

Systemes d'Exploitation

1^{ère} année BUT Info Lannion

Gildas QUINIOU

- Enseignant Info BUT
- Développeur d'applications Web et Mobiles

1 - Introduction

Le cours de Systèmes a une **importance particulière**.

Pour un menuisier, une scie, un rabot sont des outils essentiels. Il les utilisera **toute sa vie professionnelle**.

Pour l'informaticien, le **Système d'Exploitation** est aussi un outil qu'il utilisera **toute sa vie professionnelle**.

Le Système d'Exploitation est un **élément fondamental** de tout ordinateur, qu'il soit sous une forme **traditionnelle** (Unité Centrale, clavier, écran) ou sous la forme plus **exotique** d'un **téléphone**, une **montre**, un **appareil photo** ou un **robot tondeuse à gazon**.

Tous ces équipements embarquent un **Système d'Exploitation** !

C'est une UE (Unité d'Enseignement) qui n'est **pas difficile**, sous réserve de :

- Être présent (faut-il le préciser ?)
- Actif
- Curieux
- Poser des questions

Ressource R1.04

- R : Ressource.
- Semestre 1.
- Ressource n° 04 dans le semestre.
- R1.04 s'appelle "Introduction aux Systèmes"

SAÉ S1.03

- S : SAÉ = Situation d'Apprentissage et d'Évaluation.
- Semestre 1.
- SAÉ n° 03 dans le semestre.
- S1.03 s'appelle "Installation d'un poste de développement"

Calendrier

- 5 semaines x 2H TP
 - 5 semaines x 2H TD (machine)
+ 4H de SAÉ encadrée + 2H de SAÉ autonome
 - 4 semaines x 2H TD (machine)
+ 4H de SAÉ encadrée + 5H de SAÉ autonome
- Total : 2H CM + 18H TD + 10H TP + 15H SAÉ = 45H.

Contrôles

- Contrôles continus en TP/TD.
- Contrôle TP en temps limité (**S02**, à confirmer).
- 2 DS (**S43** et **S03**, à confirmer).
- Contrôle de SAÉ (**S50**, à confirmer).

Vous aurez (en principe) droit à 1 feuille A4, recto/verso, **manuscrite et personnelle** pour les DS. Commencez la rédaction synthétique dès à présent, ça vous servira aussi de support/pense-bête en TP/TD !

2 - Qu'est-ce qu'un OS ?

Le terme **Système d'Exploitation** est généralement raccourci en **SE** ou **OS** (Operating System).

C'est le **cerveau logiciel** de l'ordinateur.

Un **chef d'orchestre** pour que le matériel et les logiciels fonctionnent en harmonie.

L'OS reçoit des **ordres** et les **exécute**.

L'OS est :

- Lancé au **démarrage** de l'ordinateur.
- **Prépare** l'ordinateur (composants et périphériques).
- Tourne jusqu'à l'**extinction** de l'ordinateur.

L'OS sert d'**intermédiaire** entre le matériel et les autres logiciels.

3 - Unix : Linux & Cie

Bref historique

Unix :

- Créé en 1969 par Ken Thompson, au laboratoire Bell de la société AT&T. Nom d'origine : Unics.
- Écrit en langage assembleur (maintenance complexe).
- Réécrit en 1971, nouveau langage, le C, par Ken Thompson et Dennis Ritchie (Papa du langage C).

Unix :

- 1969 (août) : Un peu la **préhistoire** de l'informatique.
- Conception incroyablement **moderne**.
- De nombreuses évolutions, mais des **bases toujours présentes**.
- Aujourd'hui : Un des OS **les plus utilisés**.
- Un OS mais aussi un **concept** d'OS.
- **Plusieurs Unix**, dérivés de l'Unix originel d'AT&T.
Exemple : Linux.

Linux :

- 1991.
- Œuvre d'un finlandais, **Linus Torvalds**.
- Clin d'œil à l'histoire, Linus est né en **1969** (comme Unix).

Aujourd'hui, Linux est :

- L'OS le plus présent sur les **serveurs**.
- De très loin, l'Unix **le plus populaire**.

Principes de base

Unix :

- Multitâche.
- Multi-utilisateur.

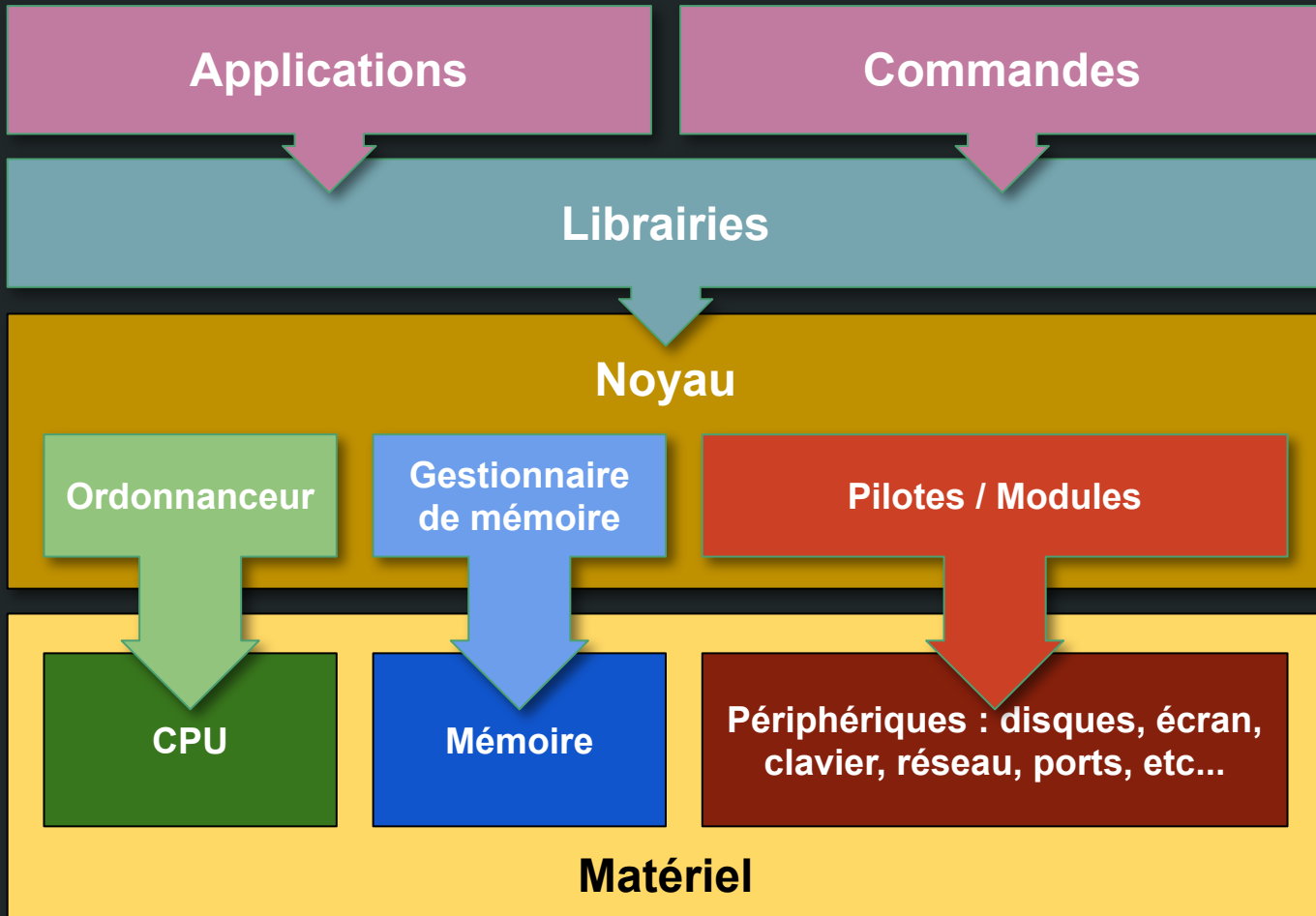
Des caractéristiques communes :

- Gestion des **fichiers**, des **ressources**, des **processus**...
- Jeu de **commandes de base**.

A partir de maintenant :

- On va se focaliser sur Linux.
- Tout ce qui sera dit dans ce cours sur Linux se rapporte aussi aux autres Unix en général.

4 - Linux



L'OS Linux c'est :

- **Noyau** (Kernel en anglais).
- **Pilotes** ou **Modules** (interface avec le matériel).

Par abus de langage, on considère aussi que les **commandes d'administration** font partie de l'OS.
Nous aussi dans ce cours !

5 - Que va-t-on apprendre ?

Plein de choses utiles et sympathiques...

- Se **connecter** et se **déconnecter** d'un ordinateur.
- Donner des **ordres**, exécuter des **commandes**.
- **Manipuler** les fichiers.
- Se **déplacer**, manipuler l'**arborescence**.
- Comprendre, manipuler les **droits** sur les objets.
- Agir sur les **processus**.
- Écrire des **scripts**.

6 - Les “objets” Linux

Les “objets” Linux que nous allons voir et/ou utiliser :

- Utilisateurs et Groupes.
- Fichiers et Dossiers.
- Processus.

Notes sur les utilisateurs :

- **root** (c'est le nom donné à Dieu en informatique).
- Des utilisateurs **fictifs** (contrôle de services, Serveur Web ou Serveur de Mail par exemple).

7 - Le login, le compte

Chaque utilisateur de l'OS possède un compte.

Un compte est constitué de :

- Un **nom** (unique).
- Un **mot de passe** (stocké crypté).
- Un **ID** (numéro unique).
- Un **home** (répertoire d'accueil).
- Un **shell** (optionnel).

Le login

De l'anglais “log” qui est le **journal de bord** sur un bateau, dans lequel le capitaine consigne les événements.

Quand un utilisateur se connecte, cet événement est aussi consigné dans un **fichier d'événements**. Ce qui a donné l'expression “**to log in**” et finalement le **login**.

Le login est le nom que l'utilisateur a pour **se connecter**, pour se "**logger**".

Ce nom est une simple commodité. Une fois l'utilisateur loggé, l'OS utilise ensuite uniquement son **ID** (valeur numérique). Ainsi, on peut **changer** son nom plus tard sans rien chambouler pour l'OS.

Le **logout** est l'action de se déconnecter de l'ordinateur.

Le mot de passe

A la connexion de l'utilisateur, le mot de passe qu'il fournit est vérifié soit **localement**, soit en utilisant un **annuaire** sur un autre serveur du réseau.

Le mot de passe est toujours stocké **chiffré**. Il n'y a aucun moyen de connaître le mot de passe d'un utilisateur.

Le home

Chaque utilisateur a un répertoire d'accueil sur le disque de l'ordinateur. Un espace où il peut stocker ses fichiers de travail.

On l'appelle le “**home directory**” ou simplement “**home**”.

Le shell

En anglais “shell” signifie **coquille** (pour un crustacé) ou **coque** (pour un fruit). Et qu’y a-t-il dans une coque de fruit ? Il y a le cœur du fruit et son noyau.

Noyau... Kernel ! Et le message pourrait donc être :

Qui a accès au shell a accès au kernel

Si le compte de l'utilisateur n'est **pas associé** à un shell (ou un programme similaire), alors il ne pourra **pas se connecter**.

Une fois connecté, l'OS est au service de l'utilisateur :

- L'utilisateur donne des **ordres** via le shell.
- L'OS les **exécute**.

Tout ceci sous réserve que l'utilisateur dispose des **droits nécessaires** pour l'ordre qu'il donne.

8 - Le shell

Le rôle du shell est de servir d'**interface** entre l'utilisateur (via son clavier) et l'OS.

Un shell :

- Affiche un **prompt**. Exemple : **username \$ _**
- Lit des **ordres** (généralement) au clavier, terminés par l'appui sur la touche **ENTRÉE**.
- **Passe la main** à l'OS pour exécuter l'ordre.
- **Affiche le résultat**, généralement à l'écran.
- Retourne à l'affichage du **prompt**.

Attention ! Warning ! 警告 ! Achtung ! تحذير ! Attenzione !

**On n'est pas sous Windows.
Le shell ne va pas demander de
confirmations.
Vous ordonnez, il exécute.
C'est tout !**

9 - Les fichiers

Un fichier est :

- Une zone allouée généralement sur un **espace de stockage** (disque dur, clé USB, Cloud).
- Identifié par un **nom**.
- Contient des **données quelconques** (image, texte, son, vidéo).

On trouve aussi le nom de : **document**.

Un fichier peut ne rien contenir, on dit qu'il est **vide**.

Un fichier peut :

- Se suffire à lui même : **lisible directement**.
Exemple : fichier **texte**.
- Nécessiter un **logiciel** pour être exploitable.
Exemple : fichier (document) **Word** ou fichier **Excel**.

Le nom de fichier

Un fichier est identifié par son nom, tel que **logo.png**, composé de :

- Un **nom de base** : “logo”.
- Une **extension** : “.png” (commence toujours par “.”).

Chacune de ces 2 parties est **facultative** mais il en faut **au moins une** des deux pour former un nom de fichier.

Exemple : “**logo**” ou même “**.png**” tout seul.

10 - Les commandes

Comprendre les noms des commandes

Avant tout, il faut se rappeler qu'Unix :

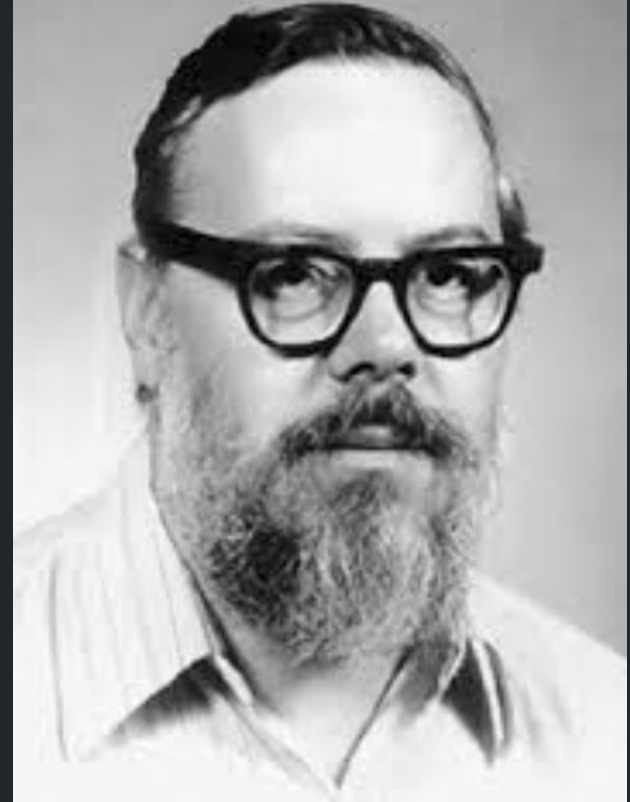
- Date des **années 70**.
- A été conçu par des **barbus** pour des barbus.



Ken THOMPSON

Le Papa d'Unix.





Dennis RITCHIE

Le Papa du langage C.





Richard STALLMAN

Initiateur et promoteur du
Logiciel Libre (dont fait
partie Linux).
Le Papa du projet “GNU”.



Attention aux contrefaçons...

... certains barbus n'ont rien à voir avec l'informatique !

Attention aux contrefaçons...

Le Papa Noël.



A cette époque, un non barbu ne pouvait pas comprendre Unix !

Bref, c'était une époque où ils (les barbus) ne se souciaient pas trop de rendre leur joujou accessible aux étudiants de 1ère année de l'IUT de Lannion !

Tout ceci a donc un **impact sur le nommage des commandes**, comme on va le voir.

Les commandes que je vais citer dans cette rubrique sont là uniquement pour illustrer les propos.

Ce n'est pas la peine de retenir leur fonction ici, on va revenir en détail sur chacune, plus loin.

Les commandes de base portent des noms pas très évidents à comprendre... si on ne dispose pas des clés.

Voici donc quelques **astuces**.

Globalement, les commandes sont des versions **raccourcies** d'un ou de plusieurs mots.

Certaines commandes de base ont des noms sur **2 lettres** :

- cp : **copy**
- mv : **move**
- ls : **list**
- rm : **remove**

Ce sont des mots simples, ils ont simplement gardé les **2 premières consonnes**.

Ensuite on trouve des commandes qui sont la combinaison ou les initiales de **plusieurs mots** comme :

- cd : **c**hange **d**irectory
- pwd : **p**rint **w**orking **d**irectory
- mkdir : **m**ake **d**irectory
- rmdir : **r**emove **d**irectory
- man : **m**anual
- wc : **w**ord **c**ount
- top : **t**able **o**f **p**rocesses
- vi : **v**isual **i**nstrument
- vim : **v**i **i**mproved
- grep : **g**lobally search **r**egular **e**xpression and **p**rint
- awk : **A**ho + **W**einberger + **K**ernighan (3 créateurs)

On a aussi des **mots courts** pour lesquels on conserve toutes les lettres :

- date : **date+heure** courante.
- who : **qui** est connecté.
- head : début (**tête**) d'un fichier.
- tail : fin (**queue**) d'un fichier.
- tree : arborescence (**arbre**) de fichiers.

Il n'y a pas de règle unique sur le nommage des commandes, mais il faut chercher la **signification** derrière les quelques lettres, ça aide à comprendre et à retenir le rôle de chacune.

Se tromper dans une commande parce qu'on a confondu **rm** (qui supprime des fichiers) et **cd**, ça peut **faire mal**. La bonne nouvelle c'est que ça ne fait mal qu'une seule fois, après on s'en rappelle.

Format d'une commande

Une commande est composée de plusieurs parties :

- **Nom** de la commande.
- Une ou plusieurs **options** (parfois facultatives).
- Des **paramètres** (parfois facultatifs).

Exemple avec `ls -l toto.png` :

- `ls` est la **commande**. Elle affiche des informations sur des fichiers.
- `-l` est une **option**. Une option permet de modifier le comportement standard de la commande. Ici cette option permet d'afficher des informations plus détaillées (`l` = **long**).
- `toto.png` est un **paramètre**. Ici c'est le nom du fichier qu'on cible.

Options d'une commande

Les options peuvent être de 2 formes :

- Courte : un tiret (-) suivi d'une lettre.
Exemple : **-a**, **-l**, **-n**
- Longue : 2 tirets (--) suivis d'un mot.
Exemple : **--inode** ou **--help**

Quand on a besoin de mettre **plusieurs options** courtes, il est possible de les coller ensemble derrière **UN SEUL** tiret.

Exemple :

ls -l -a

devient

ls -la

Quelques mots sur les sujets de TP/TD

- 1ers sujets verbeux, genre “tutos”
- Puis sujets moins guidés
- Prendre des notes (Markdown ou autre)
- Poser des questions
- Ne pas attendre le passage de l’enseignant
- Ne pas se satisfaire de copier-coller-exécuter

- Vidéos YT avec complément de cours : optionnel mais utile... À consulter hors séance.



That's all Folks!

https://commons.wikimedia.org/wiki/File:Thats_all_folks.svg