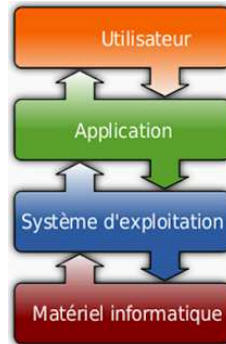


## Systèmes d'exploitation

### 1) Système d'exploitation (SE) - Operating System (OS)

Un **système d'exploitation** est un ensemble de programmes centraux d'un équipement informatique qui assure la liaison entre les ressources matérielles, l'utilisateur et les applications.



Lorsque l'on allume l'ordinateur, c'est le **BIOS** (fonctions de base permettant un démarrage de la machine) qui prend en charge le démarrage, puis **lance le système d'exploitation** qui prend le relais.

Celui-ci **tourne en permanence** sur l'ordinateur et **le contrôle** tant qu'il est allumé.

Le système d'exploitation **gère tous les composants de l'ordinateur** et **fournit une interface simplifiée avec le matériel** (on n'a pas à se préoccuper des détails de fonctionnement des ressources).

Il permet d'exécuter **simultanément plusieurs programmes**.

Le système d'exploitation gère :

- **le temps** : il permet de coordonner, d'ordonner l'utilisation du matériel entre les différents programmes d'application ; il accorde un certains temps pour l'exécution de chaque processus (programme en cours d'exécution) ;
- **l'espace mémoire** : il réserve de l'espace mémoire pour les besoins des programmes.
- **les interactions avec les périphériques d'entrées sorties** : il permet l'exploitation des périphériques, de recevoir les manipulations effectuées par l'utilisateur via le clavier, la souris... et les transmet aux différents programmes.
- **la persistance des données** : sauvegarder les données avec un système de fichiers et répertoires.

### 2) Systèmes libres ou propriétaires

Les **premiers systèmes d'exploitation** datent des années **1950**. L'interface avec l'utilisateur se fait en tapant des commandes sous la forme d'un texte.

Dans les années **1970**, est développé le système **UNIX**, par Bell Labs : c'est l'un des premiers systèmes multi-tâches et multi-utilisateurs.

**En 1977** l'université de Californie à Berkeley développe un système d'exploitation dérivé d'Unix, le **BSD**.

Des **années 1980** au début des années 1990, est développé le système **MS-DOS** de Microsoft. Il devient le système d'exploitation principal pour les particuliers et de nombreux domaines d'activités.

Dans les **années 1990** Microsoft développe un **système graphique** au dessus de MS-DOS, qui deviendra ensuite un système d'exploitation à part entière, le système **windows** et deviendra le plus utilisé sur PC.

En parallèle Linus Torvalds, alors étudiant à l'Université d'Helsinki, souhaite modifier le système d'exploitation MINIX (une variante de Unix). Bien que les sources de MINIX soient disponibles, la licence logicielle ne permet pas la diffusion des modifications.

Linus Torvalds décide donc de créer son propre système d'exploitation, le système d'exploitation **Linux**.

De **1991 à nos jours**, le système Linux connaît une adoption rapide, principalement due à sa diffusion sous **licence libre**. Il est particulièrement répandu comme système d'exploitation pour serveurs et machine de calcul ou de stockage.

En **2001**, Apple crée un nouveau système d'exploitation, basé sur le système BSD, le système **macOS**.

En **2007**, Apple reprend le cœur du système macOS et en développe une version pour les téléphones portables qu'il commercialise, **iOS**.

En **2008**, Google diffuse le système d'exploitation **Android** pour téléphones mobiles. Ce système utilise le noyau du système Linux, auquel sont ajoutés des programmes et bibliothèques non libres.



*Remarque :*

Un **système d'exploitation libre** est composé de **logiciels libres** gérés par un contrat de licence autorisant une utilisation **sans restrictions** ainsi que la **publication du code source**, sa **modification**, et sa **redistribution**. (Ex : GNU/Linux qui comporte différentes distributions comme Ubuntu, Debian, Mint...)

Par opposition aux systèmes libres, les **systèmes propriétaires** ne sont ni gratuits, ni voués à livrer leur code source. (Ex : Windows, MacOS...)

Outre la **gratuité ou non**, les différences sont notables entre système libre ou propriétaire. Un système libre comme GNU/Linux favorise la possibilité de **gérer la machine comme on l'entend**. La plupart des appareils vendus sur le marché sont équipés d'un système d'exploitation propriétaire (qui nous est vendu avec).

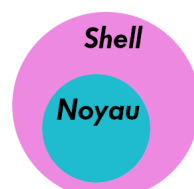
### 3) Interfaces homme-machine

Le **noyau** (ou **kernel**) est un **espace mémoire isolé et protégé**, dans lequel est placé tout ou partie des programmes du système d'exploitation.

Le noyau offre ses fonctions (l'accès aux ressources qu'il gère) au travers des **appels système**.

De manière générale, on désigne par **Shell** l'interface système permettant la **communication entre l'utilisateur et le noyau**. De nos jours, **Bash** est le shell le plus répandu.

**Utilisateur**

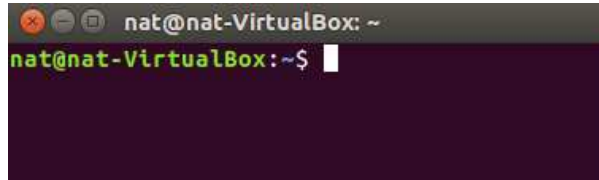


Historiquement les systèmes d'exploitations disposent d'une interface en **lignes de commandes** (en mode texte).

La plupart des systèmes d'exploitation modernes disposent en plus d'une **interface graphique** conviviale et accessible à tous.

Mais, maîtriser les lignes de commandes permet de disposer d'outils puissants de **configuration** et de **gestion de sa machine**. Cela donne la possibilité d'automatiser un certain nombre de tâches et cela permet également la prise de contrôle à distance d'une machine... L'interface par lignes de commandes, quand on sait s'en servir est **plus rapide**, permet de faire **davantage de choses** et est **très efficace**.

On utilise de nos jours un programme graphique appelé **terminal** ou **console**, pour écrire des lignes de commandes.



## 4) Système de fichiers : droits et permissions

### a) Système de gestion de fichiers

Le **système de gestion de fichiers** est une partie du système d'exploitation.

C'est une **structure de données** permettant de **stocker les informations** et de les **organiser** dans des fichiers sur des mémoires secondaires : disque dur, clé USB, SSD... Ce système **permet de localiser** ces fichiers à partir d'un **chemin d'accès**.

Les fichiers peuvent être lus, modifiés, renommés, copiés et supprimés.

Certains fichiers spéciaux, appelés **répertoires ou dossiers** ont pour seul but de *contenir* d'autres fichiers et répertoires.

A tout fichier on associe un ensemble de **Métadonnées** : son type, son propriétaire, les droits d'accès au fichier, sa taille, la date de sa dernière modification...

### b) Standards POSIX

Malgré une grande diversité des systèmes d'exploitations, il existe un ensemble de standards, regroupés sous le nom de **POSIX** (Portable Operating system Interface).

La plupart des systèmes d'exploitation sont compatibles avec le standard POSIX, à l'exception de Windows10 de Microsoft qui n'est pas compatible.

Cela peut s'expliquer historiquement par le fait que le standard POSIX est largement inspiré par le système d'exploitation Unix. Linux, Android, macOS ou iOS dérivant d'Unix sont compatibles, alors que le système Windows n'est pas un dérivé d'Unix, mais de MS-DOS.

Les systèmes POSIX sont **multi-utilisateurs** : chaque utilisateur possède un **identifiant de connexion** (login), auquel est associé en général un **mot de passe** qui permet de l'identifier. À l'identifiant de connexion correspond un identifiant numérique unique nommé **UID**.

Les utilisateurs peuvent être réunis **en groupes**. Un utilisateur appartient à un **groupe principal** et à des **groupes secondaires**. L'identifiant numérique du groupe est nommé **GID**.

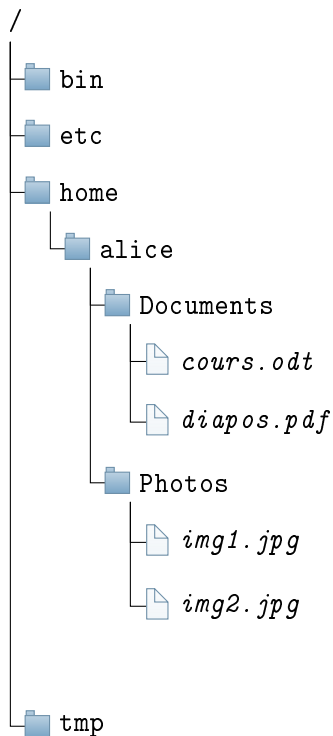
Il existe un utilisateur spécial nommé **root**, c'est le **super-utilisateur** : il correspond à l'**administrateur système** et peut modifier le système à sa guise. Son **UID** et son **GID** valent 0.

*Dans toute la suite, nous utiliserons des fonctionnalités POSIX.*

### c) Arborescence des fichiers

Le système de gestion de fichiers se représente généralement comme une **structure arborescente**, dont l'origine est la **racine**, noté « / ». C'est un répertoire contenant tous les autres.

Voici un exemple d'arborescence :



Dans cet exemple, la racine `/` contient les répertoires **bin**, **etc**, **home** et **tmp** ; le répertoire **home** contient le sous-répertoire **alice**... Le fichier *cours.odt* est dans le répertoire **Documents**.

Le **chemin d'accès** est une chaîne constituée de noms de fichiers ou de répertoires, séparés par le symbole « `/` ».

*Exemple* : Le **chemin d'accès absolu**, commence par le symbole « `/` » : il indique l'ensemble des répertoires à traverser pour accéder au fichier, en partant de la racine.

Pour le fichier *img1.jpg*, le chemin absolu est `/home/alice/Photos/img1.jpg`

Si on est dans le répertoire **alice**, que l'on appelle le **répertoire courant** , le **chemin d'accès relatif** au fichier est le chemin à partir du répertoire **alice** . Pour le fichier *img1.jpg*, le chemin relatif est `Photos/img1.jpg`

## 6) Virtualisation

La **virtualisation** est une technique pour créer, à partir d'une **machine physique** (la **machine hôte**), **une ou plusieurs machines virtuelles (VM)** isolées de l'hôte, et de les faire fonctionner simultanément.

Avec les VM on peut installer et utiliser **plusieurs systèmes d'exploitation** sur une même machine pour utiliser leurs fonctionnalités spécifiques, **tester des applications** de manière **sécurisées** en sauvegardant l'état de la machine avant un changement, ce qui permet de restaurer cet état si nécessaire.

Pour cela on utilise un logiciel appelé **hyperviseur** qui va remplacer ou travailler en collaboration avec le **système d'exploitation** de l'hôte. Il va créer les machines virtuelles (**machines invitées**), assurer l'isolation de ces machines du reste du système, puis contrôler et gérer les ressources matérielles de la machine hôte pour les distribuer aux différentes machines virtuelles.

On peut par exemple utiliser le logiciel **virtualbox** qui est simple à mettre en oeuvre et accepte plusieurs systèmes d'exploitation invités.



## 5) Terminal et lignes de commandes avec Bash

Les commandes sont **en abrégés** (de l'Anglais) ; elles **ne changent pas** (depuis les années 60). On les utilise **depuis une console ou un terminal**.

Lorsque l'on ouvre un terminal, une **invite de commande** apparaît sous la forme :

**pseudo@nom-machine :repertoire-courant\$**

Le symbole **~** indique que vous êtes dans votre dossier personnel, dans le home.

Le symbole **\$** indique que vous êtes un utilisateur normal, et le symbole **#** indique que vous êtes connecté en tant que super-utilisateur (root).

*Remarque* : Le langage Bash est **sensible à la casse**, c'est à dire qu'il fait une différence entre les lettres minuscules et les majuscules.

### Activité 1 : Fonctionnement du terminal

1. Sur vos postes de travail a été installé une machine virtuelle, ayant pour système d'exploitation une distribution linux Xubuntu que vous allez utiliser.  
Rendez-vous dans le menu Démarrer, puis dans le dossier Informatique, et sélectionnez VMWork. (Les identifiants vous seront précisés en classe.)
2. Commencez par explorer le bureau pour découvrir son organisation, puis ouvrez un **terminal** en le cherchant dans la barre de recherche, ou en tapant **CTRL+Alt+T**.
3. Quel est le nom de votre machine ? Votre nom d'utilisateur ? Quel est le répertoire courant dans lequel vous êtes placé ?
4. Tapez la commande **ls**, puis validez.


*Vous avez obtenu la liste des fichiers et dossiers du répertoire dans lequel vous vous trouvez.*

5. La plupart du temps, on ajoute aux commandes des options, les **paramètres**, séparés par un espace.  
*Ceux-ci peuvent s'écrire avec une seule lettre, précédée d'un tiret, comme par exemple **ls -a** ;  
ou bien avec plusieurs lettres, précédées de deux tirets comme par exemple **ls -all**.*

Testez ces deux commandes dans votre terminal.

*Ce paramètre ajoute les fichiers cachés (commençant par un point à liste précédente.*

6. Un moyen pratique pour éviter d'écrire le nom complet d'un dossier ou d'un fichier est l'**auto-complétion**.  
*Il suffit de taper les premières lettres, puis d'appuyer sur la touche tab (tabulation), pour que le mot soit complété.*

Essayez en tapant "Do", puis la touche tab .

7. Pour éviter de réécrire une commande déjà utilisée lors de la session, on peut utiliser les flèches (haut et bas) du clavier pour retrouver ce qui a été tapé précédemment.

Relancez la commande **ls -a**, sans la taper.

8. Si vous vous retrouvez bloqué dans le terminal, vous pouvez taper **CTRL+Z**.
9. Pour fermer le terminal, on peut taper **exit**, ou bien **CTRL+D**.

Fermez le terminal.

### Activité 2 : Commandes de base

1. Ouvrez un terminal et tapez la commande **pwd**.  
Qu'obtenez-vous ?

2. Pour se déplacer dans l'arborescence des fichiers, on utilise la commande **cd**, suivie du nom du répertoire dans lequel on veut se rendre ou du chemin (relatif ou absolu) pour s'y rendre.

Tapez **cd Bureau**. Vous voilà dans le répertoire Bureau. Quel est son chemin absolu ?

Puis tapez **cd ..**. Que s'est-il passé ? Vérifiez en tapant **pwd**.

Pour revenir directement dans votre dossier personnel il suffit de taper **cd ~** ou tout simplement **cd**.

Remontez jusqu'à la racine / en utilisant la commande **cd ..** autant de fois que nécessaire, puis tapez **ls** pour observer se qu'il s'y trouve.

Enfin, retournez directement dans **votre Bureau**, en tapant le chemin absolu.

3. En tapant **help**, vous avez une liste de commandes disponibles qui s'affichent. En tapant **man** suivi du nom d'un commande, vous en obtenez une description détaillée. En effet on y précise toutes les options associées.

Tapez **help**, puis tapez **man date**. Que permet de faire la commande date ? Vérifiez.

4. Tapez **touch fichieressai**, puis tapez **ls**.

Vous venez de créer un fichier vide dont le nom est fichieressai

5. Tapez **mkdir nouveaudossier**, puis tapez **ls**.

Vous venez de créer un dossier.

6. On peut utiliser un éditeur de texte simple directement en mode console. **Nano** est installé par défaut.

Lancez-le en tapant **nano fichieressai**.

Vous pouvez immédiatement taper du texte ; en bas de l'écran des raccourcis indiquent comment sauter une ligne, quitter l'éditeur... Le symbole **^** veut dire **CTRL**.

Tapez deux ou trois lignes dans **fichieressai**, puis quittez l'éditeur nano en sauvegardant.

Faites apparaître le contenu du fichier avec **cat fichieressai** pour visualiser vos modifications.

On peut observer le contenu d'un fichier avec la commande **cat** (qui affiche tout et la commande **less** qui l'affiche page par page.

7. On peut copier un fichier avec la commande **cp nom-fichier nom-copie**.

Copiez fichieressai et nommez la copie **copiefichier**. Affichez le contenu de copiefichier.

8. On peut déplacer un fichier en tapant **mv nom-fichier nouvel-emplacement**.

Déplacez **copiefichier** dans votre **Bureau**. Rendez-vous dans votre Bureau pour vérifier la présence du fichier déplacé.

9. On peut effacer un fichier ou un dossier avec **rm nom du fichier**.

**ATTENTION**, il n'y a pas de corbeille dans la console, la suppression sera définitive et sans avertissement !

Supprimez le fichier **copiefichier** et affichez le contenu du **Bureau** pour vérifier sa disparition.

Pour supprimer un dossier on tape **rm -r nom-dossier**.

### Activité 3 : Droits et permissions

1. Ouvrez un terminal et tapez la commande **id**.

Vous obtenez notamment votre **UID** et votre **GID**, vos **identifiants numériques** individuel et de groupe.

2. En tapant la commande **ls -l**, vous obtenez la liste détaillée des éléments du dossier courant, avec dans l'ordre : les droits du fichier, le nombre de liens, le nom du propriétaire du dossier, les groupes auxquels appartient celui-ci, sa taille en octets, la date de sa dernière modification et enfin son nom.

Exemple : Voici un extrait d'un terminal bash :

```
drwxr-xr-x  3 nat  amis   96 16 août 17:05 elements
-rw-r--r--  1 nat  famille 2872 17 août 16:59 fiche2.tex
```

Les droits et permissions du fichier **element** sont **drwxr-xr-x** ; il y a 3 liens vers ce fichier, son propriétaire est nat ; il appartient au groupe amis, fait 96 octets et a été modifié pour la dernière fois le 16 Août à 17h05. Décrivez ce qu'indique la deuxième ligne de l'exemple.

3. Parmi les permissions, le **premier caractère** est soit « **d** » pour indiquer que c'est un **dossier**, soit « **-** » si ce n'est pas le cas.

Les neuf caractères suivants sont à lire **par groupe de trois**.

Un groupe représente les droits en **lecture (r)**, en **écriture (w)** et **exécution (x)**. Si une lettre est affichée, la **permission est accordée**. Si un « **-** » est affiché, la **permission est refusée**.

Le premier groupe de trois caractères représente les permissions pour le **propriétaire**, le groupe suivant les permissions pour le **groupe propriétaire**, et le dernier groupe de permissions pour **tous les autres**.

Affichez la liste détaillée des du contenu de votre dossier personnel.

Précisez les permissions accordées pour votre dossier **Documents**.

Le « **x** » d'exécuter pour un dossier veut dire que l'on peut le traverser et observer ce qu'il contient.

Le super-utilisateur a toujours toutes les permissions !

4. Le propriétaire d'un fichier peut en modifier les permissions avec la commande **chmod**.

Les symboles liés aux utilisateurs sont « **u** » pour le propriétaire, « **g** » pour le groupe lié au fichier (ou au répertoire), « **o** » pour les autres utilisateurs et « **a** » correspond à tout le monde.

Les symboles liés à l'ajout ou la suppression des permissions sont « **+** » pour ajouter une permission, et « **-** » pour la supprimer. Avec « **=** » on affecte les permissions.

Par exemple **chmod o+w toto.txt**, veut dire que l'on ajoute la permission d'écriture au autres utilisateurs pour le fichier toto.txt.

Et **chmod g-wx action** enlève les permissions d'écriture et d'exécution au fichier action.

Créez un fichier appelé **fichierdroits** dans le dossier **Documents**. Puis, affichez les permissions de ce fichier.

Changez les permissions de ce fichier pour que tous les utilisateurs puissent écrire dessus, puis retirez le droit de lecture au groupe propriétaire. Vérifiez vos changements.

## Activité 4 : Compléments

1. Filtrer des données :

La commande **grep** permet de rechercher un mot dans un fichier et d'afficher les lignes dans lesquelles ce mot a été trouvé.

Ouvrez un terminal et créez un fichier **texte**. Éditez le avec **nano**, en y incorporant plusieurs fois le mot **magie**.

Fermez l'éditeur, puis tapez **grep magie texte**.

2. Compresser et décompresser des fichiers :

Pour envoyer un ou plusieurs fichiers par le réseau, il est toujours préférable de commencer par les compresser afin de réduire leur taille. Il y a pour cela plusieurs méthodes.

Le format **.zip** est le plus connu et le plus répandu sous Windows. On peut l'utiliser aussi sous Linux, mais il n'est pas toujours installé par défaut. La commande **unzip** permet de décompresser un **.zip**.

Cependant, il existe des alternatives libres (et souvent plus puissantes). On utilise par exemple deux outils : l'un qui permet d'assembler des fichiers avant de les compresser (**tar**), puis l'autre pour la compression (**gzip**).

La commande **tar -zcvf Essai.tar.gz Essai** permet d'archiver et de compresser en même temps le dossier **Essai**.

La commande **`tar -zxvf Essai.tar.gz`** va décompresser le dossier **`Essai.tar.gz`**.

**`-c`** signifie créer une archive tar ; **`-v`** : signifie afficher le détail des opérations ; **`-f`** : signifie assembler l'archive dans un fichier ; **`-x`** signifie extraction des fichiers.

Quelle est la taille du dossier **Documents** ?

Compressez le dossier Documents avec les commandes précédentes. Quelle est la taille du dossier Documents sous forme compressée ?

Déplacez le fichier compressé dans la dossier **Bureau**, puis décompressez-le et vérifiez son contenu et sa taille. Enfin, supprimez le fichier compressé.

### 3. Les paquets (ou packages)

*Sous Windows il y a des fichiers .exe qui permettent l'installation de nouveau programmes.*

*Sous certaines distributions Linux, on a ce qu'on appelle des **paquets** (packages). Un paquet est une sorte de dossier zippé qui contient tous les fichiers du programme et toutes les instructions nécessaires pour installer ce programme.*

*Tous les paquets sont rassemblés au même endroit sur un même serveur appelé **dépôt** (repository).*

*Pour pouvoir installer ou désinstaller un programme, il faut se mettre en mode **super-utilisateur** (root).*

*Pour cela, on tape **sudo** devant les commandes et on donne le **mot de passe** correspondant.*

*Il faut de temps en temps mettre à jour la liste des paquets proposés par le dépôt, en tapant **apt-get update**.*

Commencez par taper **sudo apt-get update**, puis renseignez le mot de passe demandé.

*Si l'on ne connaît pas le nom exact du paquet à installer, on tape **apt-cache search larecherche**.*

*Ensuite on installe le paquet en tapant **apt-get install nompaquet**.*

Tapez la commande **tree**. Si le paquet n'est pas encore installé, faites-le, puis lancez le programme en tapant **tree**. Que fait ce programme ?

Faites de même avec la paquet **cmatrix**.

*Pour désinstaller un paquet on utilise **apt-get autoremove nompaquet**.*

Désinstallez le paquet **cmatrix**. Vérifiez qu'il ne se lance plus.

*On peut mettre à jour tous les paquets installés sur le système d'un seul coup avec la commande **apt-get upgrade**, qui ira chercher les nouvelles versions de tous les programmes et les mettra à jour si une nouvelle version est disponible.*

### Activité 5 : Un petit jeu

Pour vous aider à mémoriser les commandes Bash de base, rendez-vous sur le site ci-dessous, et montrez-vous malin !

<http://luffah.xyz/bidules/Terminus/>

## 6) Travailler en lignes de commandes à la maison

- Les manipulations décrites précédemment sont réalisables sur les systèmes Linux ou macOS.
- En ce qui concerne les ordinateurs fonctionnant avec Windows 10, il est possible d'utiliser les lignes de commandes en installant une simulation d'un système POSIX, qui fournit un terminal et les commandes associées.

Pour cela, aidez-vous du tutoriel du site :

<https://korben.info/installer-shell-bash-linux-windows-10.html> (Seulement la première partie.)

- Vous pouvez aussi utiliser le site suivant, qui simule un terminal (sans identifiants) :

<http://weblinux.univ-reunion.fr/?cpu=asm&n=1>

- Enfin, vous pouvez installer le logiciel **virtualbox** (que vous trouverez sur internet) pour ensuite installer une **machine virtuelle** dont le système d'exploitation est Linux.

Vous trouverez une fiche d'aide sur Pearltree et je pourrai vous fournir un système d'exploitation à installer.