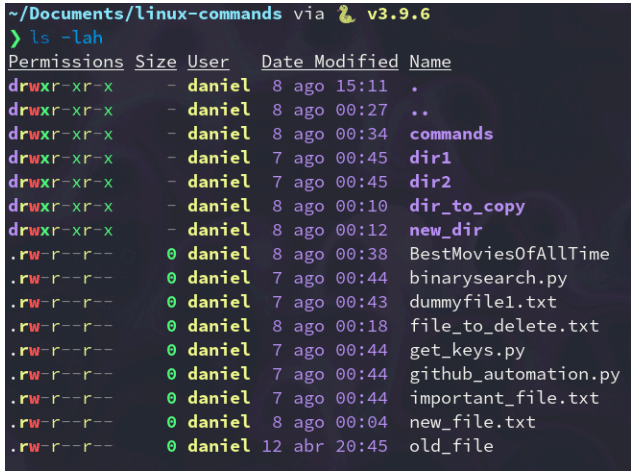
Sorbonne Nouvelle
université des cultures

Cultures numériques avancées

Récapitulatif : Bash et commandes essentielles

Ljudmila PETKOVIC

ljudmila.petkovic@sorbonne-nouvelle.fr



Source : Diaz, 2023.

Cultures numériques avancées (L2HN001)
Licence *Sciences du langage*, mineure « Humanités numériques »
Paris, le 23 février 2024, année 2023-2024

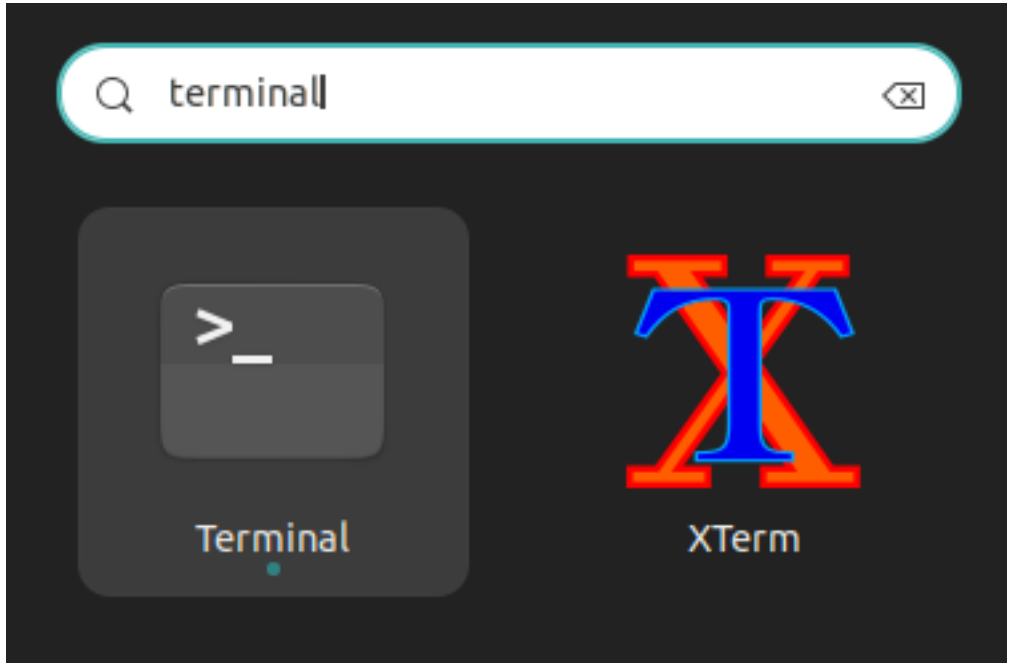
Diapositives adaptées de Simon Gabay, Simone Rebora et Élisabeth Brunet et Gaël Thomas.

Utilisation de la ligne de commande

Ouvrir le terminal

Pour **Linux** :

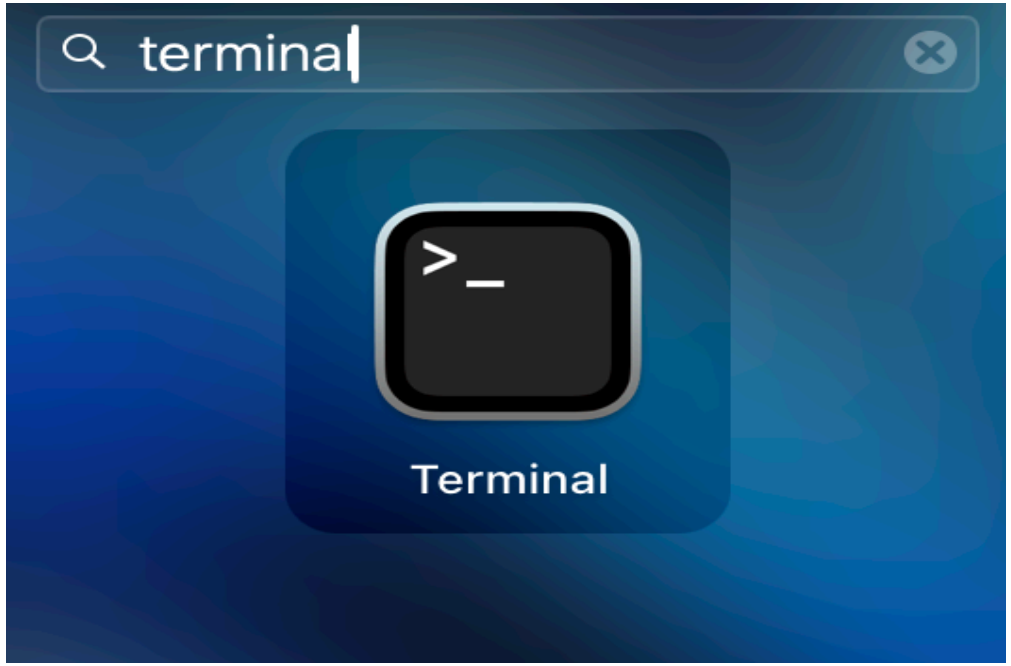
- appuyer simultanément sur les touches `Ctrl` + `Alt` + `T`
- ouvrez la barre de recherche, tapez `terminal` et appuyez sur Entrée



Ouvrir le terminal

Pour **Mac** :

- ouvrez la barre de recherche, tapez `terminal` et appuyez sur Entrée
- ou cliquer sur `Terminal.app` dans `Applications` > `Utilitaires`



Ouvrir le terminal

Pour **Windows** :

⚠️ Windows dispose de deux *shells* – `Command Prompt` et `PowerShell` qui ne fonctionnent pas de la même manière que ceux sur Linux ou Mac.

Ce n'est pas possible d'exécuter toutes les commandes Bash sur Windows, sauf si l'on n'utilise pas certaines commandes équivalentes, adaptées à Windows.

Pour contourner ce problème, les utilisateurs de Windows peuvent télécharger Git Bash (émulateur de terminal Bash sur Windows).

Git Bash pour Windows

- Téléchargez Git Bash depuis le site officiel de Git (lien)
- Suivez les étapes d'installation indiquées dans ce tutoriel + tutoriel vidéo
    - Pour l'étape 4., choisir l'option `Use Vim (the ubiquitous text editor) as Git's default editor`
- Suite à l'installation du Git Bash, assurez-vous de son bon fonctionnement en l'ouvrant et en tapant :

```
git --version
```

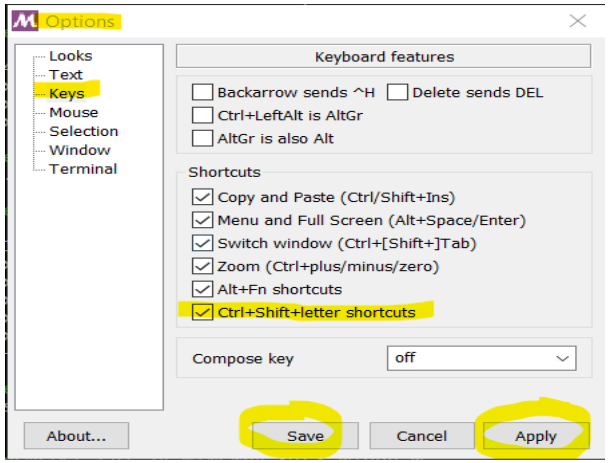
Raccourcis pour copier-coller (facultatif)

À priori, les raccourcis pour *copier* et coller sont les suivants :

- copier : `Ctrl + Shift + C`
- coller : `Ctrl + Insert` ou `Shift + Insert`

Pour activer le raccourci `Ctrl + Shift + V` pour coller :

1. Clic droit > `Options...`
2. Dans l'onglet Keys, cochez `Ctrl + Shift + letter shortcuts`



Les commandes abordées jusqu'à présent

Pour chaque commande, vous trouverez :

- sa description
- ses options courantes (si elles nous sont pertinentes)
- quelques exemples de son utilisation

L'objectif est de mémoriser ces commandes basiques et de comprendre la logique de leurs fonctionnements.

Pour tous les autres détails, référez-vous aux diapositives des séances 1-5.

`pwd`

- angl. ***p**rint **w**orking **d**irectory*
- affiche le répertoire courant
- montre où nous sommes dans le système des fichiers et des répertoires

```
pwd
```

`ls`

- angl. ***list***
- liste le contenu du répertoire courant :

```
ls
```

- liste le contenu d'un autre répertoire donné en paramètre :

```
ls mon_dossier
```

- options courantes : `-a` (pour afficher les fichiers cachés dont le nom commence par un point `.`, p. ex. `.fichier_cache.txt`)

```
ls -a
```

`cd`

- angl. ***c**hange **d**irectory*
- change le répertoire courant

```
cd # se placer dans le répertoire personnel ('home')
```

`cd` + chemin absolu / relatif

- chemin absolu (chemin à partir du répertoire-racine `/`)

```
cd /home/Utilisateur/Documents/
```

- chemin relatif

```
cd .. # remonter d'un répertoire (aller à son 'parent')
```

```
cd ../.. # remonter de deux dossiers (aller à son 'grand-parent')
```

```
cd Documents # accéder au répertoire 'Documents' depuis le répertoire actuel
```

```
cd Utilisateur/Documents # accéder au répertoire 'Documents' via le répertoire 'Utilisateur'
```

`cp`

- **c**o**p**ie un fichier

```
cp fichier_original.txt fichier_copie.txt
```

- options courantes : `-r`
  - copie un répertoire, si l'option `-r` est utilisée)

```
cp -r repertoire_original repertoire_copie
```

`mv`

- angl. ***move***
- **déplace** un fichier dans un répertoire

```
mv fichier.txt Documents
```

- déplace un répertoire dans un répertoire

```
mv mon_dossier Documents
```

- **renomme** un fichier

```
mv fichier.txt fichier_renomme.txt
```

- renomme un répertoire

```
mv repertoire repertoire_renomme
```

`rm`

- angl. ***rem**ove*
- supprime un fichier

```
rm fichier_a_supprimer.txt
```

- options courantes : `-r`
  - supprime un répertoire, si l'option `-r` est utilisée

```
rm -r repertoire_a_supprimer
```

`touch`

- crée un fichier

```
touch mon_script.sh
```

nano

- éditeur de texte en ligne de commande (*suivez les instructions de l'éditeur en bas de l'écran.*)

```
nano mon_script.sh
```

- ajouter un appel de script (*shebang*) suivi par votre script (p. ex. une commande)

```
#!/bin/bash
echo "Coucou"
```

Pour quitter : `Ctrl + X` → *Sauver l'espace modifié ?* `O` (ou `Y`, si votre Terminal est configuré en anglais) → *Nom du fichier à écrire :* `mon_script.sh` (valider par Entrée)

`echo`

- affiche un message (ou les résultats d'autres commandes), ici une **chaîne de caractères**

```
echo "Coucou René"
```

  - une chaîne de caractère peut être stockée dans une **variable** (p. ex. `nom`)

```
nom="Michel"
```

  - quand on appelle une variable, on ajoute le signe du dollar `$` avant son nom

```
echo "Coucou $nom"
```

`grep`

- angl. ***g**lobal **r**egular **e**xpression **p**rint*
- recherche une chaîne de caractères dans des fichiers
  - recherche simple

```
grep "Ceci" fichier_test.txt
```

  - recherche avancée
    - options courantes : `-E` (regex), `-o` , `-i`
    - recherche en utilisant les expressions régulières (**regex**), p. ex. « trouver tous les mots commençant par la lettre « C » en majuscule »

```
grep -E "C\w+" fichier_test.txt
```

`bash` ou `sh`

- exécute un script *shell* (avec l'extension `.sh`)

```
bash mon_script.sh
```

```
sh mon_script.sh
```

- la commande `bash` peut être enlevée :

```
./mon_script.sh
```

Si vous obtenez un message d'erreur : `bash: ./mon_script.sh: Permission non accordée`, il faut donner au fichier la permission d'exécution (le rendre exécutable) en tapant `chmod +x mon_script.sh`, et ensuite relancer la commande ci-dessus.

`mkdir`

- angl. ***make directory***
- crée un répertoire

```
mkdir repertoire
```

`cat`

- angl. *con**cat**enate*
- lire le contenu du fichier

```
cat mon_fichier.txt
```

Redirection du contenu : `>` et `>>`

- pour **rediriger** le contenu du fichier `source.txt` vers le fichier `destination.txt` , utiliser l'option `>` :

> Une redirection avec `>` sur un fichier qui existe déjà efface le contenu du fichier avant d'y placer le résultat de la commande.

```
cat source.txt > destination.txt
```

- pour **ajouter** le contenu dans le fichier de destination, utiliser l'option `>>`

```
cat source.txt >> destination.txt
```

Avantages de l'utilisation de la ligne de commande

- **Flexibilité** : combiner les commandes et obtenir une palette pratiquement infinie de fonctions nouvelles
- **Fiabilité** : tendance à s'exécuter de la même manière sur différents SE (« couteau suisse »)
- **Rapidité** : automatisation des tâches à grande échelle (p. ex. renommer un ensemble des fichiers d'un seul coup)
- **Expérience** : communiquer avec l'ordinateur plus directement qu'avec les interfaces graphiques, en apprenant ainsi énormément sur son fonctionnement interne
- **Économisation des ressources** : utilise les ressources de l'ordinateur beaucoup plus parcimonieusement que les programmes graphiques

Source : Floss Manuals

Références

- **Brunet, É. & Thomas, G.** (*s. d.*). « Le shell bash » [*diapositives*]. Télécom SudParis. http://www-inf.telecom-sudparis.eu/cours/CSC3102/Supports/ci1-bash/ci-bash.pptx.pdf
- **Combeau, M.** (2022). « La différence entre le terminal, la console et le shell ». https://www.codequoi.com/difference-entre-terminal-console-et-shell/
- **Champin, P.-A.** (2020). Séance 5 : Automatisation et scripts. Département Informatique (IUT Lyon 1). https://perso.liris.cnrs.fr/pierre-antoine.champin/enseignement/linux/s5.html
- **Rebora, S.** (2022). « Connaître son propre ordinateur » [*dépôt GitHub*]. EnExDi2022. https://github.com/ABC-DH/EnExDi2022/tree/main/materials/1_KnowYourComputer/slides
- **Gabay, S.** (2022). « Les lignes de commandes et Bash » [*dépôt GitHub*]. Université de Genève, Chaire des humanités numériques, Faculté des Lettres. https://github.com/gabays/Fondamentaux/blob/main/Lignes_de_commandes/DistRead_1_2.pdf
- **Diaz, D.** (2023). « Les 40 commandes Linux les plus utilisées que vous devez connaître ». Kinsta. https://kinsta.com/fr/blog/commandes-linux/
- **Xyoos** (*s.d.*). « Interface graphique ». https://cours-informatique-gratuit.fr/dictionnaire/interface-graphique/