

Commande utiles sous Linux

Demander à la machine de marquer un texte :

[echo "Hello World"]

Connaître quel utilisateur est connecté sur la machine :

[whoami]

Interagir avec les fichiers

Lister les fichiers présents dans le dossier actuel :

[ls]

Changer de dossier :

[cd nom_du_fichier]

Revenir au dossier précédent :

[cd ..]

Lire un fichier :

[cat exemple.txt]

Editer un texte :

[nano exemple.txt]

Il existe également VIM qui a plus de fonctionnalités que "nano".

Connaître le chemin vers le dossier actuel :

[pwd]

Chercher des fichiers :

La commande find :

On peut utiliser find pour trouver un fichier par son nom, son type, sa dernière modification, etc...

par contre avec le nom :

[find -name Nom_du_fichier]

Il existe également la commande "grep" pour pouvoir chercher des valeurs spécifiques dans le contenu d'un fichier :

[grep "exemple" Fichier]

Par exemple si on utilise [grep "gmail" fichier] le terminal va afficher toutes les lignes avec "gmail" dedans.

Savoir utiliser des logiciels précis :

La commande : `-help`

[`ls -help`]

Permet de logiciel la page aide d'un logiciel (ici "`ls`")

Commande `man` (uniquement sur Ubuntu)

[`man ls`]

Permet d'ouvrir le manuel du logiciel (ici "`ls`", également)

Les opérateurs sur Linux

L'opérateur `&`

Vous pouvez exécuter des commandes en arrière plan avec "`&`"

`apt update & apt upgrade`

Lance un `apt update` et un `apt upgrade` en même temps

L'opérateur `&&` et `;`

On peut également lancer deux commandes l'une après l'autre en une seule ligne avec "`&&`".

`apt update && apt upgrade`

`apt update ; apt upgrade`

L'opérateur `>`

L'opérateur "`>`" fait la redirection d'une commande quelque part, par exemple avec la commande `echo` :

`echo test > score`

Ici la commande "`echo`" va s'exécuter dans le fichier `score`

L'opérateur `>>`

Il fait également la redirection d'une commande quelque part, sauf que celui ci n'efface pas le contenu précédent.

`echo test2 >> score`

Ici la commande "`echo test2`" va se rajouter à la suite de "`echo test`" dans le fichier `score`

Les interactions avec les fichiers système

Créer un fichier :

[`touch exemple`]

Créer un dossier :

[`mkdir exemple`]

Supprimer un fichier/dossier :

Pour supprimer un fichier on utilise la commande [rm] mais [rm -r] pour les dossiers.

```
[rm exemple.txt]
```

Copier + renommer un fichier

la commande [cp] permet de copier et de renommer le fichier copié lors de l'opération :

```
[cp score nouveau_score]
```

Ici nous avons copié le fichier "score" et l'avons renommé en "nouveau_score"

Déplacer ou renommer un fichier

La commande [mv] est utilisé pour les deux

```
[mv score Dossier/]
```

(Déplace le fichier score dans Dossier/)

```
[mv score nouveau_score]
```

(renomme le fichier "score" en "nouveau_score")

Connaître les propriétés d'un fichier

Pour savoir si notre fichier est un fichier texte, une vidéo ou autre, nous allons utiliser la commande "file".

```
[file fichier]
```

Changer d'utilisateur

La commande "su" est utilisée à but de changer d'utilisateur depuis le terminal.

```
[su NomUtilisateur2]
```

```
[su -l NomUtilisateur2]
```

note : selon la commande utilisée, nous ne serons pas amenés aux mêmes fichiers

Télécharger des fichiers

Pour cela nous allons utiliser la commande "wget"

```
[wget https://www.LienExemple.com/download]
```

Ici nous téléchargeons ce qui se trouve à "download" (si c'est un lien de téléchargement direct)