Systèmes d'exploitation

Le système et la ligne de commande

Mise en situation

Quand vous allumez votre ordinateur, vous voyez un bureau avec des icônes, vous pouvez lancer un navigateur, écouter de la musique et écrire un programme Python en même temps. Qui orchestre tout cela ? Ce n'est pas le processeur seul (il ne sait qu'exécuter des instructions une par une), ni les applications elles-mêmes. C'est le **système d'exploitation**.

Qu'est-ce qu'un système d'exploitation ?

Un **système d'exploitation** (en anglais *Operating System*) est un programme qui se charge en mémoire vive dès le démarrage de l'ordinateur et qui y reste jusqu'à son extinction. Son rôle est de servir d'**intermédiaire entre le matériel et les applications** : il gère les ressources de la machine et les met à disposition des logiciels.

Concrètement, le système d'exploitation assure quatre fonctions essentielles :

— **gestion des programmes** : il lance les applications, leur attribue du temps processeur et donne l'illusion qu'elles s'exécutent simultanément (c'est le **multitâche**) ;

— **gestion de la mémoire** : il répartit la mémoire vive entre les programmes en cours et empêche qu'un programme n'accède à la mémoire d'un autre ;

— **gestion des fichiers** : il organise les données sur le disque en dossiers et fichiers, et contrôle les droits d'accès (qui peut lire, écrire, exécuter) ;

— **gestion des périphériques** : il communique avec le matériel grâce à des programmes spéciaux appelés **pilotes** (*drivers*).

Libre ou propriétaire ?

Les systèmes d'exploitation se répartissent en deux grandes familles.

Windows (Microsoft) et **macOS** (Apple) sont des systèmes **propriétaires** : leur code source n'est pas accessible au public. L'utilisateur ne peut ni l'étudier, ni le modifier, ni le redistribuer.

GNU/Linux est un système **libre** : son code source est accessible en ligne. L'utilisation, l'étude, la modification et la redistribution sont permises. Cette philosophie, portée par Richard Stallman (projet GNU, 1983) et Linus Torvalds (noyau Linux, 1991), garantit le contrôle du programme par ses utilisateurs.

Linux est le système dominant sur les serveurs, les supercalculateurs, les systèmes embarqués (box Internet, robots, drones) et les smartphones (Android repose sur un noyau Linux). Sur les postes de travail, Windows reste le plus utilisé.

Le shell : communiquer avec le système

Pour interagir avec le système d'exploitation, on utilise un **shell** (interpréteur de commandes). Il en existe deux catégories :

— les **interfaces graphiques** (GUI), que l'on utilise au quotidien avec la souris : bureau, fenêtres, icônes ;

— les **interfaces en ligne de commande** (CLI), où l'on tape des commandes textuelles dans un **terminal**.

En NSI, nous utiliserons principalement le terminal et le shell **Bash**, qui est le shell standard des systèmes GNU/Linux. À l'ouverture d'un terminal, une invite de commande apparaît :

```
utilisateur@machine:~$
```

Elle indique le nom de l'utilisateur, le nom de la machine et le répertoire courant (`~` signifie le dossier personnel). On tape sa commande après le `$`, puis on appuie sur Entrée.

L'arborescence de fichiers

La racine et les dossiers principaux

Sous GNU/Linux, tous les fichiers et dossiers sont organisés dans une **arborescence** qui part d'un unique point appelé la **racine**, notée `/`. Il n'y a pas de lettres de lecteur (`C:`, `D:`) comme sous Windows.

Dossier	Contenu
`/`	La racine, point de départ de l'arborescence
`/home`	Les dossiers personnels des utilisateurs
`/bin`	Les programmes essentiels (commandes de base)
`/etc`	Les fichiers de configuration du système
`/tmp`	Les fichiers temporaires

Deux particularités : sous Linux, les chemins utilisent le slash `/` (et non l'antislash `\` de Windows), et les fichiers dont le nom commence par un point (par exemple `.bashrc`) sont **cachés**.

Chemins absolus et relatifs

Pour désigner un fichier dans l'arborescence, on peut utiliser deux types de chemins.

Le **chemin absolu** part toujours de la racine `/`. Il est complet et sans ambiguïté. Par exemple : `/home/elie/documents/tutu.txt`.

Le **chemin relatif** part du **répertoire courant**. Pour remonter au dossier parent, on utilise `..`. Par exemple, depuis le dossier `famille`, le chemin relatif vers `tata.txt` est `../vacances/tata.txt`.

Comment les distinguer ? Un chemin qui commence par `/` est absolu ; un chemin qui ne commence pas par `/` est relatif.

Symbole	Signification
`/`	La racine
`~`	Le dossier personnel (`/home/utilisateur`)
`.`	Le répertoire courant
`..`	Le répertoire parent

Les commandes Bash essentielles

La syntaxe d'une commande Bash est : `commande -options arguments`. Les options commencent par un tiret et modifient le comportement de la commande.

Naviguer et explorer

Commande	Rôle	Exemple
`pwd`	Afficher le répertoire courant	`pwd` → `/home/alice`
`ls`	Lister le contenu d'un répertoire	`ls`
`ls -l`	Lister avec les détails	`ls -l`
`cd chemin`	Se déplacer	`cd Documents`
`cd ..`	Remonter au dossier parent	`cd ..`
`cd ~`	Revenir au dossier personnel	`cd ~`

Gérer les fichiers et dossiers

Commande	Rôle	Exemple
`cat fichier`	Afficher le contenu	`cat notes.txt`
`touch fichier`	Créer un fichier vide	`touch brouillon.txt`
`mkdir dossier`	Créer un dossier	`mkdir projets`
`cp src dest`	Copier	`cp notes.txt copie.txt`
`mv src dest`	Déplacer ou renommer	`mv ancien.txt nouveau.txt`
`rm fichier`	Supprimer	`rm brouillon.txt`

Autres commandes utiles

Commande	Rôle
`man commande`	Afficher le manuel d'une commande
`find chemin -name "nom"`	Rechercher un fichier
`grep motif fichier`	Rechercher un texte dans un fichier
`sudo commande`	Exécuter en tant qu'administrateur

⚠ **Attention.** `rm` supprime définitivement (pas de corbeille) et `sudo` donne tous les droits sur la machine. Ces commandes doivent être utilisées avec précaution.

Les droits d'accès

Sous GNU/Linux, chaque fichier et chaque dossier possède des **droits d'accès** qui déterminent qui a le droit de faire quoi. On distingue trois actions :

— **lire** (`r`, pour *read*) : consulter le contenu d'un fichier, ou lister un dossier ;

— **écrire** (`w`, pour *write*) : modifier un fichier, ou ajouter et supprimer des fichiers dans un dossier ;

— **exécuter** (`x`, pour *execute*) : lancer un fichier lorsqu'il s'agit d'un programme, ou entrer dans un dossier.

Ces droits sont définis séparément pour trois catégories d'utilisateurs : le **propriétaire** du fichier, son **groupe**, et les **autres** utilisateurs.

Lire les droits avec `ls -l`

La commande `ls -l` affiche les droits au début de chaque ligne, sous la forme de dix caractères :

```
-rwxr-xr-- 1 alice profs 1240 12 mars sauvegarde.sh
```

Le bloc `-rwxr-xr--` se lit en quatre parties :

Partie	Caractères	Signification
Type	`-`	`-` pour un fichier, `d` pour un dossier
Propriétaire	`rwx`	alice peut lire, écrire et exécuter
Groupe	`r-x`	le groupe profs peut lire et exécuter, mais pas écrire
Autres	`r--`	les autres utilisateurs peuvent seulement lire

Un tiret `-` à la place d'une lettre signifie que le droit correspondant est **refusé**.

Modifier les droits avec `chmod`

La commande `chmod` (*change mode*) modifie les droits d'un fichier. On indique la catégorie (`u` propriétaire, `g` groupe, `o` autres), un `+` ou un `-`, puis le droit visé :

```
chmod g+w sauvegarde.sh
chmod o-r sauvegarde.sh
```

Seuls le propriétaire du fichier et l'administrateur (avec `sudo`) peuvent changer les droits.

Résumé

Concept	Ce qu'il faut retenir
Système d'exploitation	Gère les ressources et sert d'interface matériel/applications
Multitâche	Le système bascule rapidement entre les programmes
Logiciel libre	Code source accessible, modifiable, redistribuable
Logiciel propriétaire	Code source fermé
Shell	Interface de communication avec le système (GUI ou CLI)
Bash	Shell en ligne de commande standard sous Linux
Racine `/`	Point de départ de l'arborescence
Chemin absolu	Part de la racine `/`
Chemin relatif	Part du répertoire courant
Commandes essentielles	`ls`, `cd`, `cat`, `touch`, `mkdir`, `cp`, `mv`, `rm`, `pwd`
Droits d'accès	Lecture (`r`), écriture (`w`), exécution (`x`) pour le propriétaire, le groupe et les autres ; affichés par `ls -l`