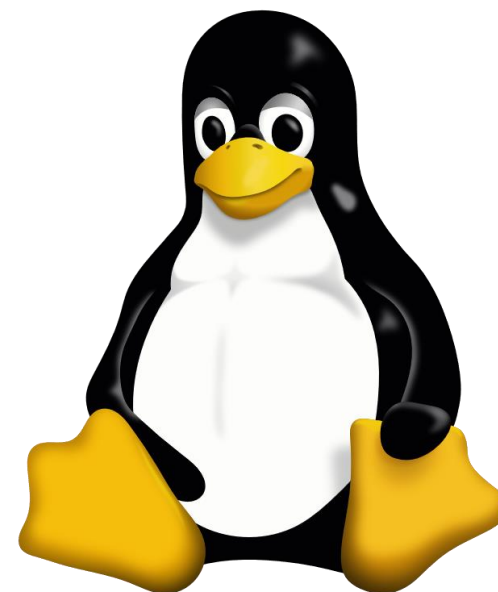


Introduction à Unix - GNU/Linux

Dr Nils Beaussé

nils.beausse@isen-ouest.yncrea.fr

Bureau au 2ème étage (ou voir en salle recherche A1-50)



Déroulement du cours

- Cours + TP 1 - 3hrs
- Cours + TP 2 - 3hrs
- TP 3 - 3hrs
- TP 4 - 3hrs
- TP 5 - 3hrs
- **Évaluation** - 2hrs

Objectif du cours

- Une introduction rapide au
 - Système Linux
 - Système d'utilisateur Linux
 - Système de fichier Linux
 - À l'environnement Shell
- Beaucoup de point du cours seront ensuite introduits dans les « labs » : des TPs avec de nombreux points de cours introduit en contexte.



Une courte histoire d'Unix et GNU/Linux

- Projet Multics puis Unics et enfin Unix en 1969 :
 - Écrit par une équipe des Laboratoires AT&T Bell (Ken Thomson, Dennis Ritchie, Rudd Canaday, etc), qui ont été à l'origine de nombreux progrès important du début de l'informatique.
 - Pour des raisons juridiques, AT&T ne pouvait pas vendre ce système, il l'a donc peu à peu passé sous la férule des universités.
 - Unix est un **noyau (kernel) de système d'exploitation** !



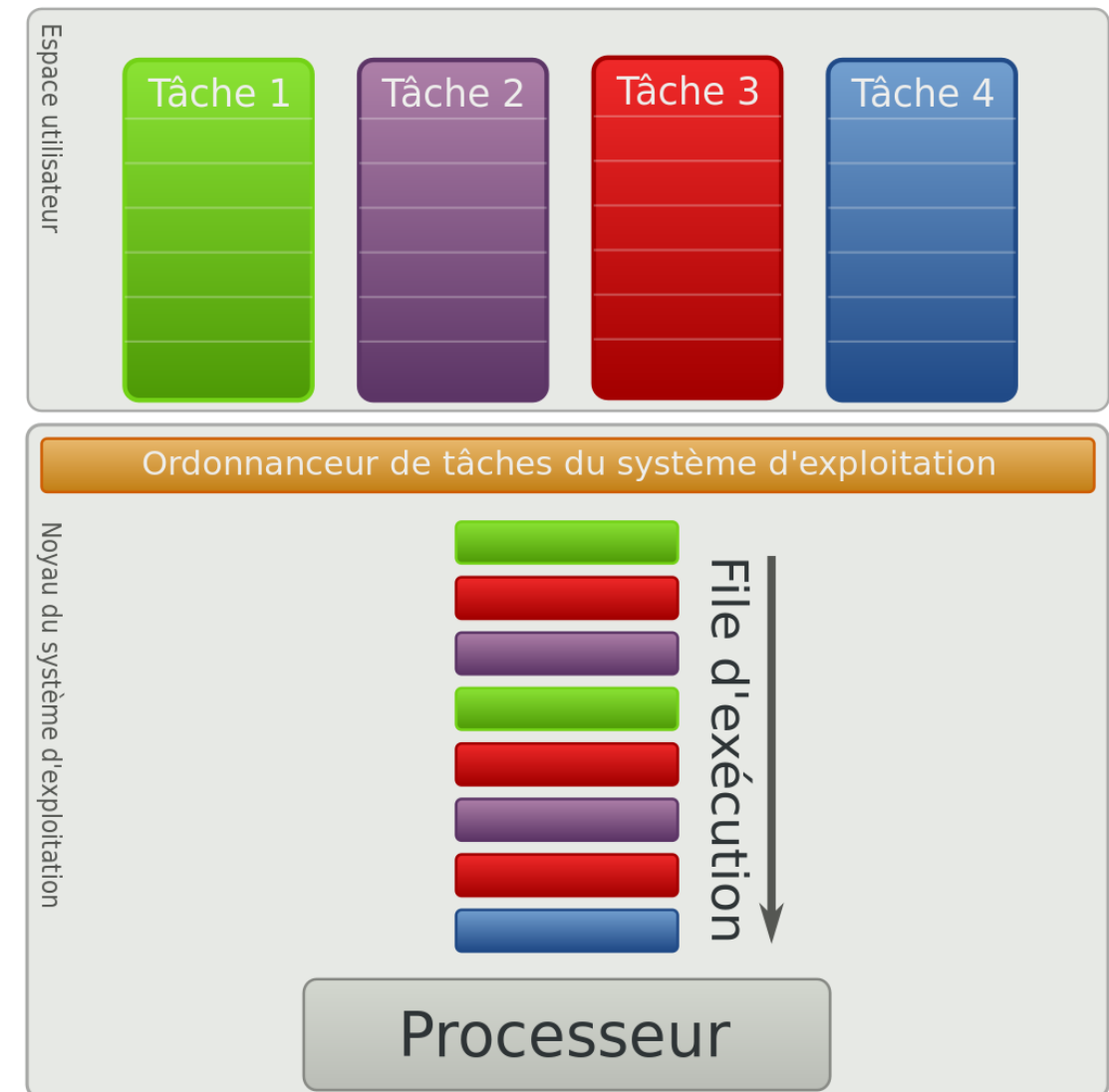
Aparté : Noyau de système d'exploitation ?

- En informatique, le noyau d'un système d'exploitation est le logiciel qui assure :
 - la communication entre les logiciels et le matériel ;
 - la gestion des divers logiciels d'une machine (lancement des programmes, ordonnancement...) ;
 - la gestion du matériel (mémoire, processeur, périphérique, stockage...).

Aparté : Noyau de système d'exploitation ?

- Exemple de fonctionnalité du noyau :

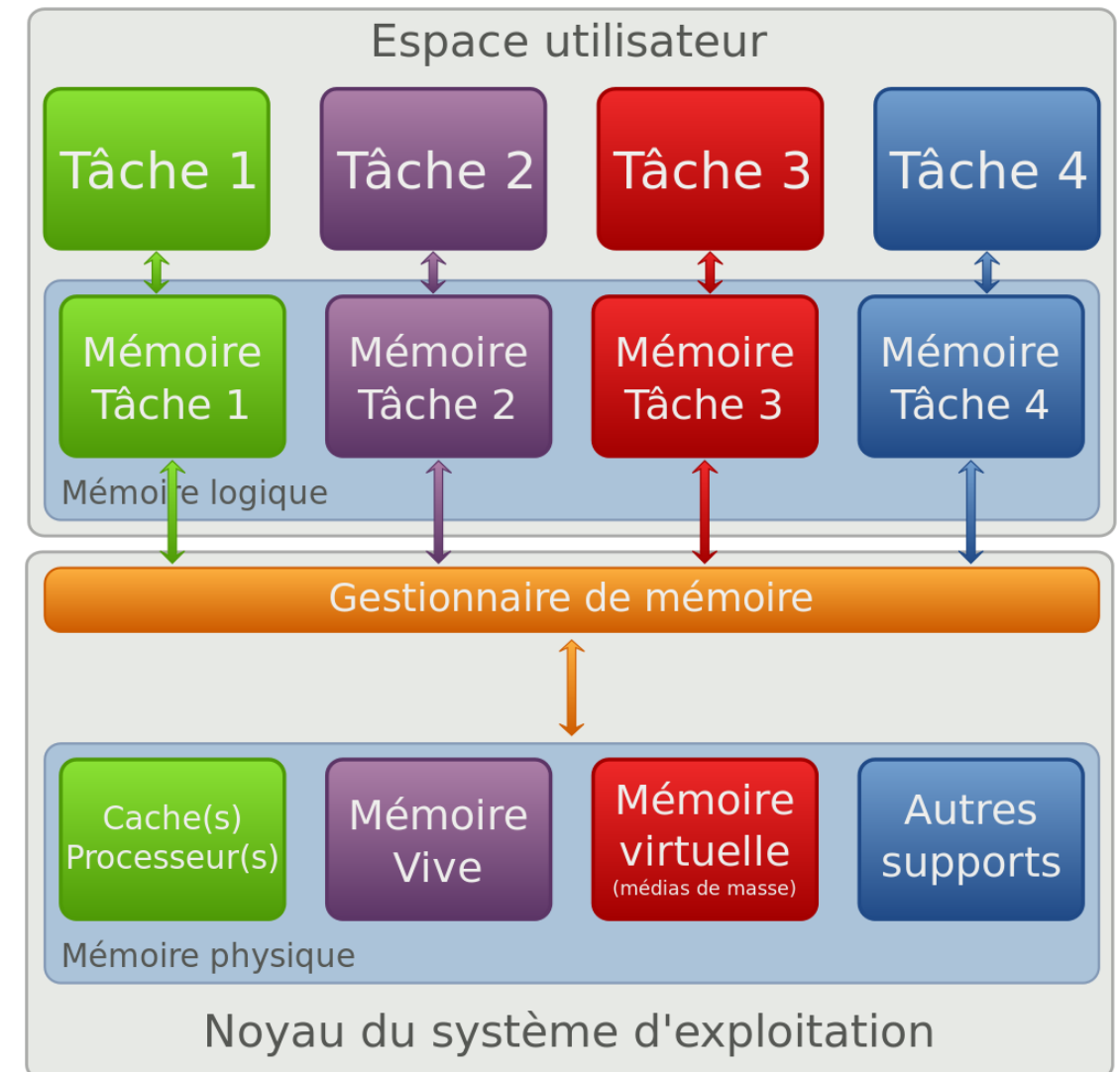
L'ordonnanceur



Aparté : Noyau de système d'exploitation ?

- Exemple de fonctionnalité du noyau :

La gestion mémoire



Une courte histoire d'Unix et GNU/Linux

- 1971: Portage d'Unix sur un PDP-11 (24 KB de RAM : 8 pour les utilisateurs, lecteur de disque de 512 KB)
- 1973: **Le langage B, puis C est inventé pour réécrire Unix**, Unix est réécrit en C



- 1974: La License UNIX est transférée pour des usages académiques aux universités.

C'est à partir de cette époque que les étudiants et les jeunes entreprises liées aux universités se mettent à disposer d'un outil très pratique qui va ouvrir la voie à ce que l'informatique est aujourd'hui.

Une courte histoire d'Unix et GNU/Linux

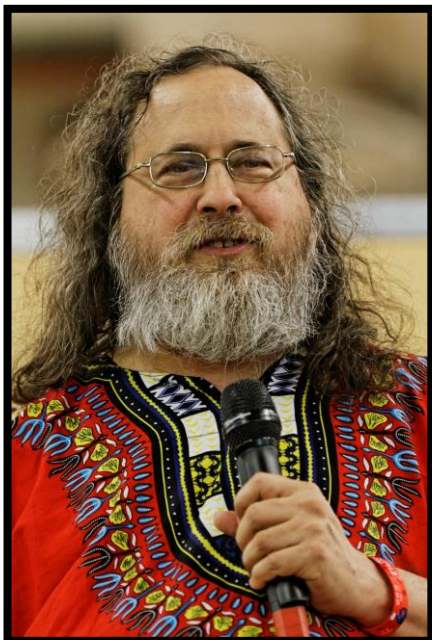
- Depuis 1974: Normalisation et évolution.
- 1979 : Version 7 d'Unix, portabilité !
- **Naissance de BSD** (Berkeley Software Distribution) qui devient l'une des variantes universitaires majeures d'Unix.

BSD

- BSD est attaqué en justice dans les années 90 par la société AT&T d'origine. Les compromis du procès permettront de sortir à terme des versions libres d'Unix, FreeBSD notamment. Mais cela mettra du temps.

Une courte histoire d'Unix et GNU/Linux

- Richard Stallman, l'un des fondateurs du **logiciel libre**, décide, face au fait que Unix reste peu libre en 1980, de construire le système **GNU** (**GNU's Not UNIX**).



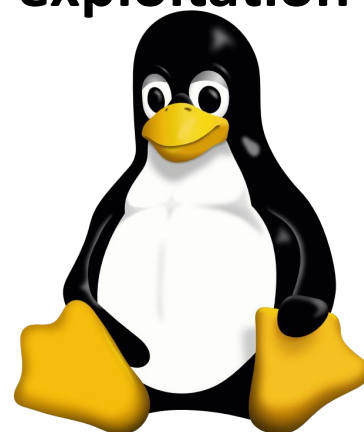
- **GNU** est construit comme un système entièrement libre et construit par une communauté de programmeur (que n'importe qui peut rejoindre).



Une courte histoire d'Unix et GNU/Linux

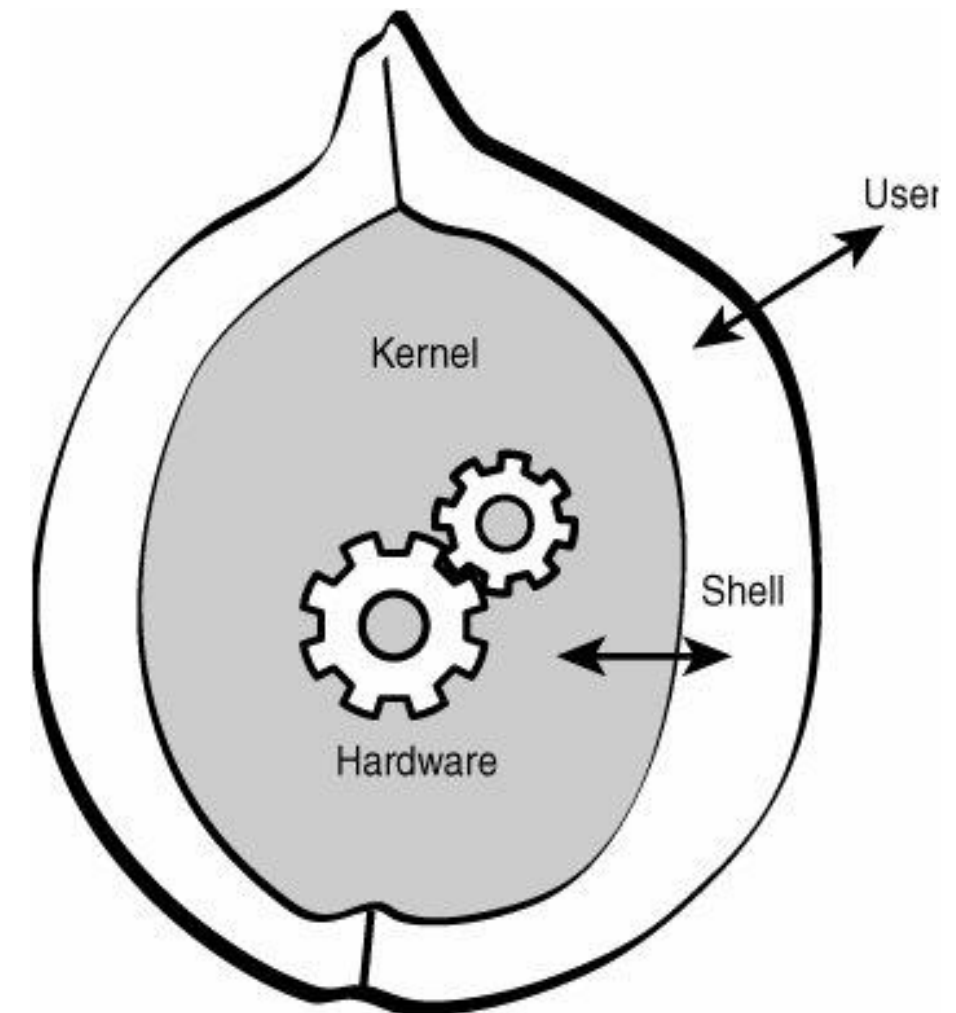
- En parallèle, en 1990, **Linux Torvald**, étudiant à l'université d'Helsinki, est frustré par les limitations d'Unix (rappelons qu'Unix est limité par le fait de n'être que peu libre à l'époque). Il réécrit peu à peu une sorte d'Unix à part, qui n'a aucun code d'Unix mais fonctionne pareil, au début comme un jeu, mais il finit par le publier sous **licence libre GNU GPL** (la licence créée par Richard Stallman) en appelant à des participations des programmeurs extérieurs.
 - Il aboutit alors au noyau d'un système d'exploitation : [Le noyau Linux.](#) (linux = jeu de mot entre Linus et Unix)

Tux, le symbole de Linux



Aparté : Noyau, Coquille, Distribution

- On a vu ce qu'était un **noyau**, c'est lui qui gère la communication entre le matériel et les autres programmes.
- Mais un noyau seul n'a pas beaucoup d'usage, il faut pouvoir communiquer avec lui. En somme il lui faut tout un ensemble de programmes qui viennent par-dessus. La première couche qui permet de communiquer avec le noyau est souvent dénommée la coquille ([Le Shell](#)).



Aparté : Noyau, Coquille, Distribution

- Le Noyau + le Shell forme un système très minimaliste. Par exemple il ne contiendra pas de programme pour éditer du texte ou de programme pour afficher une image, ni d'interface graphique, ni de système de mise à jour...
- Au fur et à mesure on a trouvé pratique de réunir tous ces outils de bases + le noyau dans un unique ensemble. On parle de **Distribution**.

Il existe de très nombreuses distributions pour un même noyau ! Et même des sous distributions etc. !



Une courte histoire d'Unix et GNU/Linux

- Au même moment que la publication du noyau de Linux, le projet GNU de richard Stallman **est presque complet**, il a tous les ingrédients pour former un système complet :
 - Le compilateur GCC (GNU compileur collection)
 - La GLIBC (GNU C Library, un ensemble de programme permettant de faire fonctionner un programme C)
 - GDB (GNU Debugger) : permet de debugger des programmes
 - L'interface de bureau GNOME (GNU Network Object Model Environment)
 - etc etc... (voir https://fr.wikipedia.org/wiki/Liste_des_paquets_GNU)

Une courte histoire d'Unix et GNU/Linux

- Au même moment que la publication du noyau de Linux, le projet GNU de richard Stallman **est presque complet**, il a tous les ingrédients pour former un système complet :
 - Le compilateur GCC (GNU compiler collection)
 - La GLIBC (GNU C Library, un ensemble de programme permettant de faire fonctionner un programme C)
 - GDB (GNU Debugger) : permet de debugger des programmes
 - L'interface de bureau GNOME (GNU Network Object Model Environment)
 - etc etc... (voir https://fr.wikipedia.org/wiki/Liste_des_paquets_GNU)
- Sauf ... le noyau. Il décide alors d'utiliser **le noyau de Linus Torvald** et d'y mettre tout le reste autour pour obtenir un ensemble complet et fonctionnel !

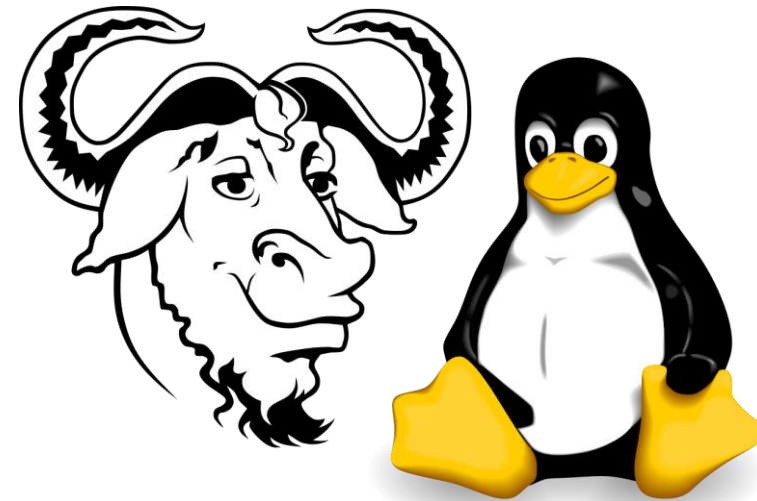
Une courte histoire d'Unix et GNU/Linux

- Au même moment que la publication du noyau de Linux, le projet GNU de richard Stallman **est presque complet**, il a tous les ingrédients pour former un système complet :
- Sauf ... le noyau. Il décide alors d'utiliser **le noyau de Linus Torvald** et d'y mettre tout le reste autour pour obtenir un ensemble complet et fonctionnel !

Une courte histoire d'Unix et GNU/Linux

- Au même moment que la publication du noyau de Linux, le projet GNU de richard Stallman **est presque complet**, il a tous les ingrédients pour former un système complet :
- Sauf ... le noyau. Il décide alors d'utiliser **le noyau Linux de Linus Torvald** et d'y mettre tout le reste autour pour obtenir un ensemble complet et fonctionnel !

C'est la naissance du système **GNU/LINUX**



Une courte histoire d'Unix et GNU/Linux

- Points importants :
 - GNU est philosophiquement et politiquement au cœur du logiciel libre. De fait, les outils GNU sont en général très protégés d'un point de vue libre.
 - À l'inverse Linus Torvald a une approche plus « pragmatique », il est ainsi possible d'utiliser le noyau Linux et de construire autour des systèmes plus commerciaux : **Android** par exemple est basé sur le noyau Linux, et même si il est Open Source il reste très lié à Google.



Une courte histoire d'Unix et GNU/Linux

- Points importants :
 - Pendant ce temps les systèmes basés sur l'Unix d'origine ont continué à vivre.
 - Il existe de nombreuses distributions commerciales basés sur ces systèmes.
 - L'exemple le plus connus est **macOS**, le système d'exploitation des Mac.



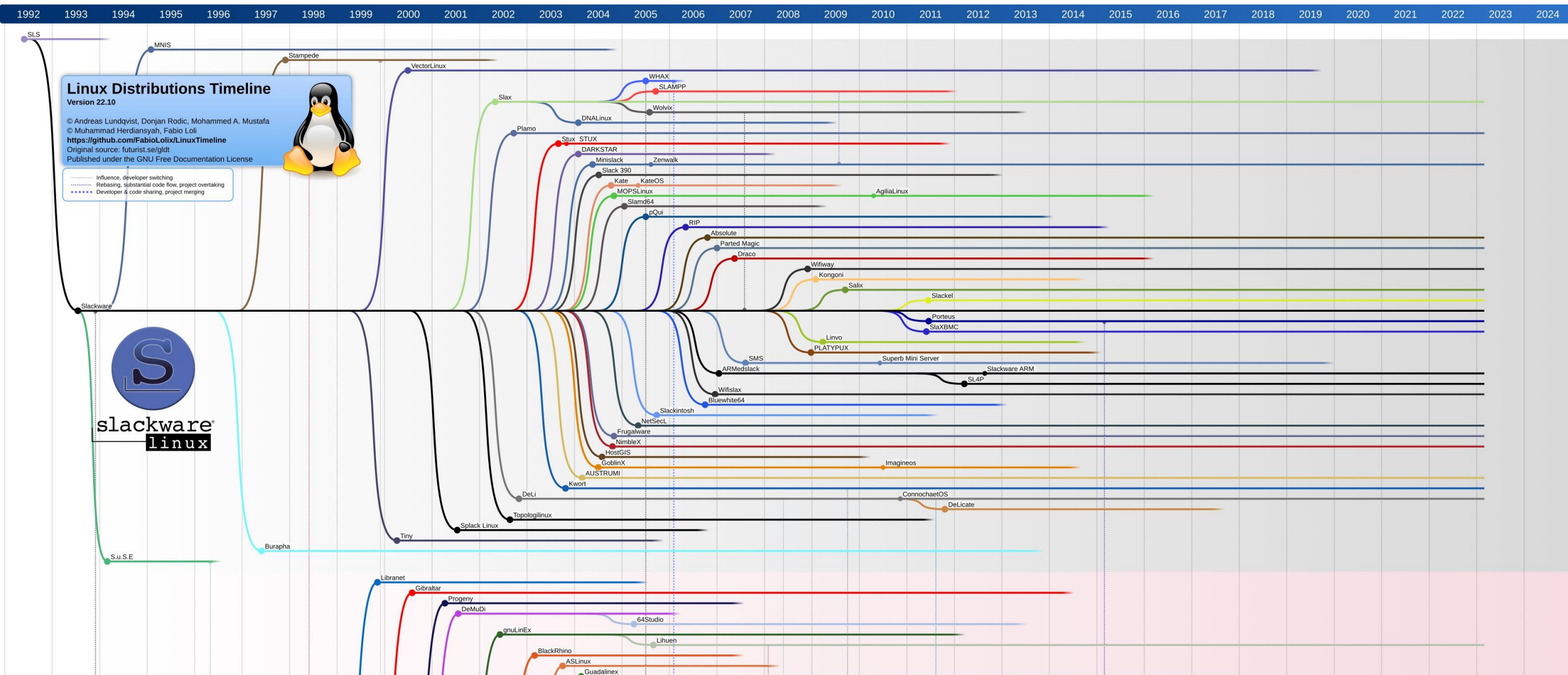
Mac OS X

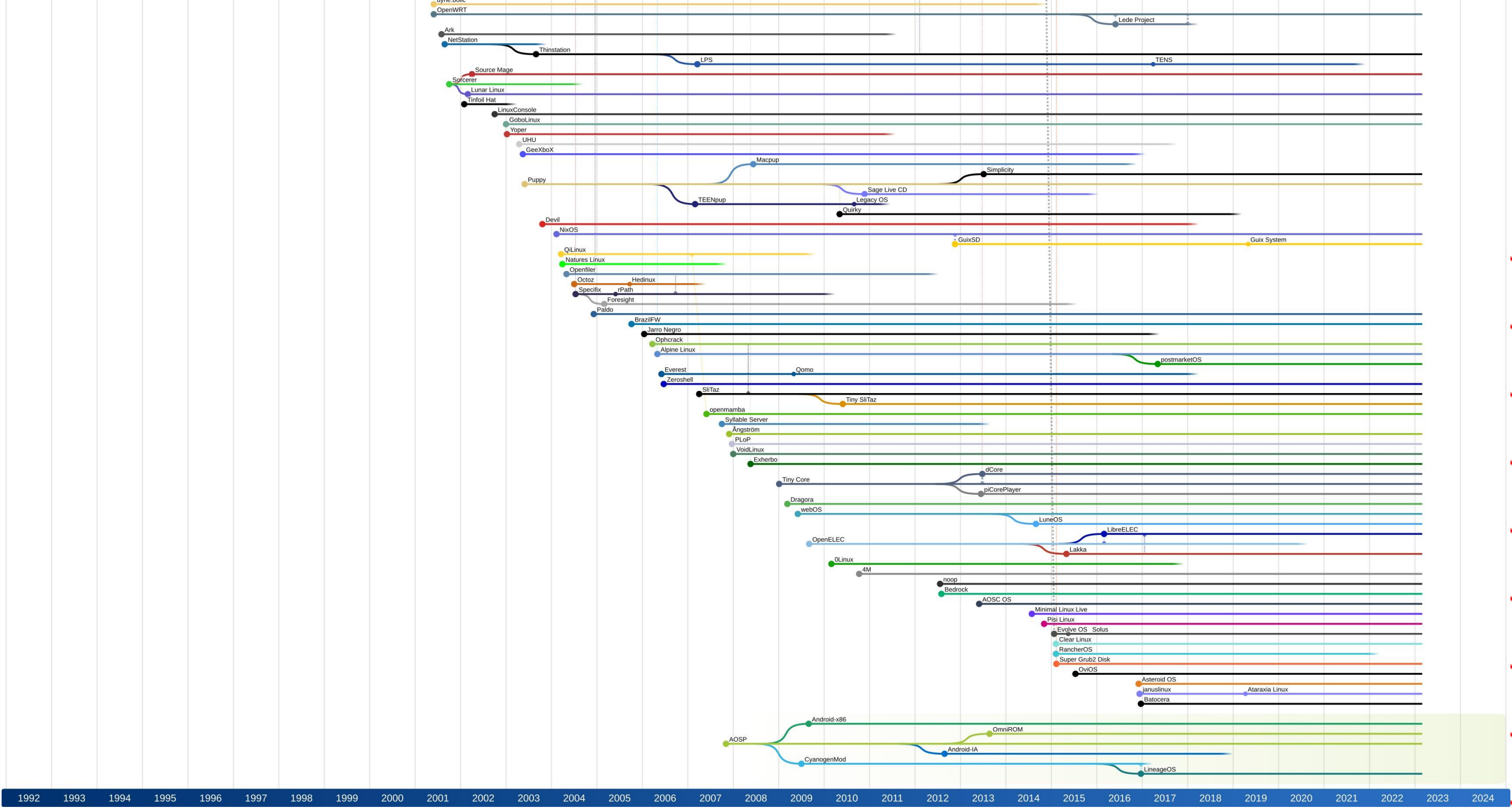


Tant de systèmes et d'histoire indépendante, comment les maîtriser ?

- Par chance, tous ces systèmes, même ceux développés indépendamment comme le noyau Linux ou les outils GNU, sont très souvent « compatibles UNIX ».
- En clair : ils ne se différencient pas énormément dans la manière de parler au système et dans la plupart des opérations.
- Par exemple : que ce soit sur Mac ou sur Smartphone, les commandes de bases de déplacement etc. sont strictement les mêmes.
- **La très large majorité de ce que vous verrez dans ce cours est valable pour l'ensemble de ces systèmes.**

Quelle distribution GNU/LINUX Choisir ?





Quelle distribution GNU/LINUX Choisir ?

- De par le fait qu'elle est la base de nombreuses distributions (le plus gros arbre dans ce que vous avez vu précédemment) et qu'elle est simple, nous utiliserons la distribution Debian;
- C'est celle que vous avez pu installer dans le TP d'installation WSL.



Les bases d'Unix – GNU/Linux

Aperçu général

- Qu'est-ce qu'Unix
- Architecture du système Unix
- Le système de fichier Unix
- L'environnement Shell

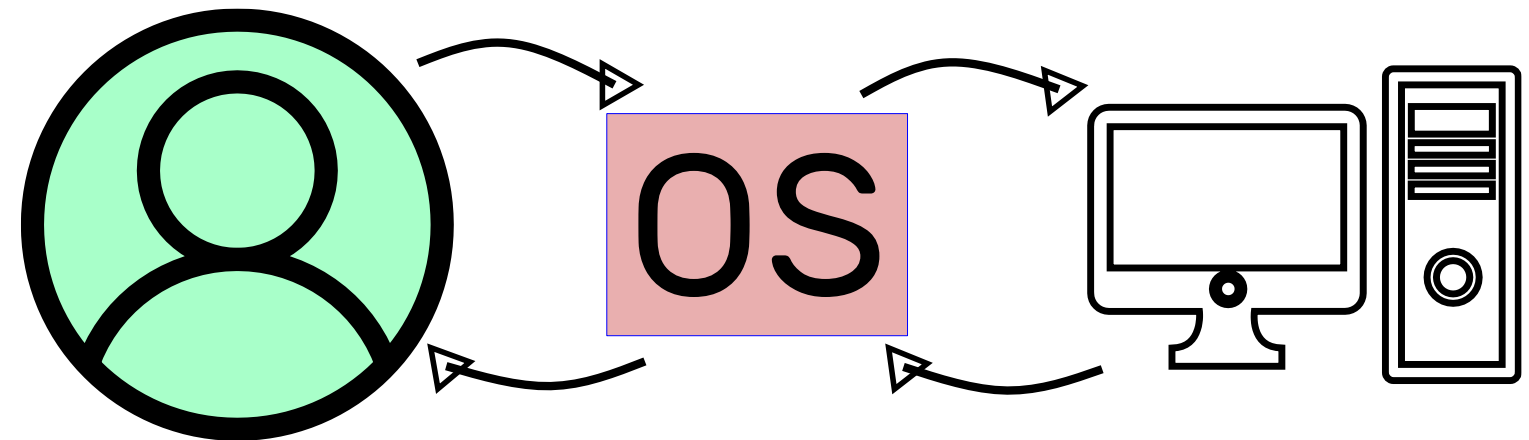
Qu'est-ce qu'Unix ?

- UNIX est un système d'exploitation (operating system, OS)



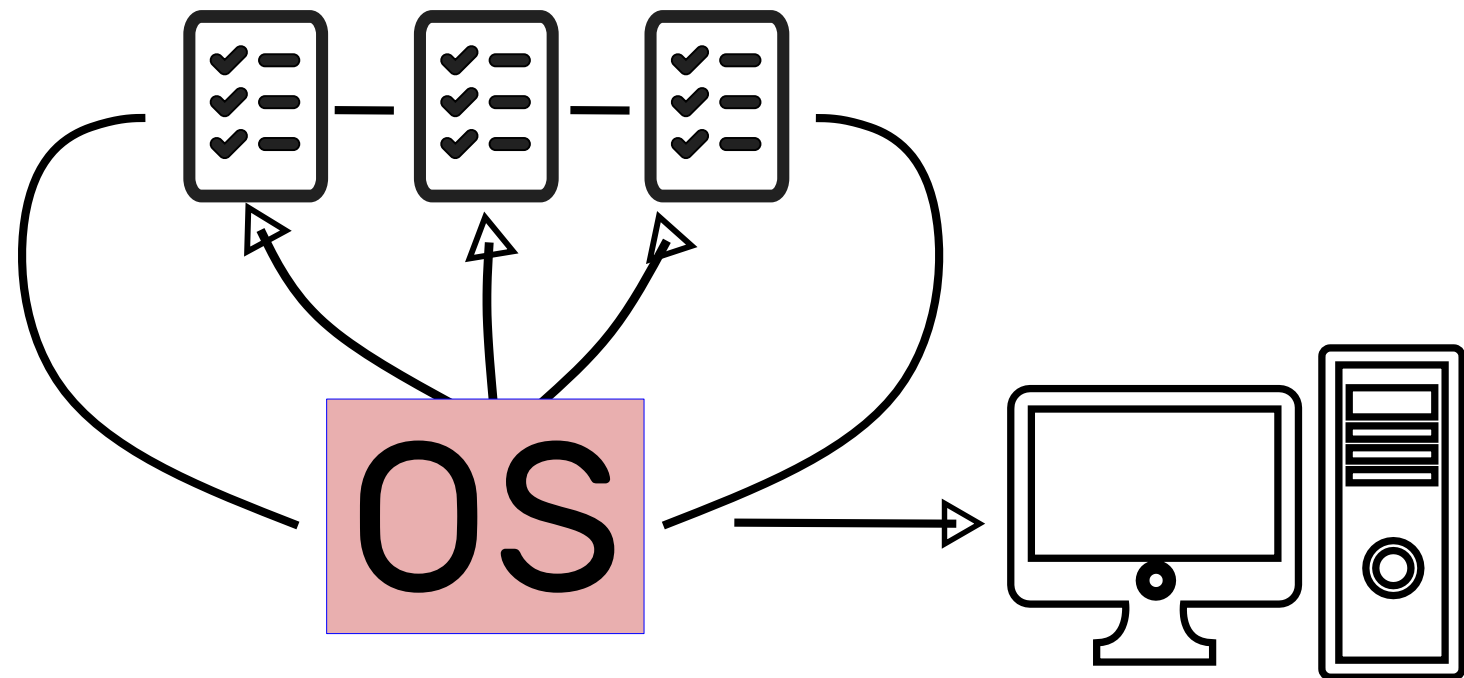
Qu'est-ce qu'Unix ?

- UNIX est un système d'exploitation (operating system, OS)
- UNIX est **interactif**



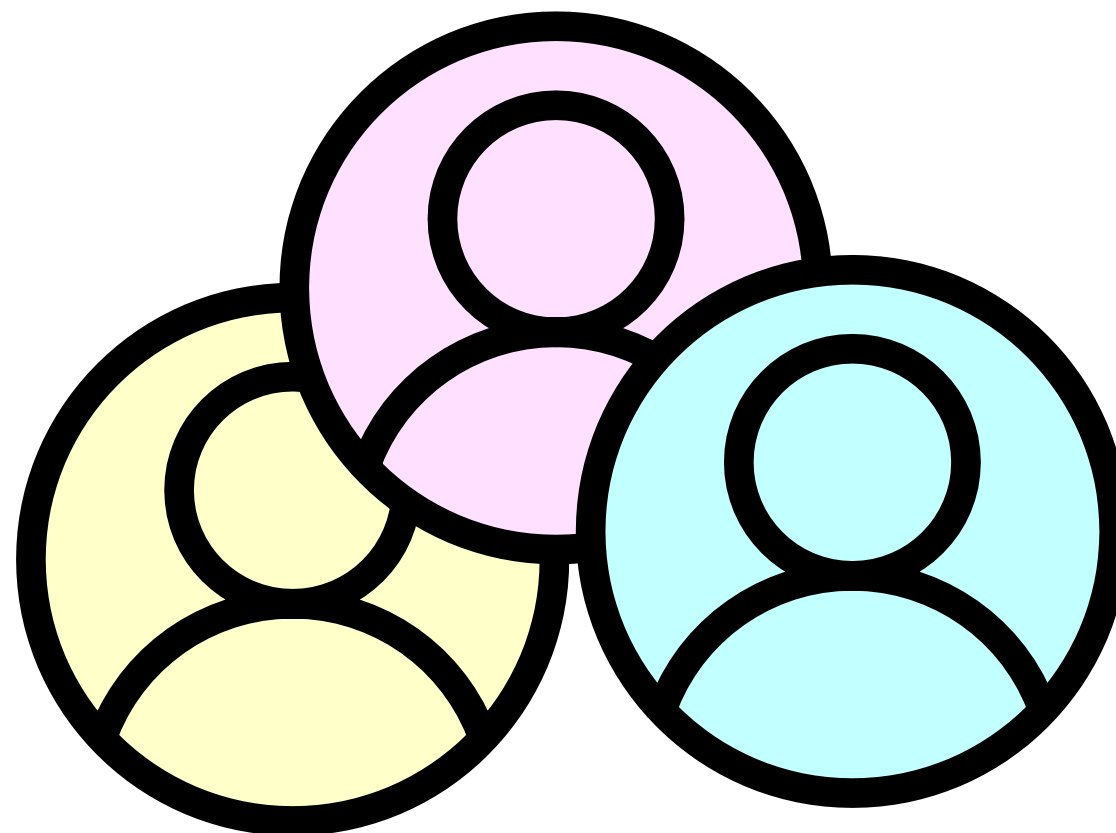
Qu'est-ce qu'Unix ?

- UNIX est un système d'exploitation (operating system, OS)
- UNIX est **interactif**
- UNIX est **multi-tache**



Qu'est-ce qu'Unix ?

- UNIX est un système d'exploitation (operating system, OS)
- UNIX est **interactif**
- UNIX est **multi-tache**
- UNIX est **multi-utilisateur**

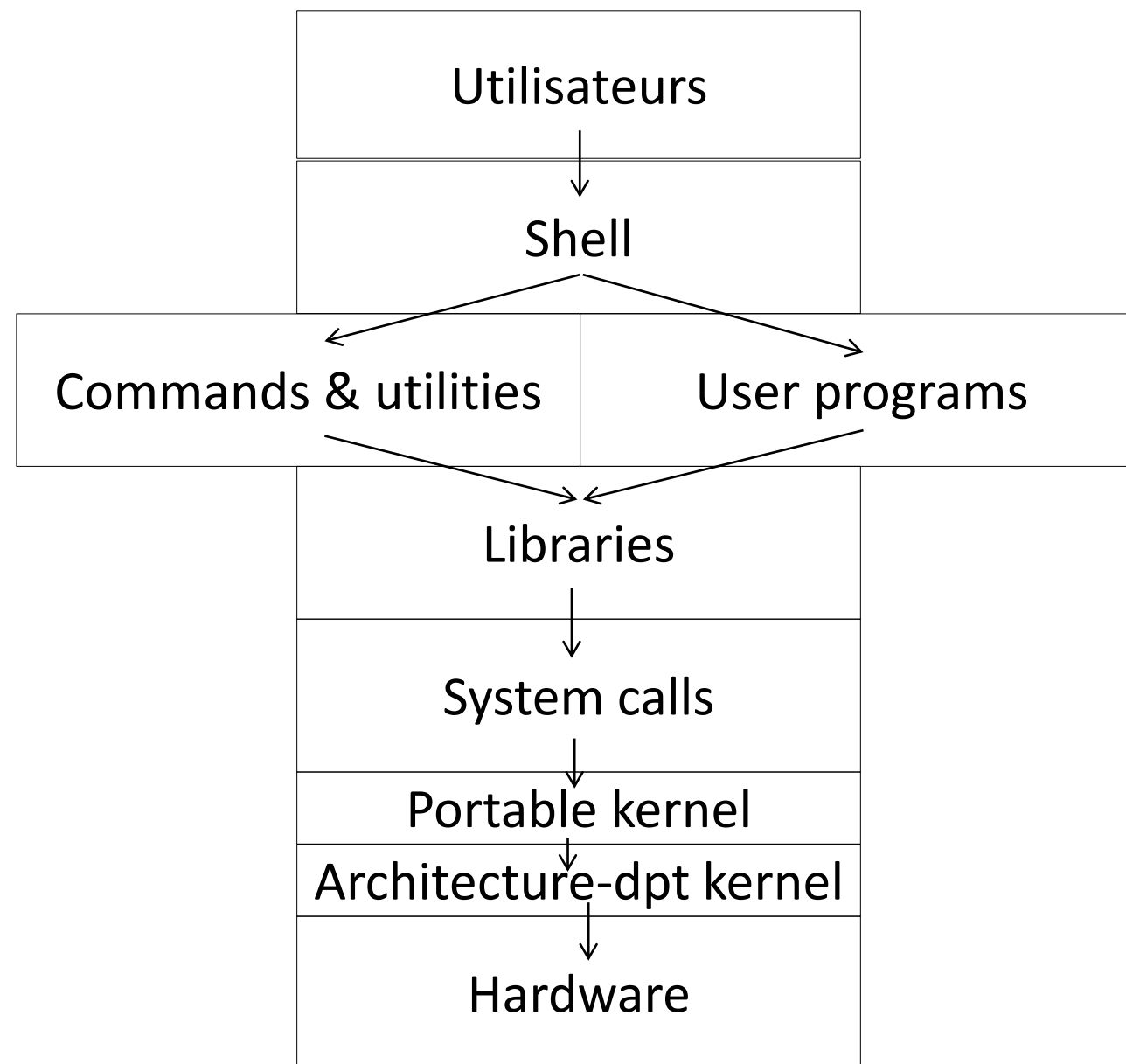


Qu'est-ce qu'Unix? (2)

- UNIX peut être :
 - Limité au noyau (Kernel) du système d'exploitation.
 - Noyau + programme de bases (compilers, editors, shells, etc.)
 - Noyau + programme de bases + divers programmes supplémentaires.

Architecture des systèmes Unix

- Architecture en couches
- Portabilité :
 - Du noyau
 - Des programmes



Architecture des systèmes Unix

- Le Kernel Linux :
 - Contrôle et gère les accès au matériel
 - Implémente les fondamentaux des systèmes d'exploitation
 - Ordonnancement et allocation des ressources :
 - Mémoire, CPU, Bande passante, etc.
 - Gère des **sécurités de bases** (espaces mémoire pour les programmes, différenciation des utilisateurs et des droits etc.)
 - Réponds aux appels systèmes des programmes

Architecture des systèmes Unix

- Le noyau gère :
 - Les Entrées Sorties (I/O)
 - Les interruptions
 - Les processus
 - La Mémoire
 - Le système de fichier
 - Les appels systèmes
 - L'interface réseau

Utilisateurs

- Les Utilisateurs :
 - Sont identifiés par un nom et un ID unique (unique ID => **uid**)
 - Appartiennent à 1 ou plusieurs **groupes**
 - Ont un mot de passe par utilisateurs

Utilisateurs (2)

- Les groupes :
 - Sont identifiés par un nom et un identifiant unique (Groupe ID => **gid**)
 - Peuvent avoir un mot de passe, mais c'est optionnel.



Utilisateurs (3)

- La phase de login :
 - Les utilisateurs doivent s'identifier (login) avant d'utiliser le système
 - Cette phase d'identification lance en general divers scripts et démarre un shell interactif (bash)

Utilisateurs (4)

- Ce qu'un utilisateur peut faire est conditionné par ses **droits**.
- Des paramètres globaux peuvent être définis pour tous les utilisateurs
- On peut également en définir pour chaque utilisateurs séparément :
 - Type de shell, interface graphique par utilisateur, répertoire « home » etc.

Utilisateurs (5)

- Les paramètres peuvent être définis par :
 - L'administrateur du système (les utilisateurs normaux ne peuvent pas modifier ces valeurs)
 - L'utilisateur lui-même (personnalisation de l'expérience utilisateur)

Utilisateurs (6)

- Le compte super-utilisateur : **root**
 - Est le compte d'administration du système
 - Possède tous les droits sur le système
 - Peut changer l'identifiant (ID) : peut devenir n'importe quel utilisateur défini

Utilisateurs (7)

- Le compte super-utilisateur : **root**
 - Peut suspendre/arrêter n'importe quel processus
 - Peut modifier les droits sur les fichiers
 - Peut changer/définir des mots de passe

Utilisateurs(6)

- Mots de passe :
 - Renforcent la sécurité de base de l'utilisateur du système
 - Sont stockés de manière chiffrée dans un fichier en lecture seule sur le système :
 - /etc/passwd dans les anciennes versions
 - /etc/shadow dans les systèmes récents

Utilisateurs (7)

- Les mots de passe :
 - Ne peuvent pas être déchiffrés
 - Chaque utilisateur peut (normalement) changer son propre mot de passe
 - "root" peut définir et modifier n'importe quel mot de passe

Utilisateurs (8)

- Conseils de sécurité :
- Comment choisir un bon mot de passe :
 - Ne jamais choisir un mot basé sur un dictionnaire
 - Utiliser la sensibilité à la casse (majuscules et minuscules) dans le mot de passe
 - Inclure des chiffres et des caractères spéciaux
- Si la sécurité est importante :
 - Utiliser des systèmes de mots de passe étendus : mots de passe en mode "shadow", chiffrement plus robuste
 - Appliquer des politiques de changement de mot de passe
 - L'utilisation d'un gestionnaire de mots de passe ("password safe") aide à conserver des mots de passe complexes pour chaque compte.

Le système de fichier d'Unix et Linux

*« On a UNIX system, everything is a file;
if something is not a file, it is a process. »*

- Les fichiers sont stockés **dans une structure en forme d'arbre.**
- Le point d'entrée de la structure s'appelle la racine (root) et est représenté par le symbole /. Un fichier faisant parti d'un répertoire nommé « poneys » présent à la racine sera ainsi à l'emplacement « /poneys/monfichier »
- Le système de fichier est divisé en répertoire, qui ne sont que des fichiers spéciaux (voir la maxime ci-dessus).
- L'accès aux fichiers et répertoires est faites selon **les droits** qu'ont les utilisateurs.

Le système de fichier

- Divers types de fichiers :
 - Fichiers "normaux"
 - Répertoires
 - Liens (liens physiques ou symboliques)
 - Fichiers spéciaux (périphériques)
 - Pipes

Organisation des dossiers dans les OS

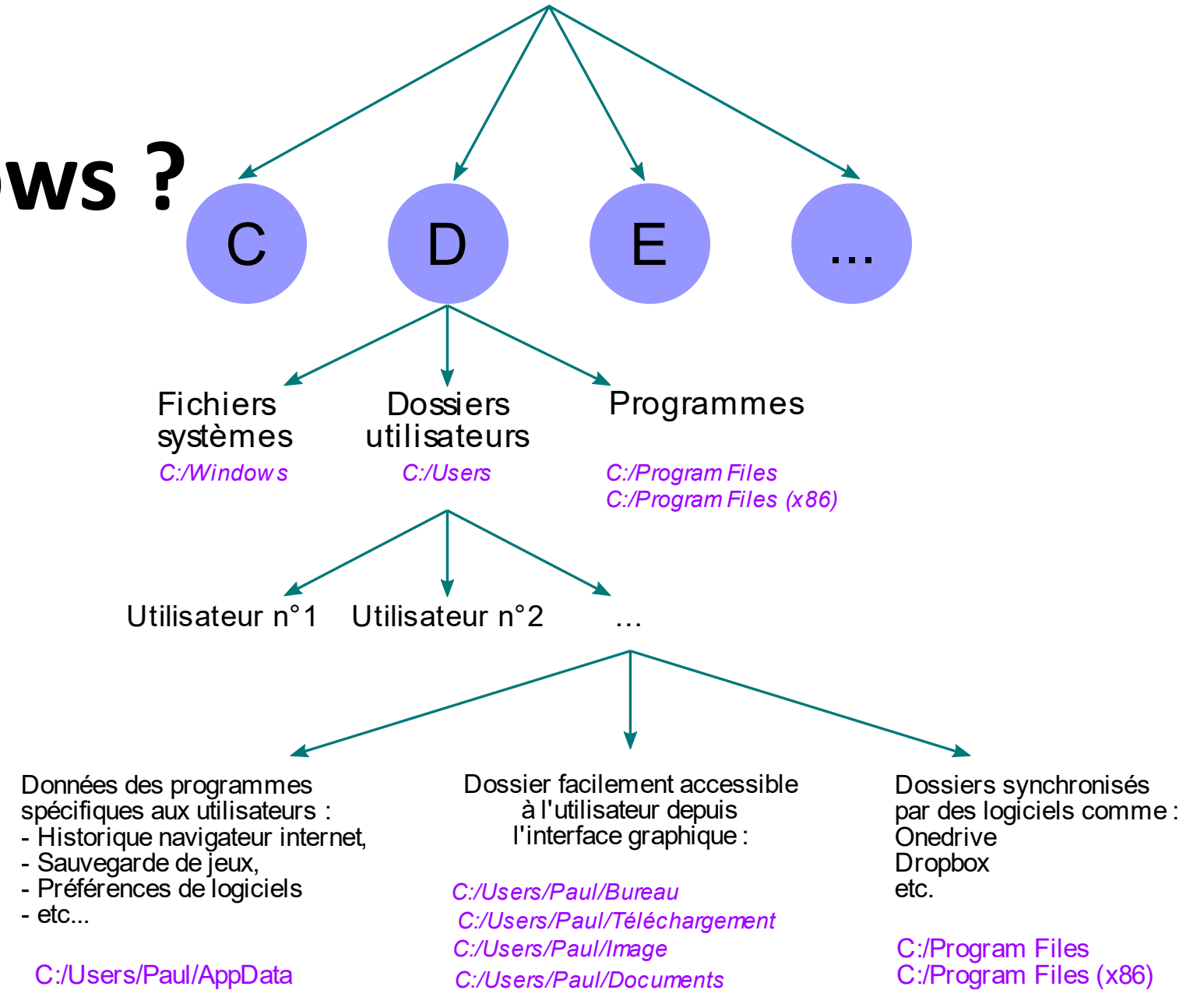
- Le système d'exploitation (OS) gère différemment certains répertoires :
 - Systèmes de fichiers :
 - **Sous Windows** : Une lettre par système de fichiers présent
 - C: En général utilisé comme système de fichier principal par Windows
 - D: , E: , ... en fonction des périphériques branchés

Contenu de la partition C sous Windows ?

- **Utilisateurs** > login > Répertoires de l'utilisateur.
 - dont **Mes documents**, **Mes images**, **Téléchargements**, **Bureau**, ...
- **Program files** et **Program Files (x86)** pour les installations d'applications.
- **Windows** pour les fichiers nécessaires au système d'exploitation.

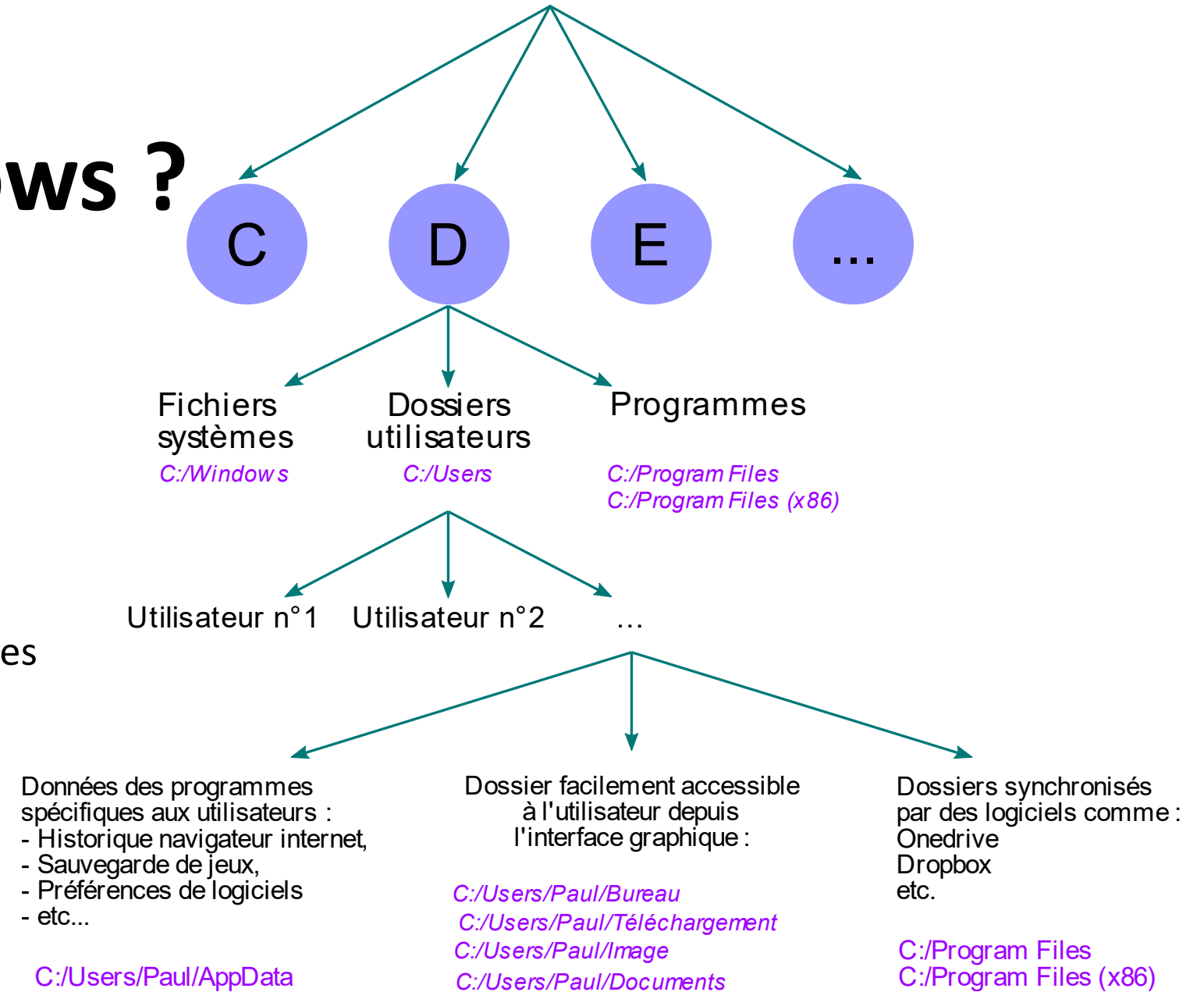
Organisation des dossiers sous Windows ?

- Schéma →



Organisation des dossiers sous Windows ?

- Schéma →
- Notes :
 - On peut changer l'emplacement de beaucoup de ces dossiers
 - Certaines lettres de lecteurs sont **réservées** pour des raisons historiques : A et B étaient les **lecteurs de disquette** à l'époque !
 - Pour des raisons de retro-compatibilités, ils sont conservés.

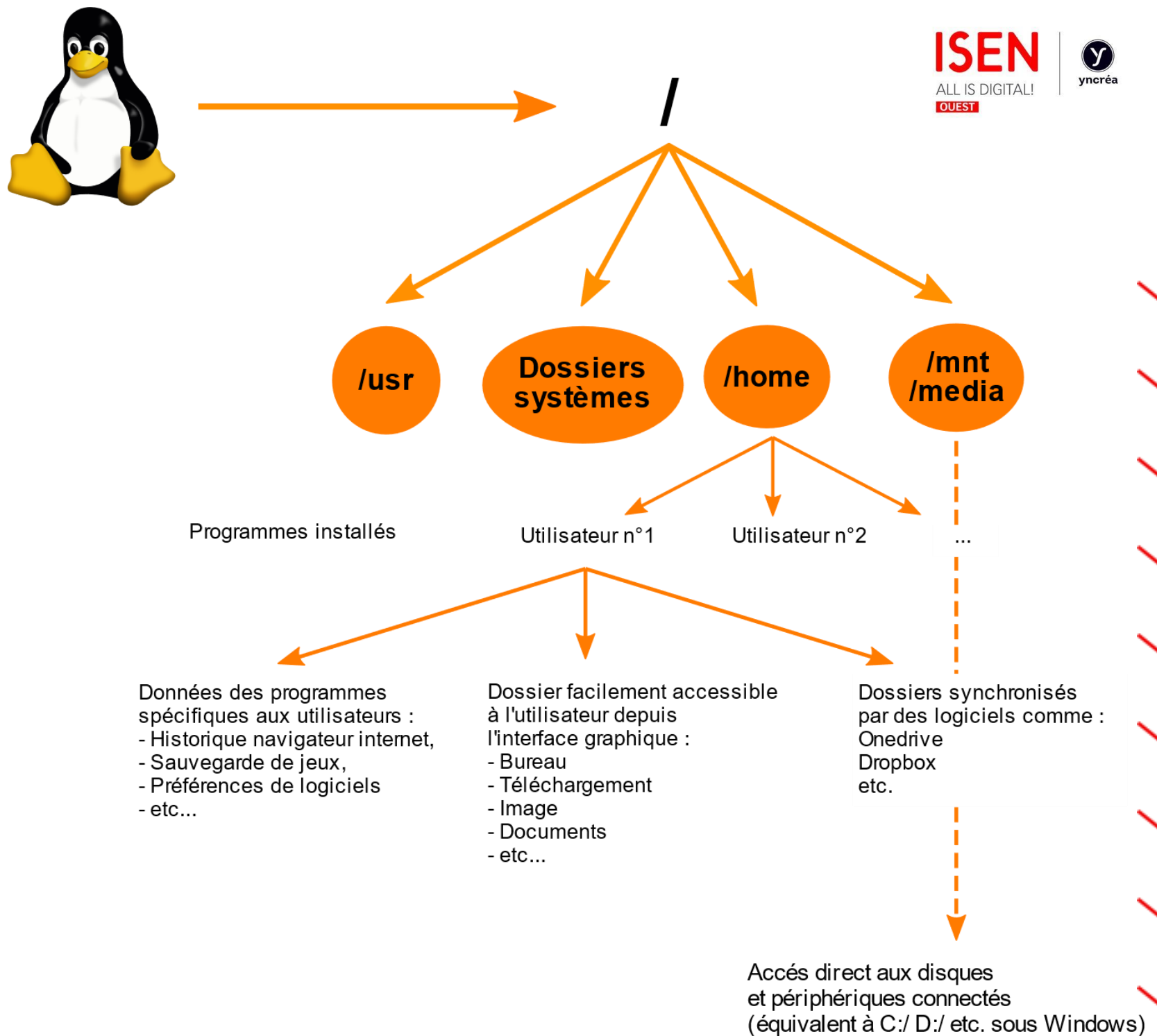


Organisation des dossiers dans les OS

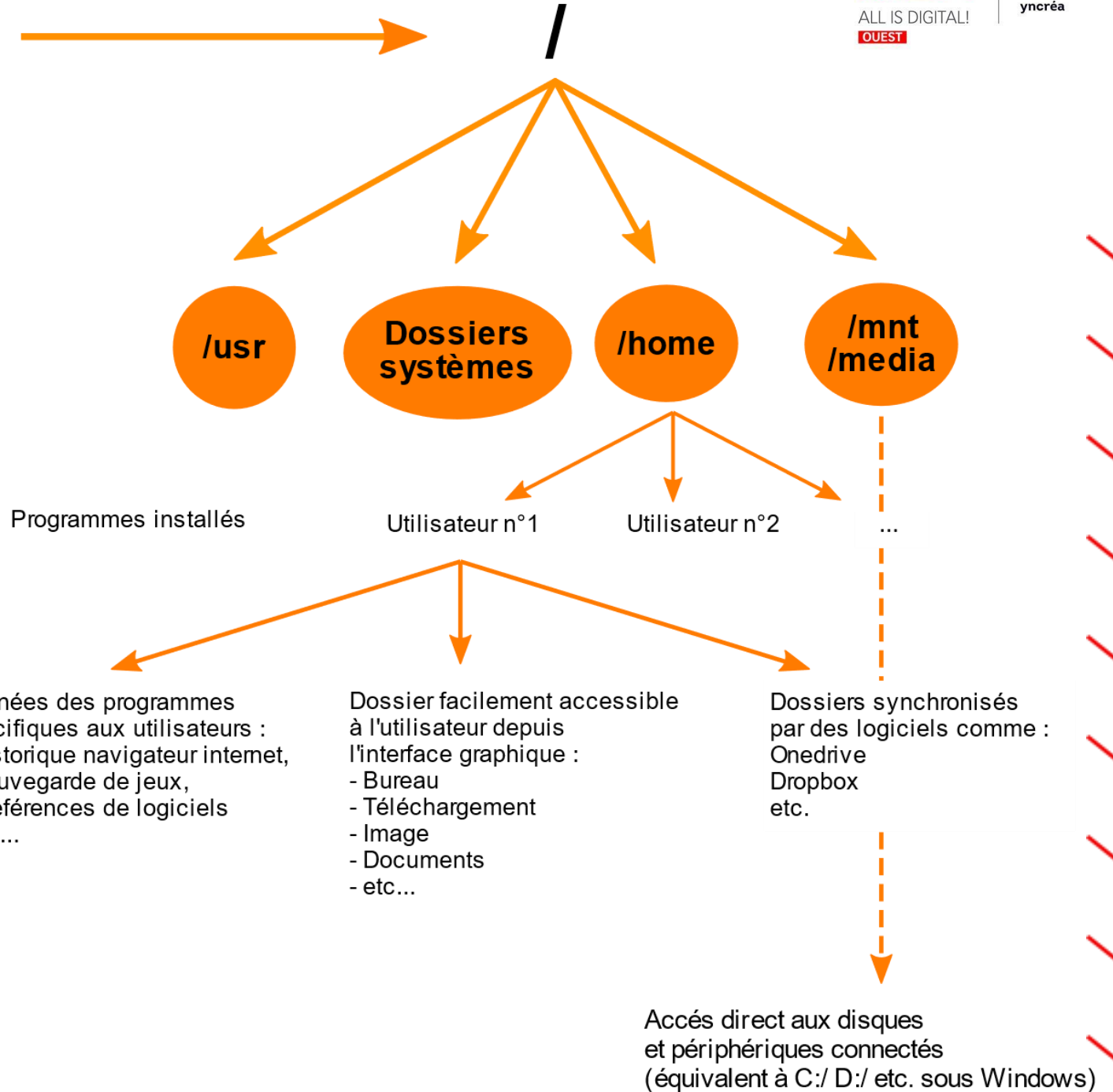
- **Sous Linux** : C'est plus abstrait.
 - « / » est la base de l'arborescence
 - Les dossiers systèmes permettant à Linux de fonctionner sont ensuite réparties dans /
 - Par exemple :
 - /bin** : dossiers contenant les programmes de base
 - /etc** : dossier contenant les fichiers de config
 - /usr** : dossier contenant les programmes installées (firefox etc.) :
 - à peu près équivalent à program files sous windows
 - /home** → contient toutes les données spécifiques à un utilisateur (de manière très rigoureuse : si on supprime tout sauf home et qu'on réinstalle, aucune donnée personnelle n'est perdue) :
 - à peu près équivalent à C:/Users sous Windows.

Organisation des dossiers sous Linux ?

- Schéma →



Organisation des dossiers sous Linux ?



- Schéma →
- Notes :
 - Chaque dossier peut être sur un disque physique parfaitement différent !
 - Cela permet notamment de sauvegarder le répertoire utilisateur à part.

Organisation des dossiers dans les OS

- **Sous MAC** : Idem que sous Linux car construit sous la même base.
- Sauf que :
 - **/Applications** contient les applications installées
 - **/Library** ou **/Bibliothèque** remplace **/lib** sous linux.
 - **/Users** remplace **/home**
 - **/System** contient les fichiers systèmes
- En somme : c'est un mélange des logiques de Linux et Windows.

Le système de fichier (2)

Propriétés standard d'un fichier :

- Taille
- UID du propriétaire
- GID du propriétaire
- Nombre de liens physiques
- Permissions
- Dates

C'est parti !

