



1 – Démarrage

Nous avons vu qu'un ordinateur étant décomposable principalement en trois parties :

- 1) le **microprocesseur (CPU)** comportant
 - une **Unité de Commande (UDC)** qui communique avec la mémoire et interpréter les instructions
 - une **Unité Arithmétique et Logique (UAL)** qui exécute les calculs que lui demande l'UDC.
 - des mémoires volatiles, très petites mais très rapides : les **registres** (accès 1ns).
- 2) la **mémoire** notamment :
 - la **ROM (Read-Only-Memory)** : de petite taille mais non volatile.
 - la **mémoire vive / RAM (Random-Access-Memory)** : plus grande, plus lente (1µs) et volatile.
 - la **mémoire de masse** : encore plus grande, plus lente (accès 1ms) mais non volatile.
- 3) les **périphériques** : écran, clavier, souris...

Un ensemble de **bus** permettant aux différentes parties de communiquer.

Le **chargeur d'amorçage (bootloader)** : Programme exécuté automatiquement au démarrage.

Ce programme va notamment permettre le chargement du noyau du **système d'exploitation** (Operating System OS) dans la RAM.

2 - Accès à un terminal Linux

- Directement sur un PC sous Linux.
- Avec une solution en ligne (Capytale) ou un autre (<https://bellard.org/jslinux/>)
- En se connectant à un serveur distant en SSH chez un fournisseur (exemple <https://www.alwaysdata.com/fr/>)

Quelques informations à savoir sur l'**invite de commande** : `qui @ machine : où $ _`

Ci-dessus **où** désigne le répertoire de travail, celui sur laquelle se trouve la console actuellement.

Sous Linux, la **racine de l'ordinateur** se note `/`.
 A la racine `/`,
 on trouve plusieurs répertoires dont le répertoire `home`
 qui contient lui-même un répertoire par utilisateur

Racine	/
home	/home
alice	/home/alice
bob	/home/bob

La commande **pwd** (pour **print working directory**) permet d'obtenir l'adresse absolue du répertoire sur le système :

```
rv@monordi:~$ pwd          renvoie          /home/rv  puisque rv et ~
```

3 – Inspection de contenu

La commande **ls** (pour **list**) permet de lister le contenu d'un répertoire : elle affiche les fichiers et les sous-répertoires (souvent dans une autre couleur ou en gras).

La commande **ls -l** (pour **list long**) → plus d'informations

La commande **ls -a** (pour **list all**) → même les choses cachées, celles dont le nom commence par un `.`

La commande **ls -al** combine les deux.

```
testprofad@ssh1:~$ ls -la
total 4
Type Droits      Nb liens  Propriétaire Groupe      Taille Modif      Nom
d     rwxrwx---    4         root        testprofad  51     May 31 18:29 .
d     rwxr-x--x   826       root        root         0      May 31 22:06 ..
d     rwxr-xr-x    5         root        root         69     May 31 12:46 admin
-     rw-----    1         testprofad testprofad   18     May 31 18:29 .bash_history
d     rwxr-xr-x    2         testprofad testprofad   24     May 31 12:46 www
```

Type : - pour fichier, d pour directory.

Nb liens : nombre de liens dont `.` Et `..` : « `.` » est le **répertoire lui-même**, « `..` » est le **répertoire parent**.

Taille : la taille en blocs mémoire (souvent un bloc correspond à 512 octets).

La commande **cat** (pour **concatenate/contatener**) permet d'afficher, d'écrire et des transférer des données entre fichiers.

\$ cat fichier1

Renvoie le contenu de fichier1.

\$ cat > fichier1

Ecrit (en mode « w ») dans le fichier le contenu tapé au clavier en mode « écrase ».

\$ cat >> fichier1

Idem mais en mode « a » pour append, rajoute à la fin.

\$ cat f1 f2 > f3

Concatène les fichiers f1 et f2 et place cela dans le fichier f3.

\$ cat fichier2 > fichier1

Remplace le contenu de fichier1 par celui de fichier2.

\$ fichier2 >> fichier1

Rajoute le contenu de fichier2 à la fin de fichier1.

Remarque : **\$ gcc hello.c -o bonjour**

\$./bonjour

Compile avec le compilateur gcc le code-source hello.c et avec **-o** (comme out/sortie) nomme **bonjour** l'exécutable. Exécute le fichier **bonjour** présent dans le répertoire courant.

4 - Se déplacer dans l'arborescence

La commande **cd** (pour **change directory**) permet d'aller dans un autre répertoire dont on fournit soit l'**adresse relative**, soit l'**adresse absolue**.

~\$ cd www

Rend toi dans le répertoire **www** en partant du répertoire **~** où tu te trouves actuellement.
→ Adresse relatif

rv@monordi :~/ect\$ cd ~/www

Rend toi dans le répertoire **~/www** c'est à dire **/home/rv/www** quelque soit l'endroit où tu te trouves actuellement.
→ Adresse absolue

~\$ cd ..

Remonte dans le parent du répertoire actuelle (**~**) : on arrive donc dans le répertoire **/home**.
→ Adresse relative

5 - Compléments

On peut lister un répertoire sans avoir à y aller : **rv@monordi:~\$ ls /home/alice/documents**

L'utilisateur rv veut accéder au contenu du répertoire documents d'alice. Bien entendu, il faut qu'il en ai le droit. Nous verrons comment gérer cela dans une autre activité.

Commandes Windows équivalentes :

Racine /
Racine C:\

cd
cd

ls
dir

cat
edit ou editln

6 - Rôles du système d'exploitation

Les systèmes d'exploitation font office de tampon entre programmes et matériel technologique. Ils ont pour rôles

1. de **garantir l'intégrité du système** (en refusant qu'un programme ne provoque une erreur fatale)
2. de **permettre la communication entre un programme et les périphériques**
3. de **répartir les ressources communes** entre **plusieurs programmes** en « même temps »

Les programmes vont appel aux périphériques en utilisant les **appels-systèmes** du système d'exploitation.

