

La ligne de commande

Une **ligne de commande**, également appelée interface en ligne de commande (CLI), est un moyen d'interagir avec un système d'exploitation en saisissant des commandes à l'aide d'un clavier au lieu d'utiliser une interface graphique. Les commandes sont généralement entrées dans un terminal ou une fenêtre de commande et exécutées par le système d'exploitation. Les commandes peuvent effectuer des tâches telles que la gestion de fichiers, la modification de paramètres de configuration, la gestion des processus et la surveillance du système. Les lignes de commandes sont utilisés à l'aide d'un interpréteur de commande (par exemple BASH).

Une fois connecté à un terminal texte, un programme nommé shell est lancé automatiquement. Il permet de lancer des commandes.

Syntaxe des commandes

La **ligne de commande** regroupe une ou plusieurs instructions qui sont saisies et se termine par un retour à la ligne. La commande est composée d'un nom qui décrit une action ou un programme, éventuellement suivie d'arguments.

Par exemple la ligne de commande `cal` comporte la commande `cal` sans argument.

La ligne de commande `cal -m apr` comporte la commande `cal` avec 2 arguments : `-m` et `apr`.

La ligne de commande `echo "Bonjour Alpinux"` comporte la commande `echo` avec 1 argument, `"Bonjour Alpinux"`.

Les arguments d'une commande peuvent être des options ou des arguments, selon le contexte dans lequel ils sont utilisés.

Les options sont des arguments supplémentaires qui peuvent être utilisés pour personnaliser le comportement de la commande. Elles peuvent être courtes (précédées d'un tiret simple (-)) ou longues (précédées d'un double tiret (--)).

Les arguments sont des informations nécessaires à la commande pour accomplir son travail. Ils peuvent inclure des noms de fichiers, des répertoires, des noms d'utilisateurs, etc.

Par exemple, dans la commande `ls -l /home/user`, `-l` est une option qui demande à la commande `ls` d'afficher les fichiers et répertoires sous forme de liste détaillée, tandis que `/home/user` est un argument qui spécifie le répertoire à afficher.

Il est important de comprendre les différences entre les options et les arguments pour utiliser correctement les commandes en ligne de commande sous Linux.

Options mono-caractère

Les options mono-caractère sont des options courtes qui peuvent être utilisées pour personnaliser le comportement des commandes en ligne de commande sous Linux. Elles sont généralement précédées d'un tiret simple (-) et peuvent être combinées pour obtenir des effets plus complexes.

Par exemple, la commande `ls` peut être utilisée avec l'option mono-caractère `-l` pour afficher les fichiers et répertoires sous forme de liste détaillée. La combinaison des options `-l` et `-h` peut être utilisée pour afficher les tailles de fichier de manière lisible pour les humains.

```
ls -l -h
ls -h -l
ls -lh
ls -hl
```

Certaines options en ligne de commande peuvent être précédées d'un signe plus (+) au lieu d'un tiret simple (-). Ce type d'options sont généralement utilisées pour ajouter ou activer des fonctionnalités supplémentaires dans les programmes.

Par exemple, la commande `ls` peut être utilisée avec l'option `+l` pour afficher les fichiers et répertoires sous forme de liste détaillée, similaire à l'utilisation de l'option `-l`.

Il est important de noter que l'utilisation de signes plus (+) pour les options est assez rare et n'est généralement pas standard pour la plupart des commandes en ligne de commande sous Linux. La plupart des options sont précédées d'un tiret simple (-). Il est donc toujours préférable de consulter la documentation pour comprendre les options disponibles et comment les utiliser correctement.

Certaines options en ligne de commande peuvent prendre une chaîne de caractères en argument. Ces options sont généralement utilisées pour fournir des informations supplémentaires au programme sur la façon de se comporter ou sur les données à utiliser.

Par exemple, la commande `grep` peut être utilisée avec l'option `-e` pour rechercher une expression régulière spécifiée dans un fichier ou sur l'entrée standard. L'expression régulière est spécifiée comme une chaîne de caractères après l'option `-e`.

Supposons que vous avez un fichier nommé `file.txt` qui contient du texte. Pour rechercher la chaîne de caractères `"sample"` dans ce fichier, vous pouvez utiliser la commande `grep` avec l'option `-e` :

```
grep -e "sample" file.txt
```

Vous pouvez écrire la commande `grep` de cette façon :

```
grep -esample file.txt
```

Cela fonctionnera de la même manière que la commande précédente, avec l'option `-e` suivie de la chaîne de caractères `"sample"`. La commande affichera la ligne du fichier `"file.txt"` qui contient la chaîne de caractères `"sample"`.

Il est important de noter que le format des options qui prennent une chaîne de caractères en argument peut varier pour chaque commande. Il est donc toujours préférable de consulter la documentation pour comprendre les options disponibles et comment les utiliser correctement.

Les options mono-caractère peuvent varier pour chaque commande, et il est souvent utile de consulter la documentation pour comprendre les options disponibles et comment les utiliser correctement.

Les options longues

Les options longues sont généralement précédées d'un double tiret (-) et peuvent inclure des mots entiers pour décrire plus clairement leur fonctionnement.

Par exemple, la commande `ls` peut être utilisée avec l'option longue `--human-readable` pour afficher la taille des fichiers sous forme conviviale en utilisant des unités de taille, similaire à l'utilisation de l'option `-h`.

La séparation des options et des arguments

La séparation des options et des arguments peut être faite en utilisant un double tiret (-) pour signaler la fin des options et le début des arguments. Tout ce qui suit le double tiret est considéré comme un argument, même s'il commence par un tiret simple (-).

Par exemple, considérons la commande `cp`, qui permet de copier des fichiers d'un emplacement à un autre. Si vous souhaitez copier un fichier nommé `-f` à un emplacement nommé `destination`, vous pouvez utiliser la commande suivante:

```
cp -- -f destination
```

Dans ce cas, le double tiret (-) signale la fin des options et le début des arguments. Le premier argument, `-f`, est considéré comme un nom de fichier, même s'il commence par un tiret simple (-). Le second argument, `destination`, est considéré comme l'emplacement de destination pour la copie de fichier.

L'utilisation de ce double tiret peut être utile lorsqu'il est nécessaire de traiter des arguments qui commencent par un tiret simple (-), qui pourraient autrement être interprétés comme des options.

Aspect de la ligne de commandes

Sous Linux, l'invite de commande peut prendre différentes formes en fonction de la configuration. Par exemple, elle peut être configurée pour afficher le nom d'utilisateur, le nom d'hôte et le répertoire courant, ou seulement le nom d'utilisateur et le répertoire courant. La couleur et les caractères utilisés pour l'affichage peuvent également être personnalisés.

Il est également possible de changer le comportement de l'invite de commande en utilisant des alias, des variables d'environnement et d'autres outils de configuration. Cela peut aider à améliorer l'efficacité et la productivité en ligne de commande.

L'invite de commande

L'**invite de commande est l'endroit** où les utilisateurs peuvent **saisir des commandes** pour interagir avec un système d'exploitation.

Il s'agit généralement d'une fenêtre ou d'un **terminal** (appelé SHELL) qui **affiche un curseur clignotant**, où **l'utilisateur peut saisir des commandes** et les exécuter en appuyant sur la touche "Entrée".

Les commandes saisies dans l'invite de commande **sont** interprétées par le système d'exploitation et **exécutées** en conséquence.

Le dernier caractère de l'invite de commande peut être un dollar "\$" ou un dièse "#". Cela dépend du niveau de privilèges de l'utilisateur actuel.

Un dollar \$ indique que l'utilisateur actuel est un utilisateur normal, sans privilèges d'administration.

Un dièse # indique que l'utilisateur actuel est un utilisateur avec des privilèges d'administration, également appelé "superutilisateur" ou "root".

Cela aide à identifier rapidement le niveau de privilèges de l'utilisateur actuel et peut être utile pour éviter de saisir accidentellement des commandes dangereuses avec des privilèges élevés.

Par exemple :

```
cedric@eeepc:~$
```

cedric » correspond à l'utilisateur connecté.

eeepc » correspond au nom de l'hôte, c'est-à-dire au nom de l'ordinateur.

~ » correspond au répertoire courant. Lorsqu'il est représenté par le symbole ~, cela indique qu'il s'agit du dossier personnel de l'utilisateur, appelé en anglais **home**.

\$ » indique que l'utilisateur est de type normal. Avec le caractère #, il indiquera qu'il s'agit d'un super-utilisateur, identifié **root**.

Applications avec une interface graphique

Il est possible d'exécuter des applications avec une interface graphique depuis un shell sous Linux.

Par exemple, pour exécuter une application telle que "gedit" avec une interface graphique depuis un shell, vous pouvez utiliser la commande suivante :

```
gedit
```

Cette commande ouvrira l'application fenêtrée **gedit**.

Cette méthode n'est pas toujours la plus pratique ou la plus efficace pour exécuter des applications avec une interface graphique depuis un shell. Dans de nombreux cas, il est plus facile de simplement cliquer sur l'icône de l'application dans le menu ou le gestionnaire de fichiers. Cependant, cette méthode peut être utile pour exécuter des applications depuis un script ou pour effectuer des tâches de ligne de commande.

Les raccourcis clavier en mode texte

Il existe de nombreux raccourcis clavier qui peuvent être utilisés en mode texte sous Linux pour effectuer des tâches rapidement et efficacement. Voici quelques-uns des raccourcis clavier les plus couramment utilisés :

- **Tab ↵** : Complète automatiquement un nom de fichier ou de répertoire.
- **↑** : Affiche la dernière commande entrée.
- **↓** : Affiche la commande précédente entrée.

- **Ctrl + C** : Interrompt une commande en cours d'exécution.
- **Ctrl + Z** : Suspend une commande en cours d'exécution et la place en arrière-plan.
- **Ctrl + D** : Termine une session en ligne de commande.
- **Ctrl + L** : Efface l'écran.
- **Ctrl + R** : Recherche dans l'historique des commandes.

- **Ctrl + ↑ Shift + C** : Copie dans le presse papier la sélection.
- **Ctrl + ↑ Shift + V** : Colle le contenu du presse papier à l'endroit du curseur. Équivalent : **Shift + Insert**

- **Ctrl + A** : Déplace le curseur au début de la ligne actuelle.
- **Ctrl + E** : Déplace le curseur à la fin de la ligne actuelle.
- **Ctrl + Left** : Déplace le curseur d'un mot à gauche.
- **Ctrl + Right** : Déplace le curseur d'un mot à droite.
- **Alt + Left** : Déplace le curseur d'un mot à gauche en conservant les mots en entier.
- **Alt + Right** : Déplace le curseur d'un mot à droite en conservant les mots en entier.

- **Ctrl + U** : Efface tout ce qui se trouve à gauche du curseur.
- **Ctrl + K** : Efface tout ce qui se trouve à droite du curseur.
- **Ctrl + W** : Efface le dernier mot à gauche du curseur.
- **Ctrl + Y** : Colle le dernier élément coupé ou copié.

Ces raccourcis peuvent varier en fonction de la distribution Linux utilisée et du shell utilisé.

From:

<https://www.abonnel.fr/> - **notes informatique & technologie**

Permanent link:

<https://www.abonnel.fr/informatique/linux/commandes/la-ligne-de-commande>

Last update: **2023/08/15 06:49**

