

Chapitre 1 : Introduction

Section 1.1 : Introduction

1.1.1 Composition d'un système informatique

Le système informatique le plus simple est **un processeur** (*cpu*) et **une mémoire**. Ce processeur est capable de : *lire et écrire* des l'informations en mémoire et de *réaliser des calculs*.

Systèmes Unix

Unix est un nom générique donné à une famille de systèmes d'exploitations. On y retrouve MacOS, FreeBSD et Linux qui sont les plus utilisés.

Un système unix est composé de trois grands types de logiciels:

1. **Le noyau du système d'exploitation** : chargé au démarrage de la machine, il se charge de toutes les interactions entre les logiciels et le matériel.
2. **Les bibliothèques** : Nombreuses, elles facilitent l'écriture et le développement d'applications.
3. **Les programmes utilitaires** : utilisés pour résoudre une série de problèmes.

API signifie *Application Programming Interface*

/usr : utilitaires et bibliothèques installées sur le système

/bin et **/sbin** : utilitaires de base nécessaire à l'admin. syst.

/tmp : fichiers temporaires (effacé au redémarrage)

/etc : configuration du système

/home : répertoire personnel des utilisateurs

/dev : fichiers spéciaux

/root : administrateur système

En Unix une application est composée de un ou plusieurs processus.

Un processus : ensemble cohérent d'instructions qui utilisent une partie de la mémoire et sont exécutées sur un processeur.

Les processus peuvent utiliser des ressources en mémoire qu'il libérera lors de sa terminaison et retourner un entier au processus parent (0 si Ok sinon autre chose)

Shell

Un **shell** est un programme qui à été spécialement conçu pour faciliter l'utilisation d'un système Unix via le clavier, généralement appelée console ou terminal. Sa puissance vient de sa capacité à écrire des commandes enchaînées : <, >, >>, |

Une **pipe** | est une redirection de la sortie standard d'un programme vers l'entrée std d'un autre sans passer par un fichier intermédiaire.

cat : affiche le contenu d'un fichier sur la sortie Standard

mv : utilitaire pour renommer ou déplacer un fichier ou dossier.

head / tail : extrait le début / la fin d'un fichier.

wc : compte le nombre de (lignes (-l), mots, caractères).

sort : trie le fichier par ordre alphabétique

uniq : retire les doublons (*Attention : fichier trié au préalable*)

tar : permet de regrouper des fichiers dans une archive (fonctionne souvent avec gzip)

(c:compress/x:extract , z:utilise gzip , v:verbose , f: indique la donnée à compresser)

```
tar czvf fichier1.txt dossier1 #compresser
tar xzvf bob.tar.gz #decompresser
```

gzip/gunzip : compression / décompression d'un fichier .gz

cp : copie un fichier ou dossier (-r pour les dossiers).

rm : efface un fichier ou dossier. (-rf : suppression récursive)

mkdir : crée un répertoire.

rmdir : efface un répertoire vide.

cd : change le répertoire courant.

pwd : affiche le répertoire courant.

./prog : est utilisé pour lancer le programme prog.

grep : utilitaire permettant d'extraire d'un fichier les lignes qui contiennent ou non une chaîne de caractère passée en argument. (cette commande est souvent utilisée après une pipe ex : cat fichier.txt | grep "bonjour")

```
grep -r "truc" # recherche recursive
grep -n "truc" # affiche la ligne
```

echo "Blabla" >> file.txt copie le texte Blabla dans le fichier file.txt.

man : lire les pages de manuel d'un système Unix.

Un script bash commence par:

```
#!/bin/bash
$# # nombre d args
$1 # arg1 ..
```

```
$@ # liste des args  
if [Cond]; then ...fi  
exit 0
```

```
$i-eq $j # vrai si les deux variables sont différentes  
$i -eq $j # vrai si les deux variables sont équivalentes  
$s = $t #vrai quand les deux chaines de caractères sont équival  
$i -lt $j # vrai si i est strictement inférieure à j  
$i -ge $j # vrai si i est supérieure inférieure à j  
-z $s #vrai si la variable est vide
```