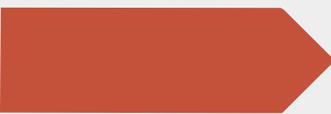




- Fin du CM2 : quelques mots sur MacOS et Windows
- mini-recap des TD 1 et 2
- les utilisateurs en UNIX
- Bonus : tips généraux

(pas de wooclap, désolée)



# Généralités sur UNIX et ses dérivés

## Dérivé n°2 : MacOS



# Mac OS (renommé OS X puis MacOS)

- Le système d'exploitation des ordinateurs Macintosh d'Apple, intimement **lié au matériel**.
- **Partiellement propriétaire** : certaines parties de FreeBSD (une saveur d'Unix) ont été réutilisées par Apple dans ce système d'exploitation.

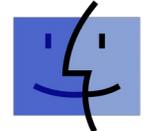


1984 - 1988



Macintosh

1988 - 1995



Mac OS

1995 - 2002



Mac OS X

2001 - 2003



Mac OS X

2003 - 2012

OS X

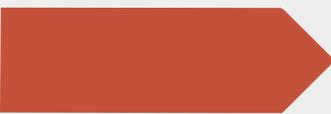
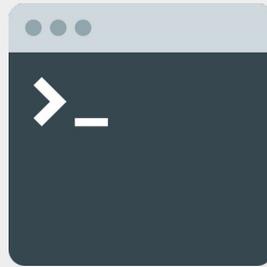
2012 - 2016

macOS

2016 - 2017

macOS

2017 - now



# Windows : la vache à lait de Microsoft

# Windows

- Propriétaire, développé par Microsoft

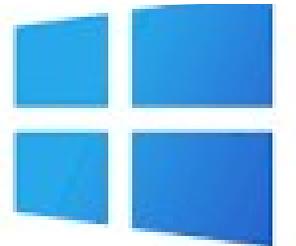
Historique :

- 1981 : **MS-DOS** « Microsoft Disk Operating System » :  
Système **monotâche, mono-utilisateur**, et **pas de mode protégé**.  
Interface en ligne de commande (langage CMD).
- 1985 : **Windows v1** = interface graphique pour MS-DOS
- 1993 : Nouveau kernel **Windows NT** (« New technology »).  
Système d'exploitation **multitâche préemptif, multi-utilisateur, multiprocesseur**.  
Langage : **powershell**.

Sous-système Windows pour Linux (**WSL**) : plus besoin de dual boot.

Écran de démarrage MS-DOS 6.22.

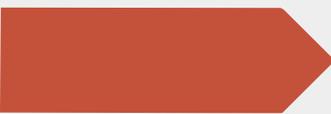
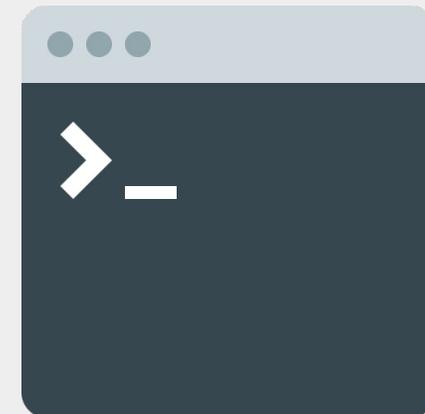
```
Démarrage de MS-DOS...  
  
Vérification de la mémoire étendue par HIMEM...  
Vérification terminée.  
  
C:\>C:\DOS\SMARTDRV.EXE /X  
  
Fonction MODE PREPARE pour la page de codes terminée  
  
Fonction MODE SELECT pour la page de codes terminée  
C:\>_
```



# Le quasi-monopole de Microsoft Windows

- « vente-liée » : Windows est (quasi) systématiquement installé quand on achète un ordinateur
- « racketiciel » (AFUL : Association francophone des utilisateurs de logiciels libres)





# Linux :

## Le ~~petit~~ grand frère de UNIX



# Ce qu'on a vu concernant Linux

- Le **noyau** Linux a été initié par **Linus Torvald**.
- Les OS Linux (**distribution**) sont des OS **UNIX-like**. Ils partagent donc énormément de concepts.
- On passe par la ligne de commande, non pas qu'Unix/Linux n'ait pas d'interface graphique, mais car pour certaines tâches, **l'utilisation de la ligne de commande s'avère bien plus pratique et plus puissante que la souris**.
- Les fonctionnalités peuvent varier, notamment en fonction du Shell utilisé.
- En TD, vous avez appris à interagir avec Linux (distribution Ubuntu) via **la ligne de commandes** (shell Bash). Vous vous êtes connecté, avez navigué dans votre espace de travail, joué avec les fichiers, utilisé un éditeur de texte « archaïque » mais puissant : **vi(m)**.  
Commandes : **cd, ls, cat, grep, ...**



## « ça marche pô »

- Vérifiez qu'il n'y a pas une faute de frappe (/!\ les commandes sont sensibles à la case).
- Vérifiez que vous avez les bons arguments, le bon nombre
- Regardez les indications du terminal (le terminal est votre ami)
- Préparez-vous à poser votre question à son.a voisin.e / le.a prof / internet :
  - Qu'est ce que je voulais faire ?
  - Quelles sont mes entrées ?
  - Quelles sont les sorties attendues ?



# Objectif pour le cours

À retenir :

- Le nom des commandes qu'on vous présente
- À quoi elles servent, indépendamment de leurs arguments (= le synopsis du **man**, avec vos propres mots)
- Savoir à quoi servent les principaux arguments (ceux qu'on vous présente).

Ce qui n'est pas demandé :

- Réciter le manuel

**Astuces** : se concentrer en cours et en TD (apprentissage par exposition, répétition) et se faire des petites fiches (moyens mnémotechniques).

# Le prompt

- C'est ce qui s'affiche par défaut dans votre terminal / console une fois connecté.e.
- En général : « username@machinename \$ »
- Configurable, via la variable **PS1** (Prompt String)

---

```
nanis@jammy[~] $ echo $PS1
\[ \e[34m\]nanis\[ \e[m\]@\[ \e[44m\]jammy\[ \e[m\][\w] $
nanis@jammy[~] $ PS1=toto
toto
toto
toto
toto
toto
toto
totoecho $PS1
toto
totoPS1="`parse_git_branch`\[ \e[34m\]nanis\[ \e[m\]@\[ \e[44m\]jammy\[ \e[m\][\w] $ "
nanis@jammy[~] $ □
```

# Le prompt – exemples post configuration

```
(base) (jammy-base r5828) nanis@C302L-G24P07:~$ ls
baseconfig      Desktop      miniconda3  ownCloud
baseconfig-gitlab Documents    Modèles     Public
bluetooth-pods.sh Images       Musique      signal-desktop-keyring.gpg
(base) (jammy-base r5828) nanis@C302L-G24P07:~$ pwd
/home/nanis
```

```
andy > dell > ~ > cd Documents/Inbox/
andy > dell > ~/Documents/Inbox > :)
```

```
root@C302L-G24P07:/home/vaginay241 2024-09-18 19:50 (0) :) (jammy-base r5853)
root@C302L-G24P07:/home/vaginay241 2024-09-18 19:50 (0) :) (jammy-base r5853)
toto
bash: toto : commande introuvable
root@C302L-G24P07:/home/vaginay241 2024-09-18 19:50 (0) :( 127 (jammy-base r5853)
```

```
lin@Demeter ~/Software/munin-pihole-plugins on master 2.50 1 at 19:20:29
```

```
nanis@jammy[~] $ ls
AV_sugg
baseconfig@
baseconfig-gitlab/
bluetooth-pods.sh@
nanis@jammy[~] $
```

```
21:22:57 79% chris@cruachan ~
$ dossh
chris@192.168.0.18's password:
Welcome to Arch Linux ARM

Website: http://archlinuxarm.org
Forum: http://archlinuxarm.org/forum
IRC: #archlinux-arm on irc.Freenode.net
Last login: Tue Sep 18 18:06:59 2018 from 192.168.0.1
21:23:05 chris@alarmpi 192.168.0.18 ~
$
```

# Syntaxe générale d'une commande

- Une commande s'appelle directement par son **nom**
- On peut spécifier un ou plusieurs **arguments** à une commande :
  - directement une valeur (**paramètre de commande**),
  - un **nom d'option** (« *--option* » ou « *-o* »), suivi si besoin d'une valeur (**paramètre d'option**).

- Exemples :

**ls** # juste une commande, son nom est « ls ».

**ls .** # 1 argument qui est un paramètre de commande

**ls -a .** # 2 arguments : une option et un paramètre

**ls -l** # juste une option

**ls -la** # 1 argument correspondant à 2 options combinées (oui, des fois c'est possible)

**ls --width 5 dossier/** # 3 args (une option, un param d'option, un param de commande)

**ls -w 5 dossier/** # idem, mais option en version courte

# Cas d'erreurs d'une commande

- La commande n'**existe pas**. ATTENTION à la casse et aux espaces.  

```
nanis@jammy[~] $ jfidofhisdof  
bash: jfidofhisdof : commande introuvable
```
- Vous n'avez **pas le droit** d'exécuter cette commande.  

```
nanis@jammy[~] $ cd /root/home  
bash: cd: /root/home: Permission non accordée
```
- Les **arguments** (options et/ou paramètres) de la commande sont **erronées**.  

```
nanis@jammy[~] $ mv file  
mv: opérande de fichier cible manquant après 'file'  
Saisissez « mv --help » pour plus d'informations.  
nanis@jammy[~] $ mv f1 f2  
mv: impossible d'évaluer 'f1': Aucun fichier ou dossier de ce nom
```

Que faire ? **Lire** le retour de la commande, **chercher** des explications dans l'aide (**man**), **demander**.



# L'aide

**man** : le manuel de référence

Syntaxe de base : **man** *[option]* *[[section] page]*

Arguments :

- **option** : Facultatif
- **section** : Facultatif – Précise le n° de la section du manuel
- **page** : Page dont on souhaite consulter le manuel

- La documentation d'une commande indique sa **syntaxe générale**, ainsi que le **détail des arguments possible** : **commande** *[option ...]* *[paramètres ...]*
- Dans la syntaxe générale, si un argument est indiqué entre [ ], alors il est facultatif.



# Commandes utiles du terminal : **man**

- Affichage toutes les pages **passwd** (de toutes les sections) du manuel:

```
prompt> man -a passwd
```

- Affichage de la page **passwd** dans la section 5 du manuel :

```
prompt> man 5 passwd
```

- Trouver toutes les rubriques contenant un mot clé donné :

```
prompt> man -k mot-clé-donné
```

- Description succincte d'une commande :

```
prompt> man -f passwd
```

- Plus d'info sur man :

```
prompt> man man
```

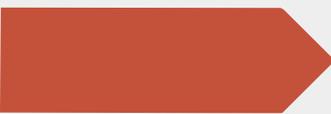


# Les sections d'aide dans le man

Section	Type de commandes
1	commandes et applications utilisateur ( <b>ls</b> , <b>cd</b> , <b>grep</b> , ..., <b>passwd</b> )
2	appels système, codes erreurs noyau ( <b>open</b> , <b>read</b> , <b>write</b> )
3	fonctions des bibliothèques (bib standard C : <b>printf</b> , <b>malloc</b> )
4	fichiers spéciaux : pilotes de périphériques et protocoles réseau ( <b>/dev/null</b> )
5	formats de fichiers standard et autre conventions ( <b>/etc/passwd</b> )
6	jeux
7	divers fichiers et documents
8	commandes d'administration système ( <b>iptables</b> , <b>mount</b> , <b>apt-get</b> )
9	divers specs noyau et interfaces (non standard)

# Les notions et commandes qu'on va (re)voir

- CM3 (aujourd'hui) ← • Utilisateurs (**CM4** et **TD3** et **TD4**)
- CM4 ← • Fichiers et arborescence (aussi vu en **TD1** – les bases, **TD2** – édition avec vi, **TD3** – navigation dans l'arborescence, **TD4** – gestion des droits)
- CM5 ← • Processus (**TD5** – les bases, **TD6** – communications)
- CM6 ← • Infos système et mise à jours
- CM7 ← • Programmation shell (**TD7** et **TD8**)



# Linux : les utilisateurs



# Vocabulaire

Linux est un  **système multiutilisateur**

Toute entité (personne physique ou programme particulier) devant interagir avec un système Linux est authentifiée sur cet ordinateur par un  **utilisateur.ice** .

Chaque utilisateur.ice est identifié par un nom unique ( **login** ) et un numéro unique : le  **UID**  – user ID).

Chaque utilisateur.ice fait partie d' **un ou plusieurs groupes**  (par défaut un groupe du même nom).

Chaque groupe est également identifié par un numéro unique : le  **GID**  – group id. Les groupes servent à identifier des utilisateurs qui ont des caractéristiques communes (profs, étudiants, invités) et permettent une gestion efficace.

Il existe un compte utilisateur particulier qui  **dispose de tous les droits**  :  **root**  ou «  **super utilisateur**  », ou «  **compte administrateur**  » (d'UID 0).

# Connexion dans un terminal

**Compte** = nom de connexion + mot de passe.

Login : **Jean**

Password : **\*\*\*\*\***

Bienvenue sur ...

Prompt >

```
jfa@jfa-VirtualBox:~$
```

```
root@jfa-VirtualBox:/home/jfa#
```

« \$ » (ou « % », ou « # » - généralement pour root) est le **prompt** ou l'**invite** de l'**interprète de commande** utilisé (shell).

L'interprète attend que l'utilisateur tape une commande, exécute cette commande, réaffiche la chaîne d'invite, et attend une nouvelle commande, jusqu'à ce qu'on quitte...

**exit** ou **^D (CTRL-D)** en début de ligne ⇒ déconnexion (fin du shell)



# Où suis-je connecté.e ?

**tty** : renvoie le nom du terminal sur lequel on est connecté (en soit c'est juste un nom de fichier ; en UNIX tout est fichier).

Exemple :

```
nanis@jammy[~] $ tty  
/dev/pts/0
```

<https://www.geeksforgeeks.org/tty-command-in-linux-with-examples/>

<https://www.malekal.com/quest-ce-que-tty-comment-utiliser-commande-tty-sur-linux/>



# Qui d'autre est connecté ?

**who** : liste des utilisateurs connectés,

Exemple d'utilisation :

```
nanis@jammy[~] $ who
vaginay241 :1          2024-09-15 12:34 (:1)
vaginay241 tty3       2024-09-18 16:02
```

<https://www.geeksforgeeks.org/who-command-in-linux/>

# Informations sur les utilisateur.ices

`/etc/passwd` est un fichier statégique qui rassemble des **infos sur toutes les entités** ayant un compte sur le système (user physique ou certains programmes spécifiques).

Exemple :

```
root:x:0:0:root:/root:/bin/bash
```

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Une ligne pour chaque entités, 7 champs par ligne (séparés par le caractère « : »)

- **Nom** : login
- **Mot de passe** : soit rempli et crypté (sur 13 caractères) par le programme `passwd`, soit contient « x » avec un mot de passe crypté et déporté dans un autre fichier, accessible par l'administrateur.ice.
- **UID [= User Identification]** : numéro d'identification de l'utilisateur.ice (0 pour root).
- **GID [= Group Identification]** : numéro d'identification du groupe primaire de l'utilisateur.ice.
- **Commentaire** : champ facultatif.
- **Répertoire HOME** : répertoire d'accueil lors de la connexion de l'utilisateur.
- **Commande** : commande lancée au moment de la connexion.



# Informations sur les utilisateur.ices et groupes

Syntaxe : **id** [*user ...*] : Affiche l'UID, le GID et la liste des groupes d'un utilisateur

Paramètres : *user* : Nom de l'utilisateur dont on souhaite afficher les informations.

Si non-spécifié : affiche les informations de l'utilisateur exécutant la commande.

Il est également de spécifier plusieurs utilisateurs (séparés par des espaces).

Syntaxe : **groups** [*user ...*] : Affiche la liste des groupes d'un utilisateur

Paramètre : *user* : Nom de l'utilisateur dont on souhaite afficher les informations.

Si non-spécifié, affiche les informations de l'utilisateur exécutant la commande.

Il est possible de spécifier plusieurs utilisateurs (séparés par des espaces).



# Liste des groupes

L'ensemble des groupes est listé dans le fichier `/etc/group`

Exemple :

```
root:x:0:root
```

```
daemon:x:1:daemon
```

Les champs, dans l'ordre :

- nom du groupe,
- mot de passe du groupe,
- GID,
- entités du groupe (séparés par une virgule)

Lister uniquement les noms des groupes : `compgen -g`

# Gestion des utilisateur.ices et groupes

Ces opérations (création, suppression modification) nécessitent d'être **root** (administrateur).

	utilisateur	groupe
Ajouter	<code>adduser</code> / <code>useradd</code>	<code>addgroup</code> / <code>groupadd</code>
Supprimer	<code>deluser</code>	<code>groupdel</code>
Modifier	<code>usermod</code>	<code>groupmod</code>

Exemple :

ajouter un.e utilisateur.ice existant à un groupe existant : `adduser username groupname`

[https://doc.ubuntu-fr.org/tutoriel/gestion\\_utilisateurs\\_et\\_groupes\\_en\\_ligne\\_de\\_commande](https://doc.ubuntu-fr.org/tutoriel/gestion_utilisateurs_et_groupes_en_ligne_de_commande)

# Gestion des utilisateur.ices et groupes

- **groupadd *groupname*** : Ajoute un groupe d'utilisateurs.  
Exemple :  
prompt> **groupadd but1a**
- **groupdel *groupname*** : Détruit un groupe d'utilisateurs.  
Exemple :  
prompt> **groupdel but1a**
- **useradd *username*** : Ajoute un utilisateur  
Exemple :  
prompt> **useradd jfa**
- **usermod *username*** : Modifie les paramètres d'un compte utilisateur  
Exemple :  
prompt> **usermod -c "This is the best user" jfa**  
prompt> **usermod -d /home/jfahome jfa**  
prompt> **usermod -e 2020-05-29 jfa**

<https://www.geeksforgeeks.org/usermod-command-in-linux-with-examples/>



# Modifier son mot de passe

```
prompt> passwd
```

```
Changing password for Jean
```

```
Old password : *****
```

```
New password : *****
```

```
Re-enter new passwd : *****
```

Un mdp :

- Doit être difficile à trouver !  
(mélanger des chiffres, des lettres, des majuscules/minuscules, mettre des caractères spéciaux, et qu'il ait suffisamment de symboles (plus de 8))
- Ne doit pas être partagé

Mdp oublié ?

- Sous linux si vous avez oublié **votre** mot de passe utilisateur, l'administrateur (root) peut vous le changer.
- Si c'est le mot de passe de **root** que vous avez oublié, tout est perdu ! **Il faut réinstaller le système !**

# A propos de l'utilisateur root, de sudo et de su

Rappel : **root** est un utilisateur particulier qui a **tous les droits**. Certaines commandes nécessitent d'avoir les droits root. Quand on est un utilisateur autre que root et qu'on veut administrer, il faut « prendre les droits root ».

Souvent, l'utilisateur root n'a **pas de mot de passe dédié**, et on ne peut pas se connecter en tant que root (c'est sécurisé ! :)))

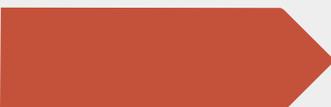
Si on en a le droit (liste de sudoers), on peut lancer une commande avec les droits administrateurs : **sudo commande** (demande **\*notre\*** mdp utilisateur et lance **commande** en se faisant passer pour **root**)

Exemple :

```
$ cat /etc/group | grep sudo
sudo:x:27:Tom,nanis,vaginay241
$ whoami
nanis
$ sudo whoami
root
```

Il faut (généralement) mettre **sudo** à chaque ligne. Si on a la flemme : **sudo bash** ouvre une session en tant que root (changement de prompt), et on peut lancer des commandes sans **sudo** alors qu'on aurait dû en temps normal). Exemple : **wc /etc/shadow**

Si on assigne un mdp à l'utilisateur root, alors on peut se connecter en tant que root (via la commande **su**). Mais c'est moins sécurisé, vu que tous les utilisateurs qui vont utiliser **su** doivent connaître le mdp de root (très mauvaise pratique de partager un mdp **#!/**).



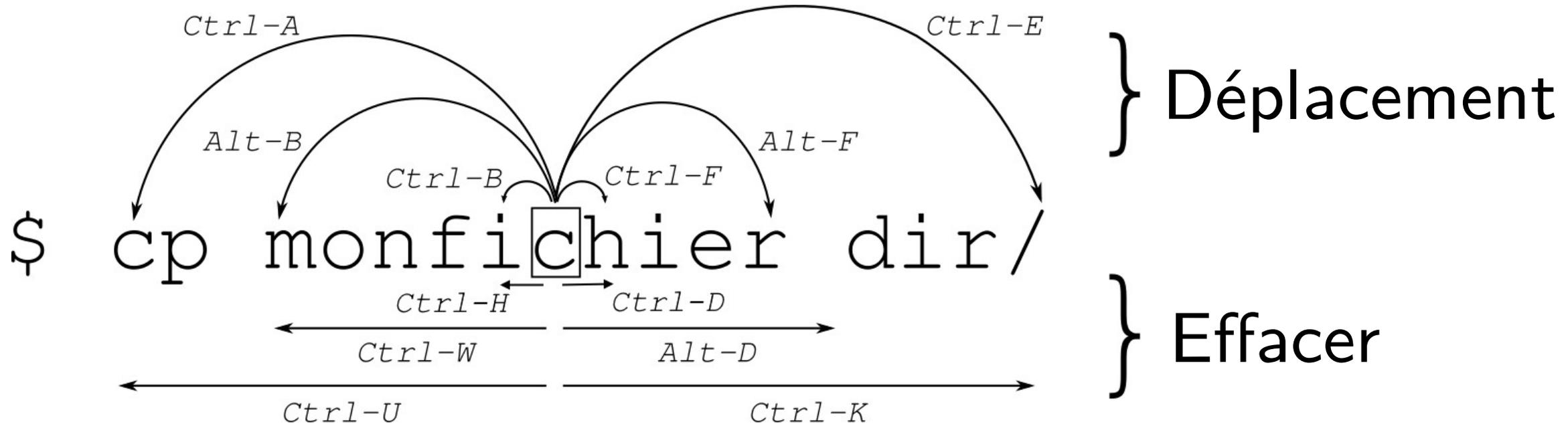
# **Bonus : protips sur l'utilisation du terminal**

# Utilisation du terminal

Bash (et d'autres CLI) utilise l'outil **readline** pour interagir via la ligne de commandes. À ce titre, on a accès à quelques raccourcis claviers :

- **tab** : autocompletion
- **^H** ou **DEL** : annule le dernier caractère frappé,
- **^L** : vide la fenêtre du terminal (commande **clear**)
- **^U** : annule tous les caractères de la ligne,
- **^C** : interrompt la commande en cours
- **^D** : simule une fin de fichier (eof)
- **↑** : parcourt l'historique de commandes
- **^R** : rechercher dans l'historique de commandes
- **^Z** : suspend l'application (retour à Bash ; faire **fg** pour rendre la main à la commande suspendue)

# Utilisation du terminal



Raccourcis inspirés de l'éditeur de texte **Emacs**, mais un mode **vi(m)** est possible ;) **set -o vi** pour essayer.

Plus de commandes : <https://ss64.com/bash/syntax-keyboard.html>



Fin prévisionnelle  
cours 3.

**Teaser TD3** : arborescence du système de fichiers Linux (+ un peu de gestion d'utilisateurs et groupes)