier2 monfichier3 monfichier4
```

```
jordan@ubuntu:~$ touch monfichier5 monfichier6
```

```
jordan@ubuntu:~$ ls
```

```
monfichier1 monfichier2 monfichier3 monfichier4 monfichier5  
monfichier6
```

Vidéo

Section 2 – Premières utilisations d'Ubuntu et commandes de base

- Exploration de mes premières commandes Linux
- Exercice en ligne – commandes Linux de base

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Commandes et manipulation des fichiers et des répertoires
- Exercice en ligne - Manipulation des fichiers et des dossiers
- [Correction] Exercice en ligne - Manipulation des fichiers et des dossiers

uptime

Description

Permet d'afficher le temps écoulé depuis le dernier démarrage du système, ainsi que le nombre d'utilisateurs connectés et la charge moyenne du système.

Options

Uptime n'a pas d'options spécifiques.

Exemple

```
jordan@ubuntu:~$ uptime
```

```
12:10:25 up 1:42, 1 user, load average: 0.00, 0.01, 0.05
```

Dans cet exemple, le système est en fonctionnement depuis 1 heure et 42 minutes, avec 1 utilisateur connecté. La charge moyenne du système sur les dernières 1, 5 et 15 minutes est de 0.00, 0.01 et 0.05, respectivement.

Vidéo

Section 6 - Administration avancée d'un système Linux

- Visualiser la charge des processeurs avec : uptime, iostat, et mpstat

usermod

Description

La commande "usermod" permet de modifier les informations d'un compte utilisateur existant sous Linux, telles que le groupe principal, le répertoire d'accueil, le nom de connexion, etc.

Options

- a, --append : Ajoute l'utilisateur aux groupes supplémentaires, sans supprimer les groupes existants.
- c, --comment : Modifie la description (commentaire) de l'utilisateur.
- d, --home : Modifie le répertoire d'accueil de l'utilisateur.
- e, --expiredate : Modifie la date d'expiration du compte.
- g, --gid : Modifie le groupe principal de l'utilisateur.
- G, --groups : Modifie les groupes supplémentaires de l'utilisateur.
- l, --login : Modifie le nom de connexion (login) de l'utilisateur.
- L, --lock : Verrouille le compte utilisateur.
- U, --unlock : Déverrouille le compte utilisateur.

Exemple

Modifier le répertoire home de l'utilisateur "jordan" en "/new/home/jordan" :

```
sudo usermod -d /new/home/jordan jordan
```

Ajouter l'utilisateur "jordan" aux groupes "group1" et "group2" :

```
sudo usermod -a -G group1,group2 jordan
```

Vidéo

Section 4 - Administration de base de son système Linux

- Quelques manipulations pour créer et supprimer des groupes
- Exercice en ligne - Apprendre à manipuler les groupes
- [Correction] Exercice en ligne - Apprendre à manipuler les groupes

vim

Description

Permet d'éditer des fichiers texte dans un terminal à l'aide de l'éditeur de texte "Vi IMproved", un éditeur puissant et très configurable.

Options

- R : Ouvre le fichier en mode "read-only" (lecture seule).
- N : Désactive la compatibilité avec Vi, utilisant uniquement les fonctionnalités Vim.
- o : Ouvre plusieurs fichiers en mode horizontal.
- O : Ouvre plusieurs fichiers en mode vertical.
- p : Ouvre plusieurs fichiers dans des onglets.
- u : Spécifie un fichier de configuration Vim personnalisé.

Navigation dans vim :

- h : se déplacer vers la gauche
- j : se déplacer vers le bas
- k : se déplacer vers le haut
- l : se déplacer vers la droite

- w : se déplacer d'un mot vers la droite
- b : se déplacer d'un mot vers la gauche

- ^ : se déplacer au début de la ligne
- \$: se déplacer à la fin de la ligne

Insertion et suppression dans vim:

- i : insérer du texte à l'endroit où se situe le curseur
- I : insérer du texte au début de la ligne
- a : insérer du texte après le curseur
- A : insérer du texte à la fin de la ligne

- x : supprime le caractère situé sous le curseur
- dw : supprime le mot situé au niveau du curseur
- dd : supprime/coupe la ligne où se situe le curseur
- D : supprime l'ensemble des éléments entre le curseur et la fin de la ligne

- yy : copie la ligne actuelle
- p : colle l'élément situé dans le presse-papier

- u : annule la dernière action effectuée

Commandes principales :

- :w : sauvegarder les modifications du fichier
- :q : quitter le fichier
- :wq : sauvegarde les modifications et quitte le fichier
- :x : commande équivalente à ":wq"
- :set nu : affiche le numéro des lignes
- :set nonu : désactive l'affichage du numéro des lignes

Exemple

Vidéo

Section 3 – Gestion des fichiers et des dossiers sur Linux

- Vim - Installation et navigation dans l'éditeur de texte
- Vim - Insérer et supprimer du texte
- Vim - Quelques commandes de base

watch

Description

Permet d'exécuter une commande spécifiée à intervalles réguliers et d'afficher les résultats.

Options

-n SECONDES : Définit l'intervalle entre les exécutions de la commande en secondes (par défaut : 2 secondes).

-d : Met en évidence les différences entre les mises à jour successives.

-t : Supprime l'affichage de l'en-tête avec l'heure actuelle, le nom de la commande et l'intervalle.

Exemple

```
jordan@ubuntu:~$ watch -n 5 free -m
```

```
Every 5.0s: free -m
```

	total	used	free	shared
buff/cache	available			
Mem:	7848	1206	4746	236
1895	6222			
Swap:	2047	0	2047	

Permet de surveiller l'utilisation de la mémoire toutes les 5 secondes

Vidéo

Section 6 - Administration avancée d'un système Linux

- Lister les processus du système avec "watch" et "ps"