

Linux pour DevOps

DEVOPS BOOTCAMP

FORMATEUR M. Ngor SECK

Samane corporation: <https://www.samanecorporation.com>

nseck@samanecorporation.com

ngorsecka@gmail.com

Définition

Linux est un système d'exploitation de type Unix, initialement créé par Linus Torvalds en 1991. C'est un système d'exploitation open source et gratuit, largement utilisé dans le monde entier, tant sur les serveurs que sur les postes de travail.

- Voici quelques caractéristiques clés de Linux :
 - Noyau Linux : Le noyau Linux est le cœur du système d'exploitation. Il gère les ressources du matériel, les processus système, la mémoire, les systèmes de fichiers, les pilotes de périphériques, et d'autres aspects fondamentaux.
 - Open Source : Linux est distribué sous une licence open source, ce qui signifie que son code source est accessible au public. Cela encourage la collaboration et permet à quiconque de modifier, distribuer et améliorer le système d'exploitation.
 - Distribution Linux : Linux est généralement distribué sous forme de distributions, ou "distros". Chaque distribution est un ensemble de logiciels système basé sur le noyau Linux. Certaines distributions populaires incluent

Ubuntu, Fedora, Debian, CentOS, et Arch Linux.

- Interface en ligne de commande : Bien que de nombreuses distributions Linux disposent d'une interface graphique, l'interface en ligne de commande (CLI) est également largement utilisée. La ligne de commande offre un contrôle plus avancé et est fréquemment utilisée dans l'administration système et le développement.
- Multi-Utilisateur et Multi-Processus : Linux est un système multi-utilisateur, ce qui signifie qu'il peut simultanément gérer plusieurs utilisateurs. Il est également multi-processus, capable d'exécuter de nombreux processus en même temps.
- Sécurité : Linux est réputé pour sa stabilité et sa sécurité. Les systèmes Linux sont souvent utilisés comme serveurs web, serveurs de bases de données, et pour d'autres applications nécessitant une robustesse et une sécurité élevées.
- Compatibilité matérielle : En raison de son support pour de nombreux pilotes de périphériques, Linux peut être installé sur une variété d'architectures matérielles, des ordinateurs personnels aux serveurs d'entreprise et aux systèmes intégrés.
- Communauté active : la communauté open source de Linux est vaste et active. Des milliers de développeurs du monde entier contribuent au développement du noyau Linux et aux logiciels qui l'accompagnent.

Les différentes familles Linux

- Debian-based :
 - Exemples de distributions : Debian, Ubuntu, Linux Mint.

-
- Caractéristiques : Systèmes stables, processus de gestion des paquets basé sur apt, utilisation fréquente du format de paquet .deb.
 - Red Hat-based :
 - Exemples de distributions : Red Hat Enterprise Linux (RHEL), CentOS, Fedora.
 - Caractéristiques : Orientées vers les entreprises, utilisation du gestionnaire de paquets rpm, utilisation du format de paquet .rpm.
 - Arch-based :
 - Exemples de distributions : Arch Linux, Manjaro.
 - Caractéristiques : Design minimaliste, orienté vers les utilisateurs avancés, gestionnaire de paquets pacman, modèle de rolling release (mises à jour continues).
 - Slackware :
 - Exemple de distribution : Slackware.
 - Caractéristiques : L'une des plus anciennes distributions, minimaliste, met l'accent sur la simplicité et la stabilité, utilise le gestionnaire de paquets .tgz.
 - Gentoo-based :
 - Exemple de distribution : Gentoo.
 - Caractéristiques : Système source, où les logiciels sont compilés localement pour optimisation, hautement personnalisable, mise à jour continue.
 - SUSE-based :
 - Exemples de distributions : openSUSE, SUSE Linux Enterprise Server (SLES).
-

-
- Caractéristiques : Axées sur les entreprises, utilisation du gestionnaire de paquets zypper, utilisation fréquente du format de paquet .rpm.
 - Embedded Linux :
 - Exemples de distributions : OpenEmbedded, Buildroot.
 - Caractéristiques : Conçues pour les systèmes embarqués, souvent utilisées dans des dispositifs tels que des routeurs, des appareils IoT (Internet des objets) et des systèmes embarqués.
 - Raspberry Pi OS :
 - Exemple de distribution : Raspberry Pi OS (anciennement appelé Raspbian).
 - Caractéristiques : Conçue spécifiquement pour les ordinateurs Raspberry Pi, basée sur Debian, optimisée pour l'architecture ARM.

Le CLI de linux

Le CLI (Command Line Interface) de Linux, également appelé terminal ou shell, est un environnement textuel qui permet aux utilisateurs d'interagir avec le système d'exploitation grâce à des lignes de commande. Il propose une alternative à l'interface graphique (GUI) pour accomplir des tâches, et il est particulièrement apprécié des utilisateurs avancés, des administrateurs système et des développeurs. Voici quelques points clés concernant le CLI de Linux :

- **Terminal :**

Un terminal est l'application qui fournit l'interface en ligne de commande. Il peut être ouvert sur le bureau de l'environnement de bureau ou accédé via des raccourcis clavier.

- **Shell :**

Le shell est le programme qui interprète les commandes que vous saisissez dans le terminal. Les shells les plus courants incluent Bash (Bourne Again SHell), Zsh, et Fish.

- **Prompt :**

Le prompt est l'invite qui indique que le shell est prêt à accepter une commande. Il affiche généralement des informations telles que le nom d'utilisateur, le nom de l'hôte et le répertoire actuel.

Exemple de prompt Bash :

Unset

vagrant@ubuntu-jammy:~\$

vagrant@ubuntu-jammy:~#

- **Commandes :**

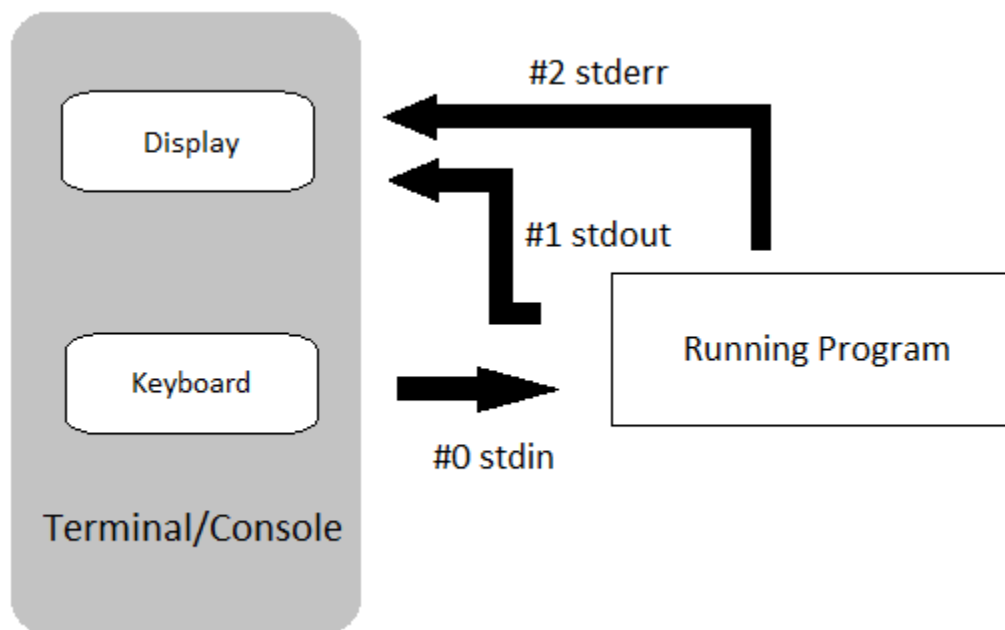
Les commandes sont des instructions que vous saisissez dans le terminal pour effectuer des tâches spécifiques. Par exemple, la commande ls est utilisée pour lister les fichiers d'un répertoire.

- **Options et arguments :**

Les commandes peuvent être accompagnées d'options (indiquées par des drapeaux comme -l) et d'arguments (informations supplémentaires nécessaires pour exécuter la commande). Par exemple :

- **Stdin / Stdout / stderr**

Sur Linux, les concepts de stdin (standard input), stdout (standard output), et stderr (standard error) sont fondamentaux pour comprendre comment les commandes interagissent avec les flux de données. Ces concepts sont liés à la manière dont les processus Linux traitent l'entrée, la sortie et les erreurs.



- stdin (standard input) :

Description : stdin est le canal standard d'entrée. Il représente la source d'entrée pour un processus.

Utilisation : Par défaut, stdin est généralement lié au clavier. Cependant, il peut être redirigé vers un fichier ou une autre source d'entrée.

- stdout (standard output) :

Description : stdout est le canal standard de sortie. Il représente la destination de sortie pour un processus.

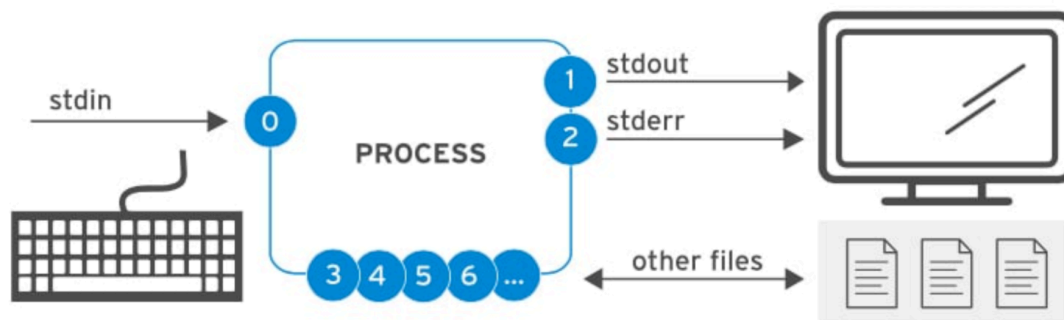
Utilisation : Par défaut, stdout est généralement lié à l'écran (terminal).

- stderr (standard error) :

Description : stderr est le canal standard pour les messages d'erreur. Il est utilisé pour envoyer des messages d'erreur ou de diagnostic à l'utilisateur ou à un fichier.

Utilisation : Par défaut, stderr est généralement lié à l'écran (terminal).

- **Redirection et pipes :**



La sortie d'une commande peut être redirigée vers un fichier (> pour la sortie standard, 2> pour la sortie d'erreur) ou utilisée comme entrée pour une autre commande (| pour le pipe).

Exemple 1:

Unset

```
vagrant@ubuntu-jammy:~$ find /etc/ -name "*.crt" > crt.txt 2> /dev/null  
  
vagrant@ubuntu-jammy:~$ cat crt.txt  
/etc/ssl/certs/ca-certificates.crt
```

Les tubes ("pipe" en anglais) sont représentés par l'opérateur "|". Les données vont de gauche à droite. La figure suivante indique comment la sortie standard du premier processus est redirigée vers l'entrée standard du second processus.

PROCESSUS1 (stdout) ---- | ---- (stdin) PROCESSUS2

Exemple 2:

Unset

```
ps aux | grep login
```

Les Permissions sous linux

Sur les systèmes Linux, les privilèges sont utilisés pour définir les niveaux d'accès et de contrôle d'un utilisateur ou d'un processus. Il existe différents types de privilèges, dont certains sont liés aux permissions sur les fichiers, tandis que d'autres concernent l'accès aux ressources système ou aux capacités spéciales. Voici quelques-uns des privilèges et des concepts clés sur Linux :

- **Privilèges utilisateur :**

- Utilisateurs réguliers : Les utilisateurs normaux ont des privilèges limités et peuvent accéder à des fichiers et répertoires en fonction des permissions définies.
- Superutilisateur (root) : Le superutilisateur a des privilèges étendus et peut effectuer n'importe quelle opération sur le système. L'accès au compte root est généralement limité et nécessite l'utilisation de la commande sudo.

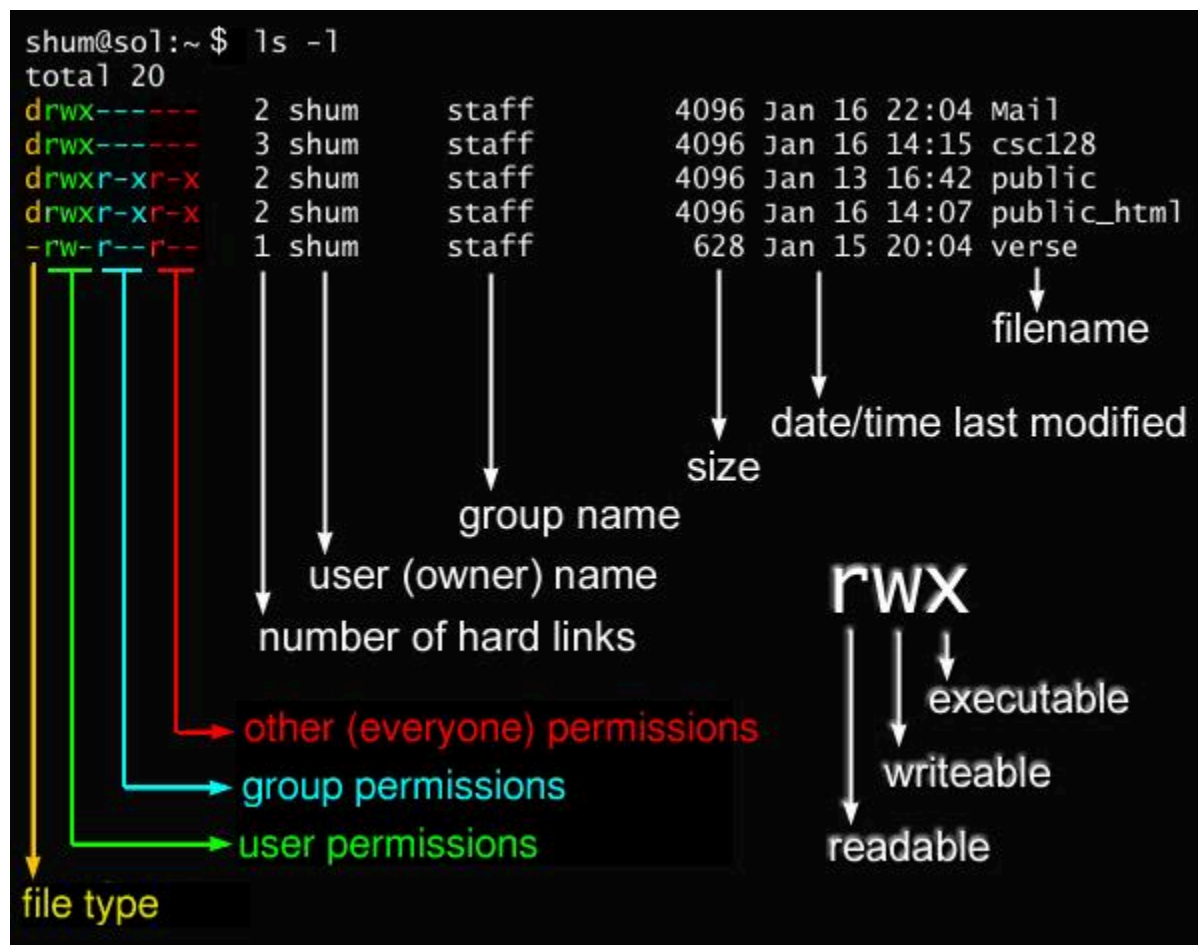
- **Permissions sur les fichiers :**

Read (r) : Permet la lecture du fichier.

Write (w) : Permet la modification du fichier.

Execute (x) : Permet l'exécution d'un fichier (ou la traversée d'un répertoire).

Ces permissions sont attribuées aux propriétaires du fichier, au groupe du fichier et à tous les autres utilisateurs.



- **Accès à sudo :**

Permet à un utilisateur d'exécuter des commandes avec les privilèges du superutilisateur (root). Les utilisateurs doivent être spécifiés dans le fichier `sudoers`.

Éditeur de texte

Pour écrire un script ou éditer des fichiers de configuration, nous avons besoin d'un bon éditeur de texte. Linux vient avec plusieurs éditeurs de texte par défaut en ligne de commande :

Unset

```
sudo update-alternatives --config editor  
There are 4 choices for the alternative editor (providing /usr/bin/editor).
```

- Vi/vim :

vi et vim sont deux éditeurs de texte en ligne de commande très populaires sur les systèmes Unix/Linux. vim est en fait une version améliorée de vi, intégrant de nombreuses fonctionnalités supplémentaires et des améliorations par rapport à l'éditeur d'origine vi.

Pour utiliser vi ou vim il suffit de taper la commande :

Unset

```
vi <nom_du_fichier.ext>
```

- Principaux raccourcis et commandes :
 - Mode :
 - i : Basculer en mode insertion (pour éditer le texte).
 - Esc : Revenir en mode normal (quitter le mode insertion).

-
- Commandes de base :
 - :w : Enregistrer le fichier.
 - :q : Quitter l'éditeur.
 - :wq : Enregistrer et quitter.
 - Navigation :
 - h, j, k, l : Déplacements respectivement à gauche, bas, haut, droite.
 - Ctrl + u, Ctrl + d : Défilement demi-écran vers le haut ou le bas.
 - G : Aller à la fin du fichier.
 - gg : Aller au début du fichier.
 - Copier/Coller (en mode visuel) :
 - v : Basculer en mode visuel (pour sélectionner du texte).
 - y : Copier le texte sélectionné.
 - p : Coller le texte copié.
 - Recherche et Remplacement :
 - /motif : Rechercher un motif dans le fichier.
 - :s/motif/remplacement/g : Remplacer toutes les occurrences d'un motif.

Lecteur de texte

- cat / tac

On utilise plus couramment ``cat`` pour envoyer du texte vers la sortie standard.

Unset

```
cat <nom_du_fichier.ext>
```

`tac` est plutôt utilisé pour lire le fichier en inverse

- head et tail

On utilise souvent les commandes `head` et `tail` pour analyser les fichiers de journaux. Par défaut, ces commandes affichent 10 lignes. En voici les utilisations les plus courantes :

- Afficher les 20 premières lignes de `/var/log/messages` :

Unset

```
head -n 20 /var/log/syslog
```

```
head -20 /var/log/syslog
```

- Afficher les 20 dernières lignes de `/etc/aliases` :

Unset

```
tail -n 20 /var/log/syslog
```

```
tail -20 /var/log/syslog
```

- tee

La commande `tee` permet à la fois de lire un flux et de le rediriger. Par exemple, `tee` donne la sortie et l'écrit dans le fichier `ls1.txt` :

Unset

```
ls | tee ll1.txt
```

Manipulation de texte

1. Compter lignes, mots et octets avec la commande `wc`

La commande `wc` compte le nombre d'octets, de mots et de lignes dans les fichiers.

Les options suivantes vous permettent de sélectionner ce qui nous intéresse :

- `-l` compte le nombre de lignes
- `-w` compte le nombre de mots (words)
- `-c` compte le nombre d'octets
- `-m` compte le nombre de caractères
- sans argument, `wc` compte ce qui est saisi dans `stdin`.

Par exemple :

- a. Connaître le nombre d'utilisateurs

Unset

```
wc -l /etc/passwd
```

b. Connaître le nombre de caractères d'une phrase

Unset

```
echo "Je suis un ingénieur devops" | wc -m
```

```
echo "Je suis un ingénieur devops" | tee char.txt | wc -m
```

2. Découper les fichiers avec la commande split

La commande split peut découper un fichier en plusieurs fichiers plus petits à partir de critères comme la taille ou le nombre de lignes. Par exemple, nous pouvons découper /etc/passwd en fichiers de 5 lignes chacun :

Unset

```
split -l 5 /etc/passwd
```

Cette commande va créer des fichiers appelés xaa, xab, xac, xad, etc., chaque fichier contenant au plus cinq lignes.

Tentez et vérifiez :

Unset

```
split -l 5 /etc/passwd
```

```
-rw-rw-r-- 1 vagrant vagrant 235 Jan 12 21:55 xab
```

```
-rw-rw-r-- 1 vagrant vagrant 263 Jan 12 21:55 xac
-rw-rw-r-- 1 vagrant vagrant 350 Jan 12 21:55 xad
-rw-rw-r-- 1 vagrant vagrant 301 Jan 12 21:55 xae
-rw-rw-r-- 1 vagrant vagrant 251 Jan 12 21:55 xaf
-rw-rw-r-- 1 vagrant vagrant 265 Jan 12 21:55 xag
-rw-rw-r-- 1 vagrant vagrant  42 Jan 12 21:55 xah
```

Il est possible de donner un préfixe plus significatif que "x", comme "pass-5" :

Unset

```
split -dl 5 /etc/passwd password-

-rw-rw-r-- 1 vagrant vagrant 189 Jan 12 23:23 password-00
-rw-rw-r-- 1 vagrant vagrant 235 Jan 12 23:23 password-01
-rw-rw-r-- 1 vagrant vagrant 263 Jan 12 23:23 password-02
-rw-rw-r-- 1 vagrant vagrant 350 Jan 12 23:23 password-03
-rw-rw-r-- 1 vagrant vagrant 301 Jan 12 23:23 password-04
-rw-rw-r-- 1 vagrant vagrant 251 Jan 12 23:23 password-05
-rw-rw-r-- 1 vagrant vagrant 265 Jan 12 23:23 password-06
-rw-rw-r-- 1 vagrant vagrant  42 Jan 12 23:23 password-07
```

Cette commande crée des fichiers identiques à la commande précédente, mais ils sont désormais nommés passwd-00, passwd-01, ...

Vérifions le nombre de lignes des fichiers générés

```
vagrant@ubuntu-jammy:~$ wc -l password*
 5 password-00
 5 password-01
 5 password-02
 5 password-03
 5 password-04
 5 password-05
 5 password-06
 1 password-07
36 total
vagrant@ubuntu-jammy:~$ █
```

3. Sélectionner les champs et les caractères avec cut

La commande cut peut extraire une plage de caractère ou de champ de chaque ligne d'un texte.

Exemple : Récupérer les utilisateurs et leurs répertoires de travail

```
vagrant@ubuntu-jammy:~$ cut -d: -f1,6 /etc/passwd
root:/root
daemon:/usr/sbin
bin:/bin
sys:/dev
sync:/bin
games:/usr/games
man:/var/cache/man
lp:/var/spool/lpd
mail:/var/mail
news:/var/spool/news
uucp:/var/spool/uucp
proxy:/bin
www-data:/var/www
backup:/var/backups
list:/var/list
irc:/run/ircd
gnats:/var/lib/gnats
nobody:/nonexistent
systemd-network:/run/systemd
systemd-resolve:/run/systemd
messagebus:/nonexistent
systemd-timesync:/run/systemd
syslog:/home/syslog
_apt:/nonexistent
tss:/var/lib/tpm
uidd:/run/uidd
tcpdump:/nonexistent
sshd:/run/sshd
pollinate:/var/cache/pollinate
landscape:/var/lib/landscape
fwupd-refresh:/run/systemd
vagrant:/home/vagrant
ubuntu:/home/ubuntu
lxd:/var/snap/lxd/common/lxd
```

Ref. @mmkmou