

1 Problématique

L'interface graphique (matérialisée par *le bureau*) est le principal mode d'interaction entre la machine et l'utilisateur. Il est ainsi aisé de créer un nouveau dossier dans l'espace disque, avec la souris. Mais ce système graphique n'est qu'un surcouche du système d'exploitation qui n'existait pas à l'origine. Ainsi *Windows* est la couche graphique du système *MS-DOS*.

Il est toujours possible d'utiliser le système d'exploitation sans couche graphique. Cela peut s'avérer utile pour intervenir sur une machine distante comme un *serveur*.

Comment interagir avec le système d'exploitation sans interface graphique ?

2 Découvrir le *Shell*

Le *Shell* est un programme permettant d'interagir avec le système d'exploitation. De nos jours nous utilisons une interface graphique appelée *terminal* ou *console* qui simule l'interface des anciens ordinateurs (figure 1).

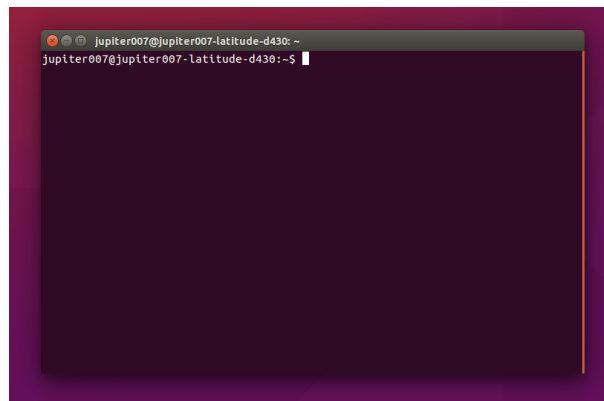


FIGURE 1 – Un terminal sous Ubuntu

Windows n'utilise pas les mêmes standards que les systèmes d'exploitation basés sur *Unix*. Nous travaillerons sur un émulateur de terminal fonctionnant sur des systèmes *Linux*. La page ci-après simule un terminal d'un système *Linux* :

<https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192>

Avant de commencer nous allons créer un compte utilisateur personnel en complétant le formulaire suivant :

<https://vfsync.org/signup>

Activité 1 :

1. Créer un compte utilisateur.
2. Dans le terminale se connecter à son compte en tapant l'instruction ci-après dans *l'invite de commande*.

```
1 vflogin mon_identifiant
```

3. Remarquer que rien ne s'affiche quand le mot de passe est tapé. Pour quelle raison ?

3 Manipuler son espace

3.1 Arborescence

Les systèmes basés sur les standards *Unix* possèdent une arborescence de dossiers et fichiers identique (figure 2).

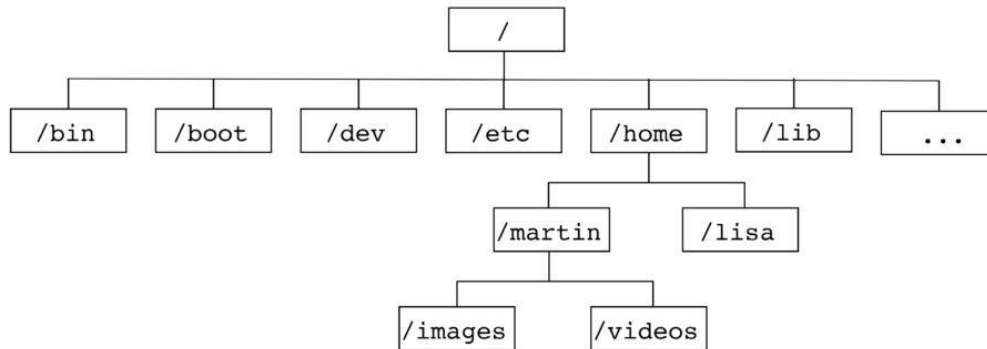


FIGURE 2 – Arborescence du système

- « / » est le répertoire racine.
- « bin » contient les commandes de base.
- « etc » contient les fichiers de configuration système.
- « home » contient les répertoires personnels des utilisateurs.

3.2 Se déplacer dans les dossiers

L'invite de commande nous donne déjà des informations.

```
1 localhost:~$
```

- *localhost* indique la machine sur laquelle nous travaillons.
- *~* indique le répertoire où nous nous trouvons. Ce signe est particulier. C'est un raccourci qui pointe vers le dossier personnel de l'utilisateur.
- *\$* indique que l'utilisateur à la main et peut entrer une instruction.

Activité 2 :

1. Taper la commande **pwd**. Elle indique le dossier courant.
2. Taper l'instruction ci-après. La commande **cd** permet de naviguer dans les répertoires.

```
1 cd /bin
```

À retenir : L'adresse est *absolue* c'est à dire qu'elle démarre de la racine /

3. Dans quel dossier sommes-nous situé ?
4. Se rendre dans le dossier *home*.
5. Taper l'instruction ci-après (en remplaçant *mon_identifiant*).

```
1 cd mon_identifiant
```

À retenir : L'adresse est *relative* c'est à dire qu'elle démarre du répertoire courant. Nous la reconnaissons car elle ne commence pas par /

6. Dans quel dossier sommes-nous situé ?

À retenir : L'instruction

```
1 cd ..
```

permet de remonter d'un niveau de répertoire.

3.3 Créer des dossiers

Activité 3 :

1. Se placer dans le répertoire personnel.
2. Taper l'instruction ci-après. La commande **mkdir** crée un répertoire.

```
1 mkdir images
```

3. Créer un répertoire *devoir* dans le répertoire personnel.
4. Créer un répertoire *photos_vacances* dans le répertoire *images*.

3.4 Visualiser l'arborescence

Activité 4 :

1. Taper l'instruction ci-après. La commande **ls** liste les répertoires et fichiers du répertoire donné.

```
1 ls ~
```

Il est possible d'utiliser la commande **ls** seule. Elle renvoie alors la liste des répertoires et fichiers du répertoire courant.

2. Se placer dans le répertoire *home*.
3. Taper **ls** puis les premières lettres du nom d'utilisateur. Appuyer sur la touche tabulation.

À retenir : La touche *Tabulation* permet d'auto-compléter les noms de fichiers ou répertoires.

4. Taper l'instruction ci-après. Que fait-elle ?

```
1 tree
```

3.5 Manipuler des fichiers

Pour importer des fichiers dans cette simulation, il faut utiliser l'icône en bas de l'écran (figure 3).



FIGURE 3 – Importer un fichier dans la simulation

Activité 5 :

1. Importer une image dans le répertoire personnel.
2. Taper l'instruction ci-après (en adaptant le nom de l'image). L'instruction **mv** déplace un fichier ou un répertoire. Le `/` à la fin de *images* est important. Il indique que nous avons affaire à un répertoire.

```
1 mv mon_image.jpg images/
```

3. Se rendre dans le dossier *Images*.
4. Taper l'instruction ci-après (en adaptant le nom de l'image). L'instruction **cp** copie un fichier (ou un répertoire avec l'option -R).

```
1 cp mon_image.jpg copie.jpg
```

4 Les permissions

Chaque fichier ou répertoire appartient à un utilisateur. Il peut y avoir plusieurs utilisateurs pour une même machine et certains documents peuvent avoir un caractère privé alors que d'autres sont utilisables par plusieurs personnes.

Activité 6 :

1. Se rendre dans le dossier *images*.
2. Taper l'instruction ci-après.

```
1 ls -l
```

3. Lire le cours *openclassrooms* sur les droits d'accès :

<https://tinyurl.com/yy9xmpa5>

<https://openclassrooms.com/fr/courses/43538-reprenez-le-contrôle-a-l'aide-de-linux/39044-les-utilisateurs-et-les-droits#/id/r-39043>

4. Qui peut lire le fichier *mon_image.jpg* ?
5. Donner à tout le monde *le droit en lecture* de l'image.
6. Donner au groupe *le droit en lecture et écriture* du dossier *photos_vacances*.