Architecture et OS | TP 01 | # Découverte de Linux

2.1 Découvrir l'OS au manchot

Linux dans le monde informatique (Tux)

Linux est un système d'exploitation très largement utilisé dans le domaine informatique. On le retrouve notamment sur de nombreux serveurs, dans les centres de calcul scientifique, dans les réseaux ou encore dans certains objets techniques.
Son développement a été initié au début des années 1990 par Linus Torvalds. Linux appartient à la famille des systèmes inspirés d'Unix, conçus dès les années 1970 pour permettre l'utilisation efficace de machines complexes.

Définition 1 — Logiciel libre

Un logiciel est dit **libre** lorsqu'il est possible :

- d'en consulter le code source ;
- de le modifier ;
- de le redistribuer.

Le système GNU/Linux constitue un exemple important de logiciel libre. Cette ouverture explique en partie sa diffusion dans le monde scientifique et technique.

Remarque 1. Dans la salle, les ordinateurs utilisent principalement le système **Windows**. Cependant, grâce à un outil appelé **WSL** (**Windows Subsystem for Linux**), il est possible d'exécuter également un environnement Linux.

Préparer Linux

Ouvrir **PowerShell** depuis le menu démarrer.
Saisir ensuite la commande suivante :

Terminal

```
wsl --install
```

Cette commande permet de finaliser l'accès à l'environnement Linux pour votre session utilisateur.

Lancer Linux

Fermer PowerShell puis lancer l'application `Ubuntu` depuis le menu démarrer.
Lors du premier lancement, créer un **nom d'utilisateur Linux** ainsi qu'un **mot de passe**.

2.2 Premières commandes dans le terminal

Définition 2 — Terminal

Un **terminal** permet d'envoyer des commandes à l'OS. Chaque commande correspond à l'exécution d'un **programme** chargé d'effectuer une tâche précise.

Identité

Saisir la commande suivante, qui permet de savoir qui vous êtes :

Terminal

```
whoami
```

Répertoire

Saisir la commande suivante :

Terminal

```
pwd
```

Cette commande permet d'afficher le **répertoire courant**.

Date

Saisir la commande suivante, le nom suffit normalement pour comprendre :

Terminal

```
date
```

Manipulation

La commande suivante est une des plus importante :

Terminal

```
ls
```

Cette commande **affiche les fichiers et dossiers** présents dans le répertoire courant.

Remarque 2. Le dossier peut sembler presque vide. En réalité, sous Linux, certains fichiers sont **cachés par défaut**.

Ces fichiers commencent généralement par un point . Ils sont souvent utilisés pour stocker des paramètres de configuration du système ou du terminal.

Options

Dans le terminal, il est possible d'ajouter des **options** à une commande. Une option permet de modifier le comportement du programme exécuté.
On rajoute quasi tout le temps des options aux commandes, et il n'y a pas de limite !

Manipulation

Saisir la commande suivante :

Terminal

```
ls -a
```

L'option `-a` signifie **all**. Elle demande au programme `ls` d'afficher également les fichiers cachés. Vous devriez maintenant observer davantage d'éléments dans le dossier courant.

2.3 Manipuler des fichiers et des dossiers

Arborescence

Dans un système d'exploitation, les informations sont organisées sous forme de **fichiers** regroupés dans des **dossiers**.
L'ensemble constitue une structure hiérarchique appelée **arborescence**. Chaque utilisateur peut créer sa propre organisation de fichiers.

Créer un dossier

Saisir la commande suivante :

Terminal

```
mkdir travail
```

La commande `mkdir` signifie **make directory**. Elle permet de créer un nouveau dossier nommé ici `travail`.

Manipulation

Observer ensuite le contenu du dossier courant :

Terminal

```
ls
```

Se déplacer

Pour entrer dans le dossier créé, saisir :

Terminal

```
cd travail
```

La commande `cd` signifie **change directory**. Elle permet de changer de dossier courant.

Manipulation

Vérifier le dossier courant :

Terminal

```
pwd
```

Créer un fichier

Saisir la commande suivante :

Terminal

```
touch notes.txt
```

La commande `touch` permet de créer un fichier vide.

Manipulation

Observer le contenu du dossier :

Terminal

```
ls
```

Modifier un fichier avec un bloc-note

Installer l'éditeur de texte graphique `gedit` :

Terminal

```
sudo apt install gedit
```

Manipulation

Lancer maintenant l'éditeur avec la commande suivante :

Terminal

```
gedit notes.txt
```

Question

Que se passe-t-il dans le terminal lorsque le programme est lancé ? Peut-on encore saisir des commandes ?

Manipulation

Ajouter un gentil message pour votre voisin le plus proche dans notes.txt puis fermer gedit, puis lancer la commande suivante, qui permet de montrer le contenu d'un fichier.

Terminal

```
cat notes.txt
```

Manipulation

Relancer maintenant le programme avec la commande suivante :

Terminal

```
gedit notes.txt &
```

Observer à nouveau le comportement du terminal.

Question

Quelle différence observez-vous avec la situation précédente ?

Programme en arrière-plan

Le symbole & demande au système d'exploitation de lancer le programme **en arrière-plan**.
Le terminal reste alors disponible pour exécuter d'autres commandes. Chaque programme lancé correspond à l'exécution d'un **processus**.

Copier un fichier

Saisir la commande suivante :

Terminal

```
cp travail/notes.txt copie.txt
```

La commande `cp` permet de copier un fichier.

Déplacer ou renommer

Saisir :

Terminal

```
mv copie.txt sauvegarde.txt
```

La commande `mv` permet soit de **déplacer** un fichier, soit de le **renommer**.

Supprimer

Saisir :

Terminal

```
rm sauvegarde.txt
```

La commande `rm` permet de supprimer un fichier.

Attention : sous Linux la suppression est définitive.

Revenir en arrière

Saisir :

Terminal

```
cd ..
```

Cette commande permet de remonter d'un niveau dans l'arborescence.

Exercice 1 — Construire une arborescence

À l'aide des commandes vues précédemment, construire l'organisation suivante :

```
travail/
  cours/
    nsi/
      chapitre1.txt
      chapitre2.txt
    maths/
      fonctions.txt
  projets/
    python/
      jeu.py
    web/
      index.html
  notes.txt
```

Visualiser l'arborescence

Installer l'outil `tree` :

Terminal

```
sudo apt install tree
```

Manipulation

Saisir ensuite :

Terminal

```
tree
```

La commande `tree` permet d'afficher la structure hiérarchique des dossiers sous forme d'arbre.

Vision globale

Contrairement à `ls`, la commande `tree` permet de voir **toute l'organisation des fichiers en une seule fois**. Elle est très utile pour comprendre l'arborescence d'un projet.

Exercice 2 — Manipuler l'arborescence

Se placer dans votre **dossier personnel**.
En utilisant uniquement les commandes vues dans cette section, réaliser les actions suivantes :

- entrer dans le dossier `travail` et ne plus jamais y bouger,
- déplacer le fichier `notes.txt` dans le dossier `cours/nsi` (donc depuis travail)
- copier le fichier `jeu.py` dans le dossier `cours` ;
- renommer le fichier `chapitre2.txtS` en `reseaux.txt` ;
- supprimer le fichier `fonctions.txt`.

Vérifier ensuite le résultat obtenu à l'aide de la commande :

Terminal

```
tree
```

2.4 Observer les processus

Question

Rappeler la définition d'un processus et ce que l'OS prévoie pour chaque processus à leur création.

Afficher les processus

Saisir la commande suivante :

Terminal

```
ps
```

Cette commande affiche la liste des processus associés à votre terminal.

Manipulation

Lancer maintenant l'éditeur de texte en arrière-plan :

Terminal

```
gedit notes.txt &
```

Manipulation

Afficher à nouveau la liste des processus :

Terminal

```
ps
```

Repérer le processus correspondant à `gedit`.

Question

Rappeler pourquoi on peut parler de hiérarchie entre les processus ?

Affichage détaillé

Pour obtenir davantage d'informations, saisir :

Terminal

```
ps -f
```

Observer les nouvelles colonnes affichées.

Question

Quel est le PID du processus correspondant à `gedit` ?

Question

Quel est le PPID de ce processus ?

Question

À quel programme correspond ce processus père ?

2.5 Observer l'activité du processeur

Partage du processeur

Un ordinateur exécute en permanence de nombreux **processus**. Le **processeur** doit donc partager son temps de calcul entre ces différents processus.

Question

Lister les différents algorithmes d'ordonnancement que vous connaissez.

Lancer un programme intensif

Saisir la commande suivante :

Terminal

```
yes
```

La commande affiche continuellement le mot `yes`. Elle mobilise fortement le processeur.

Ne pas arrêter le programme pour l'instant.

Ouvrir un second terminal

Ouvrir une nouvelle fenêtre de terminal Ubuntu.

Dans cette nouvelle fenêtre, saisir :

Terminal

```
top
```

Cette commande affiche en temps réel les processus en cours d'exécution.

Question

Observer la liste affichée.

- Retrouver le processus correspondant à la commande `yes`.
- Observer la valeur de consommation du processeur associée.

Arrêter un processus

Dans le premier terminal (celui où la commande `yes` a été lancée), appuyer sur :

Terminal

```
Ctrl + C
```

Observer ensuite la modification dans la commande `top`.

2.6 Utiliser l'aide intégrée du système

Définition 3 — Manuel

Sous Linux, chaque commande possède une **documentation intégrée** appelée **manuel**.
Cette documentation explique le rôle de la commande et les différentes options disponibles.

Lire la documentation

Saisir la commande suivante :

Terminal

```
man ls
```

Lire les premières lignes affichées.

Manipulation

Pour quitter le manuel, appuyer sur la touche :

Terminal

```
q
```