Chapitre 1 - Introduction à Linux

1) Linux et les systèmes libres

Linux est le noyau d'un **système d'exploitation** (OS) dont debian ou Ubuntu sont des distributions.

- C'est un système d'exploitation libre dont il existe de nombreuses variantes appelées distributions fonctionnant elles-mêmes sur le noyau linux.
- Linux désigné à la fois le noyau (kernel) qui lance la machine et gère le bas niveau (proche du métal) et le système d'exploitation lui même (les applications).
- Linux est un dérivé d'**UNIX**, système crée dans les années 70, codé en **langage C** (crée à cette occasion). Ces deux "logiciels" : C et UNIX constituent à la fois le langage et le système les plus importants de l'histoire informatique (la majorité des langages actuels dérivent du C et la majorité des machines professionnelles tournent sur un dérivé d'UNIX).
- UNIX et ses dérivés sont présents partout : tous les smartphones fonctionnent sur un de ses dérivés (Linux pour android, bsd pour iOS) ainsi que les mac.
- UNIX fonctionne sur n'importe quelle machine : une caméra IP, un super ordinateur (les 500 plus puissants utilisent tous linux), un chromecast, une switch etc.
- Il existe des versions payantes de Linux (pour les professionnels), debian est un système d'exploitation gratuit et reconnu par les professionnels pour sa grande stabilité. Il n'est pas rare de rencontrer un serveur debian fonctionnant sans redémarrer depuis plusieurs années :

2) Usage courant de linux

Hormis pour des logiciels très spécifiques, utiliser windows, linux ou OSX ne change rien. La grande majorité des logiciels existent sur toutes les plateformes hormis quelques exceptions notables :

- Photoshop et la suite adobe : windows et osx. De nombreuses alternatives
- les jeux vidéos : windows (possible de jouer sous linux ou osx mais demande du courage).

3) Grandes familles de systèmes d'exploitation

On en rencontre massivement deux :

- Windows et ses dérivés (MSDOS (1985), Windows NT (1999), windows 7-¿10 (2008)). Domine le marché du PC "personnel"
- UNIX et ses dérivés : bsd (systèmes embarqués, réseaux d'entreprises), linux (partout dont android, super calculateurs, PC personnels, serveur web), OSX & iOS (produits apple)

4) Utilisation d'un système UNIX

Les trois couches d'UNIX

- **Le noyau (kernel)** : proche du hardware. Lance la machine, gère la carte graphique, le réseau etc.
- **La coquille (shell)** : programme qui permet d'exécuter des utilitaires et d'intéragir (via les fenêtres ou le terminal)
- **les utilitaires** (ls, firefox) : les programmes qu'on fait tourner grâce au shell.

Deux modes d'utilisation :

- graphique (GUI : graphical user interface)
- terminal (CLI : command line interface)

Le premier vous connaissez : fenêtres, clic clic clic.

Le second repose sur la console. On est devant un terminal qui exécute une boucle REPL (read eval print loop) :

1. On tape une commande. Le shell la lit (read)
2. Le système traite la commande et calcule une réponse (eval)
3. Il affiche une réponse (print)
4. On recommence (loop)

Dans un extrait de code on symbolise une commande shell par un $ (souvent appelé prompt).
Dans un terminal on tape des commandes, elles sont exécutées par le shell. Celui de linux est souvent bash. bash est un langage de programmation interprété.
Vous pouvez aussi créer des scripts bash (scripts python moins faciles à écrire). bash est présent sous windows. OSX utilise zsh.

```
mikael@mikael-HP-EliteBook-840-G3:~$ pwd
/home/mikael
mikael@mikael-HP-EliteBook-840-G3:~$ cd enseignement/2024_2025/Terminale_NSI/projet/microbit/MNP/
mikael@mikael-HP-EliteBook-840-G3:~/enseignement/2024_2025/Terminale_NSI/projet/microbit/MNP$ ls -l
total 48
-rw-rw-r-- 1 mikael mikael  6368 feb 18 15:56 client_MNP.py
-rw-rw-r-- 1 mikael mikael  8188 feb 19 08:29 port_serie_ihm.py
-rw-rw-r-- 1 mikael mikael  1154 feb 17 08:57 port_serie.py
-rw-rw-r-- 1 mikael mikael   578 feb 16 17:01 port_serie_TX.py
-rw-rw-r-- 1 mikael mikael 13853 feb 19 08:41 projet_MNP.odt
-rw-rw-r-- 1 mikael mikael  6694 feb 18 15:53 router_MNP.py
mikael@mikael-HP-EliteBook-840-G3:~/enseignement/2024_2025/Terminale_NSI/projet/microbit/MNP$
```

On navige avec *cd*, on affiche les contenus avec *ls*, on se repère avec *pwd*.

On peut accéder à un terminal physiquement (devant la machine) en lançant un programme ou à distance grâce au service réseau ssh (secured shell). C'est ainsi que sont administrées la majorité des machines 'professionnelles'.

5) Arborescence UNIX

Tous les systèmes UNIX accordent une grande importance aux fichiers. **UNIX VOIT SES PÉRIPHÉRIQUE ET SES PROCESSUS COMME DES FICHIERS.** Par exemple :

```
mikael@mikael-HP-EliteBook-840-G3:~$ ls /
bin    dev   lib    libx32      mnt   root  snap        sys  var
boot   etc   lib32  lost+found  opt   run   srv         tmp
cdrom  home  lib64  media       proc  sbin  swapfile    usr
mikael@mikael-HP-EliteBook-840-G3:~$
```

- / le dossier racine de l'arborescence
- Tous les dossiers qui se terminent par bin contiennent des exécutables en binaire.
- home : dossiers des utilisateurs
- dev : le matériel
- etc : les reglages
- root : dossier de l'utilisateur root
- mnt et media : les points de montage des disques et partitions externes (là où apparaissent les clés usb etc.)

6) Permissions

La sécurité sous unix est gérée par la notion de permission.

- Un utilisateur ne peut pas faire ce qu'il veut.
- Le super utilisteur **root** peut tout faire. Devenir root avec *$ su*, exécuter une commande comme root avec *$ sudo commande*.
- Pourquoi ? Parce qu'en tant que root il suffit de 8 caractères pour effacer tous les disques de la machine...

L'affichage détaillé d'un fichier (ls -lah) montre les permissions de

- l'utilisateur courant
- de son groupe
- et de tout le monde

```
mikael@mikael-HP-EliteBook-840-G3:~/enseignement/2024_2025/Terminale_NSI/projet/microbit/MNP$ ls -lah
total 56K
drwxrwxr-x 2 mikael mikael 4,0K jul  1 17:42 .
drwxrwxr-x 4 mikael mikael 4,0K feb 16 09:10 ..
-rw-rw-rw- 1 mikael mikael 6,3K feb 18 15:56 client_MNP.py
-rw-rw-r-- 1 mikael mikael 8,0K feb 19 08:29 port_serie_ihm.py
-rw-rw-r-- 1 mikael mikael 1,2K feb 17 08:57 port_serie.py
-rw-rw-r-- 1 mikael mikael  578 feb 16 17:01 port_serie_TX.py
-rw-rw-r-- 1 mikael mikael  14K feb 19 08:41 projet_MNP.odt
-rw-rw-r-- 1 mikael mikael 6,6K feb 18 15:53 router_MNP.py
-rwxrwxrwx 1 mikael mikael    0 jul  1 17:42 script.sh
mikael@mikael-HP-EliteBook-840-G3:~/enseignement/2024_2025/Terminale_NSI/projet/microbit/MNP$
```

Traduction

`-` : pas activé
`d` : directory
`r` : droit de lecture
`w` : droit d'écritude
`x` : droit d'exécution

- Le fichier `client_MNP.py` appartient à l'utilisateur `mikael` et au groupe `mikael`. Tout le monde peut le lire et peut y écrire (le modifier). Personne ne peut l'exécuter.
- Le fichier `port_serie_ihm.py` appartient à l'utilisateur `mikael` et au groupe `mikael`. L'utilisateur `mikael` et les utilisateurs du groupe `mikael` peuvent le lire et peut y écrire (le modifier). Les autres utilisateurs ne peuvent que le lire. Personne ne peut l'exécuter.
- Le fichier `script.sh` appartient à l'utilisateur `mikael` et au groupe `mikael`. Tout le monde peut le lire, peut y écrire (le modifier)et l'exécuter.

On change les permissions avec `chmod` : `$ chmod +x client_MNP.py` rendra ce fichier exécutable.

7) Processus

- Programme : du texte, décrivant des opérations à la machine.
- processus : un programme en cours d'exécution par la machine.

Chaque fois qu'on lance un programme, UNIX crée un processus. On accède aux processus avec `ps`.

```
mikael@mikael-HP-EliteBook-840-G3:~/enseignement/2024_2025/Terminale_NSI/projet/microbit/MNP$ ps -f --pid=904918
UID          PID    PPID  C STIME TTY          TIME CMD
mikael    904918    2365  4 jun24 ?        07:29:28 /usr/bin/python3 /usr/bin/mu-editor
```

Cette commande affiche les informations sur le processus qui exécute le programme `mu-editor`.
On peut afficher la liste de tous les processus exécutés sur la machine avec la commande `ps -ef`.

Chaque processus possède :

- un PID (Processus Identifier) : ici, le PID du processus est 904918.
- un PPID (Parent Processus Identifier), il s'agit du processus parent qui permis de lancé le processus PID.
  Ici le PPID de `mu-editor` est 2365. Il s'agit du processus `/usr/bin/gnome-shell` (une console de commande).

Si le processus `mu-editor` ne fonctionne plus correctement (il est planté !), on peut à l'aide de la commande `kill` forcer l'arrêt du processus (on tue le processus !).
Exemple : La commande `$ kill 904918` force l'arrêt du processus `mu-editor`.

La commande `$ top` permet d'afficher le gestionnaire de ressources, indiquant les ressources CPU et mémoire utilisées par chaque processus.

8) Redirection des entrées sorties

UNIX fonctionne principalement avec des petits programmes exécutant quelques tâches simples, le plus souvent une seule. Ils communiquent avec des flux de texte qu'on peut enchaîner ou rediriger. On peut, par exemple :

- écrire dans un fichier : `$ cat cours.txt >> bonjour.txt`
  va recopier `cours.txt` dans `bonjour.txt`.
- rediriger : `$ ps -ef` affiche 20 pages... `$ ps -ef | less` les fait défiler une par une !

9) Les rudiments du réseau

Afficher l'état du réseau.

```
$ ip a
$ ifconfig
```

Puis-je joindre une machine ?

```
$ ping 192.168.1.1
$ ping google.com
```

CTRL+C pour arrêter