

# The UNIX Command Line: A Beginner's Guide

## CLI (Command Line Interface) vs. GUI (Graphical User Interface)

	CLI	GUI
Utilisation intuitive	Non	Oui
Contrôle fin des actions	Oui	Généralement non
Utilisation de ressources supplémentaires	Non	Oui
Automatisation aisée	Oui	Non

L'interpréteur de commande UNIX est dénommé "[Shell](#)." Plusieurs co-existent. On utilisera ici `bash` "[Bourne-Again shell](#)"

# UNIX CLI Part 1/4: Les commandes sans danger

## Exploration: La commande `ls`

La commande `ls` permet de lister le contenu d'un repertoire. On ne peut rien casser avec, n'hésitez pas à vous en servir.

```
> ls  
dungeon.sh  
Documents  
(...)
```

On peut ajuster la command afin d'avoir plus d'information.

```
> ls -lh
-rw-r--r--  1 chabotv assim 2,4K sept.   8  2025 dungeon.sh
drwxr-xr-x  2 chabotv assim 4,0K nov.   25  2024 Documents
```

- `l` : liste les fichiers dans un format long (montrant des détails comme les permissions)
- `h` : affiche le poids des fichiers dans un format humainement compréhensible

Avez vous une idée de la signification de `-rw-r--r--` ?

## Auto-complétion (Sans danger)

Taper chaque lettre peut être ennuyeux et surtout source d'erreurs. La plupart des terminaux permettent d'utiliser l'auto-complétion via la touche **Tabulation** (dénnotée `\t`).

Exemple : Arriver au répertoire du cours : `/home/newton/ienm2021/chabotv/COURS_CS`

```
> ls /ho\t
newton
(...)
> ls /home/ne\t
> ls /home/newton/i\t
ienm2021
ienm2022
(...)
> ls /home/newton/ienm2021/cha\t
> ls /home/newton/ienm2021/chabotv/COURS_CS\t
dungeon.sh
```

## Caractère générique

Pour accélérer les recherches vous pouvez utiliser des caractères génériques.

Utilisez `*` pour n'importe quel groupe de caractères et `?` pour un seul caractère.

```
> ls *.pdf # Lists all files ending with .pdf
> ls */*.pdf # Lists all first-level files ending with .pdf
> ls */**/*.pdf # Lists all second-level files ending with .pdf
```

Ce que vous avez appris avec `ls` (complétion, caractère générique, options) peut être appliquées à la plupart des commandes unix.

Pour avoir une idée des options d'une commande faites `man MA_COMMANDE`.

Ex : `man ls`

Quelle est l'option permettant de trier les fichiers en commençant par les plus volumineux ?

# Navigation

Pour changer de répertoire, il faut utiliser la commande `cd` (change directory).

`pwd` (present working directory) permet de savoir où vous êtes.

Si jamais vous souhaitez revenir dans votre répertoire personnel (home directory) vous pouvez faire un simple `cd`.

```
> cd  
> pwd  
/home/gmap/mrpa/chabotv
```

Si vous souhaitez revenir au répertoire précédent faites `cd -`.



## Examiner des fichiers textes (`cat`, `head`, `tail`)

Ces commandes permettent d'examiner le contenu d'un fichier texte.

`cat` affiche l'ensemble du contenu, que celui-ci soit de l'ASCII ou du binaire.

A titre d'exemple, vous pouvez utiliser le dictionnaire de mots `/usr/share/dict/words` or `/usr/dict/words`.

```
> cat /usr/share/dict/words  
A  
a  
aa  
aal  
aalii  
aam  
Aani  
(...)
```

Vous pouvez utiliser les commandes `head` ou `tail` pour voir le début ou la fin du fichier. Par défaut, cela affiche uniquement 10 lignes, mais via des options on peut adapter ce nombre de lignes.

# Rechercher les occurrences d'une chaîne de caractères

Pour effectuer des recherches au sein de fichiers `ascii`, vous pouvez utiliser la commande `grep`.

**Exemple** : Trouver l'ensemble des mots contenant "cumul" dans le dictionnaire

```
grep cumul /usr/share/dict/words
```

Combien en existe-t-il ?

# UNIX CLI Part 2/4: Commandes de gestion de fichiers

## Touch a File: `touch`

La commande `touch` permet de modifier l'heure d'accès et de modification d'un fichier sans changer le fichier en lui-même. C'est sans doute la plus petite modification que l'on peut faire.

Si le fichier n'existe pas, `touch` crée un fichier vide pour vous.

```
> ls -l tmp.txt
-rw-r--r-- 1 chabotv mrpa 121 Sep 19 14:31 tmp.txt
> touch tmp.txt
> ls -l tmp.txt
-rw-r--r-- 1 chabotv mrpa 121 Sep 19 14:32 tmp.txt
> touch tmp2.txt
> ls -lrt tmp*
-rw-r--r-- 1 chabotv mrpa 0 Sep 19 14:32 tmp2.txt
-rw-r--r-- 1 chabotv mrpa 121 Sep 19 14:32 tmp.txt
```

## Créer un repertoire : `mkdir`

```
> ls      # When there's nothing to report, nothing happens
> mkdir TEST
> ls
TEST
> mkdir TEST
mkdir: TEST: File exists
> cd TEST
> ls
```

## Copier un fichier/repertoire: `cp`

La commande `cp` permet de dupliquer des fichiers. On peut utiliser l'option `-i` pour des raisons de sécurité : cela demandera confirmation avant d'écraser le fichier.

```
> ls tmp*
tmp.txt          tmp2.txt
> cp tmp.txt tmp_copy.txt
> ls tmp*
tmp.txt          tmp2.txt          tmp_copy.txt
> cp tmp2.txt tmp_copy.txt
> ls tmp*
tmp.txt          tmp2.txt          tmp_copy.txt
> cp -i tmp.txt tmp_copy.txt
overwrite tmp_copy.txt? (y/n [n]) y
tmp.txt          tmp2.txt          tmp_copy.txt
```

Si vous avez besoin de copier un répertoire entier, il faut explicitement mettre l'option

`-r` :

```
> mkdir TMP # Create a dummy void folder
> cp TMP TMP3
cp: TMP is a directory (not copied).
> cp -r TMP TMP3
> ls TMP3
test1.txt
```



## Déplacer ou renommer des Fichiers/Répertoires: `mv`

La commande `mv` permet de déplacer des fichiers et des dossiers.

```
> mkdir TMP # Create a dummy void folder
> mkdir TMP2 # Create another dummy folder
> touch TMP/test1.txt # Create a dummy file
> mv TMP/test1.txt TMP2/test_02.txt
> tree
```

```
.
├── TMP
└── TMP2
    └── test_02.txt
```

3 directories, 1 file

## Supprimer des fichiers : `rm`

Une des commandes les plus puissantes et dangereuse : `rm` .

Dans sa forme la plus simple on supprime un fichier :

```
> tree
```

```
.
├── TMP
└── TMP2
    └── test2.txt
```

```
3 directories, 1 file
```

```
> rm TMP2/test2.txt
```

```
> tree
```

```
.
├── TMP
└── TMP2
```

```
3 directories, 0 files
```

## Le cas spécifique de `rm -rf *`

Par défaut `rm`, ne supprime pas les répertoires mais seulement les fichiers.

Il faut utiliser l'option `-r` (recursive) pour supprimer un répertoire entier + contenu.

```
> rm -r TMP2
```

```
rm -rf *
```

L'option `-f` signifie qu'on force la commande à s'exécuter. `-rf` permet donc de supprimer l'ensemble des éléments en dessous du point cible, en enlevant l'ensemble des barrières qu'il pourrait exister.

Le caractère générique `*` signifiant "tout ce qu'il y a dans ce dossier".

Par conséquent les conséquences d'un mauvais usage de cette commande peuvent être important.

# UNIX CLI Part 3/4: Edition de fichiers & Script UNIX

## Editeur de texte

Lorsque vous opérez en local, vous pouvez utiliser des éditeurs de texte graphique pour éditer vos scripts.

A titre d'exemple, vous pouvez utiliser `vscode` ou `gedit` pour éditer vos scripts.

```
> code hello.sh
```

ou

```
> gedit hello.sh
```

Dans le fichier ouvert (en mode graphique) vous pouvez créer un simple script affichant "Hello, World!".

Pour cela, vous allez utiliser la commande UNIX `echo` permettant d'écrire dans le shell :

```
echo "Hello, World!"
```

Une fois le fichier sauvegardé ( `ctrl +s` ), vérifier que le fichier est présent puis vous pouvez l'exécuter via

```
bash hello.sh
```

# UNIX CLI part 4/4 : Tester vos base

# Quelques message d'erreurs à interpréter

```
chabotv@pxassim2:~/Code/Outils-CS$ ls MySuperFile
ls: impossible d'accéder à 'MySuperFile': Aucun fichier ou dossier de ce nom
```

```
chabotv@pxassim2:~/Code/Outils-CS$ cd MyFolder
bash: cd: MyFolder: Aucun fichier ou dossier de ce nom
chabotv@pxassim2:~/Code/Outils-CS$
```

```
chabotv@pxassim2:~/Code/Outils-CS$ cp /home/newton/ienm2021/chabotv/COURS/dunge ma_copie_locale
cp: impossible d'évaluer '/home/newton/ienm2021/chabotv/COURS/dunge': Aucun fichier ou dossier de ce nom
chabotv@pxassim2:~/Code/Outils-CS$
```

```
chabotv@pxassim2:~/Code/Outils-CS$ ls -l test.txt
-r--r--r-- 1 chabotv assim 0 sept. 26 15:37 test.txt
chabotv@pxassim2:~/Code/Outils-CS$ echo "Mon info à mettre dans un fichier" >> test.txt
bash: test.txt: Permission non accordée
```



# Un peu de manipulation

Pour cet exercice, vous allez devoir exécuter le script `dungeon.sh`.

Ce script `dungeon.sh` est disponible dans le dossier de votre professeur :

```
/home/newton/ienm2021/chabotv/COURS_CS
```

Vous pouvez bien entendu l'ouvrir pour voir ce qui est fait dedans.

# Démarrons votre aventure UNIX ! Ceci est votre première quête :

1. Créer un dossier `Cours_CS` et un sous dossier `Test_unix` . Aller à l'intérieur.
2. Copier le fichier `dungeon.sh` depuis le dossier de votre professeur  
( `/home/newton/ienm2021/chabotv/COURS_CS` ) dans le dossier `Test_unix`
3. Exécutez le script
4. Si besoin, changer les droits d'accès (via `chmod` command) pour rendre le fichier soit executable
5. Comptez combien de fichiers `goblints` il y a dans votre dongeon
6. Trouver l'ensemble des occurences de `prince` et `princess` dans vos fichiers.
7. Supprimer le dossier du dongeon (sans supprimer le script `dungeon.sh` !
8. Recommencez avec un nouveau dongeon.

# Les réponses

1. Créer un dossier `Cours_CS` et un sous dossier `Test_unix` . Aller à l'intérieur.

```
cd ~  
mkdir Cours_CS  
cd Cours_CS  
mkdir Test_unix  
cd Test_unix  
pwd
```

2. Copier le fichier `dungeon.sh` depuis le dossier de votre professeur  
( `/home/newton/ienm2021/chabotv/COURS_CS` ) dans le dossier `Test_unix`

```
cp /home/newton/ienm2021/chabotv/COURS_CS/dungeon.sh ~/Cours_CS/Test_unix/dungeon.sh
```

5.

Une première option est de "compter" à la main (tapper simplement la commande `tree` dans le dossier.)

Vous pouvez aussi utiliser un pipe ( `|` ) pour enchaîner avec une autre commande et ne conserver que les lignes contenant `goblins.txt` : `tree|grep goblins.txt`

Une autre option est d'utiliser la commande `find` :

```
find . -name goblins.txt
```

Vous pouvez aussi utiliser la commande `wc` pour faire compter le nombre de ligne dans la réponse au shell à votre place.

Exemple : `tree|grep goblins.txt|wc -l`

6. Pour effectuer une recherche dans l'ensemble des fichiers du dossier Dungeon vous pouvez utiliser `grep`

```
grep -r prince Dungeon/*
```

## Quelques compléments

Il est possible (et même encouragé) de créer des alias pour des courtes commandes récurrentes.

Pour ce faire, il faut éditer le fichier `~/.bashrc` en ajoutant l'alias souhaité.

Ex : On peut par exemple ajouter le repertoire dans lequel les supports de cours seront disponible.

```
alias cdprof='cd /home/newton/ienm2021/chabotv/COURS_CS'
```