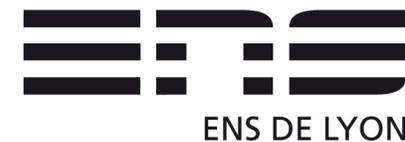


# Introduction aux architectures et systèmes d'exploitation

Francesco Bronzino

ArchiSys L3



# Sujets

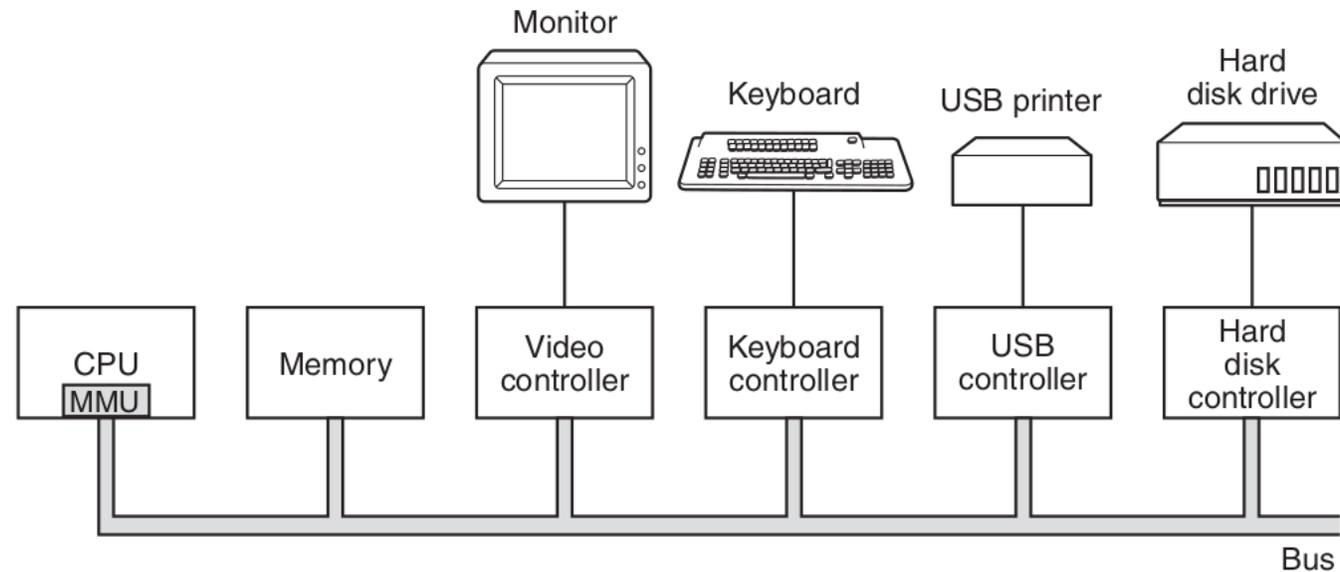
1. Les composants d'un système de traitement
2. Architecture d'un système de traitement
3. Définition d'un système d'exploitation
4. Les composants d'un système d'exploitation
5. Définitions relatives aux systèmes d'exploitation
6. Types de systèmes d'exploitation

# Architecture d'un système de traitement

- Un système de traitement peut être divisé en quatre composantes :
  - **Hardware**
    - Fournit les ressources physiques : CPU, mémoire, disque
  - **Système d'exploitation**
    - Gérer l'accès au matériel par les programmes
  - **Programmes / applications**
    - Effectuer les tâches souhaitées par les utilisateurs ou le système
  - **Utilisateurs**
    - Ils lancent et utilisent des programmes

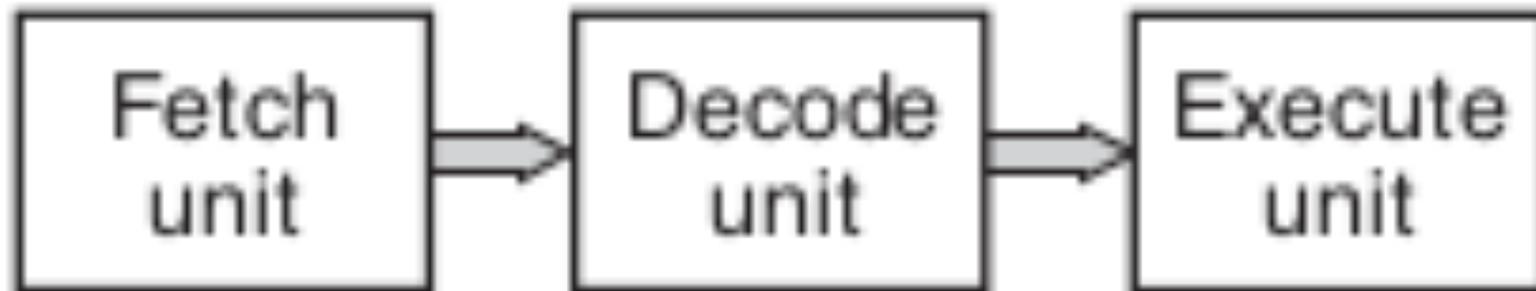
# Architecture d'un système de traitement

- Constitué de plusieurs unités
- Le **bus** relie les composants du système.



# Architecture d'un système de traitement

- Le **CPU** exécute les instructions qu'il prend en mémoire.
- Modules fondamentaux :
  - Unité arithmétique et logique
  - Unité de contrôle
  - Registres
- Trois étapes pour exécuter chaque instruction

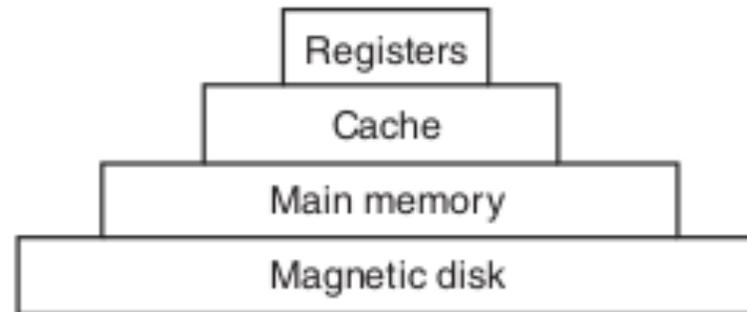


# Architecture d'un système de traitement

- La **Mémoire** offre un moyen de sauvegarder des données
- Un système possède plusieurs types de mémoire, avec des caractéristiques différentes
  - Organisation dans une hiérarchie de mémoire

Typical access time

1 nsec  
2 nsec  
10 nsec  
10 msec

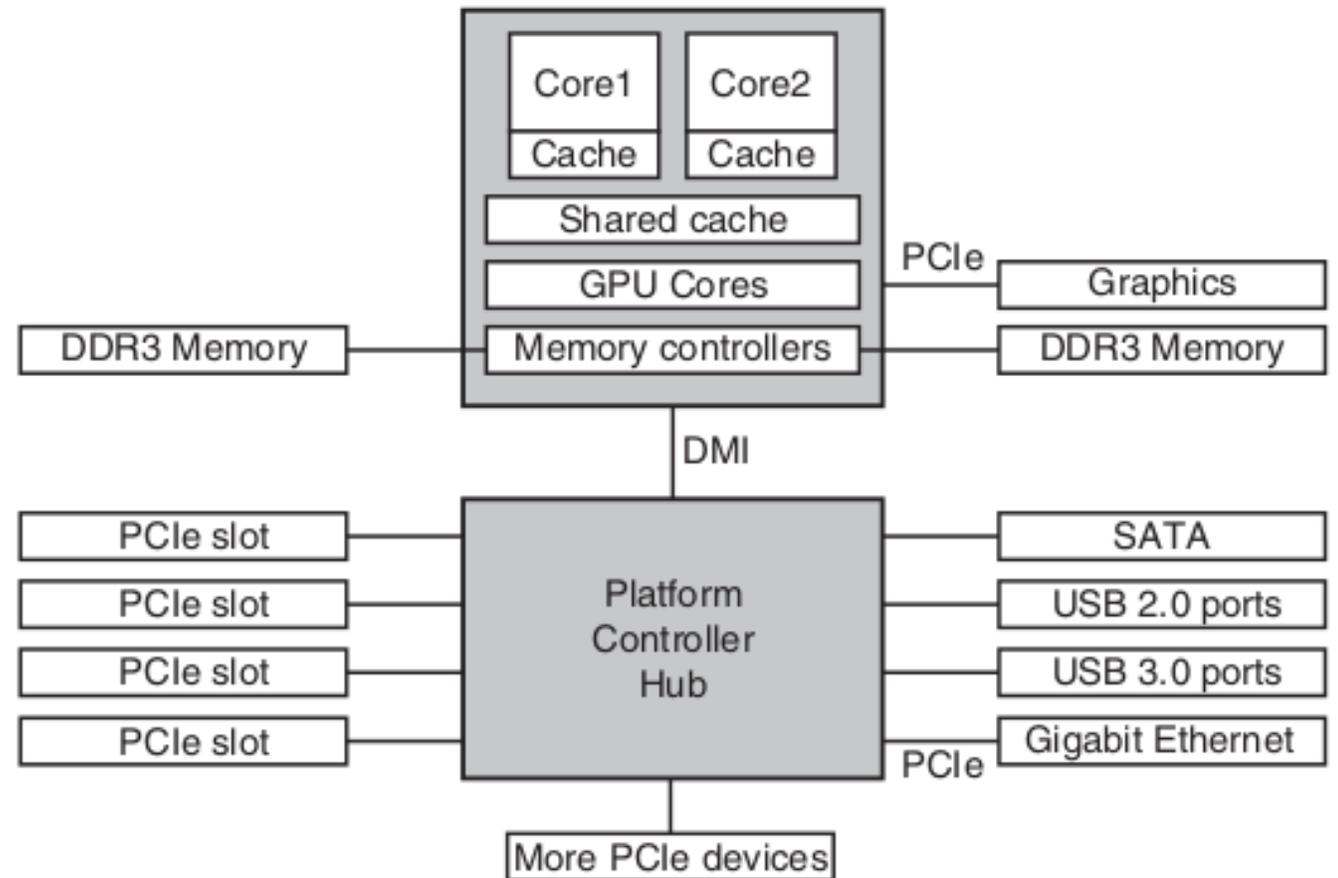


Typical capacity

<1 KB  
4 MB  
1-8 GB  
1-4 TB

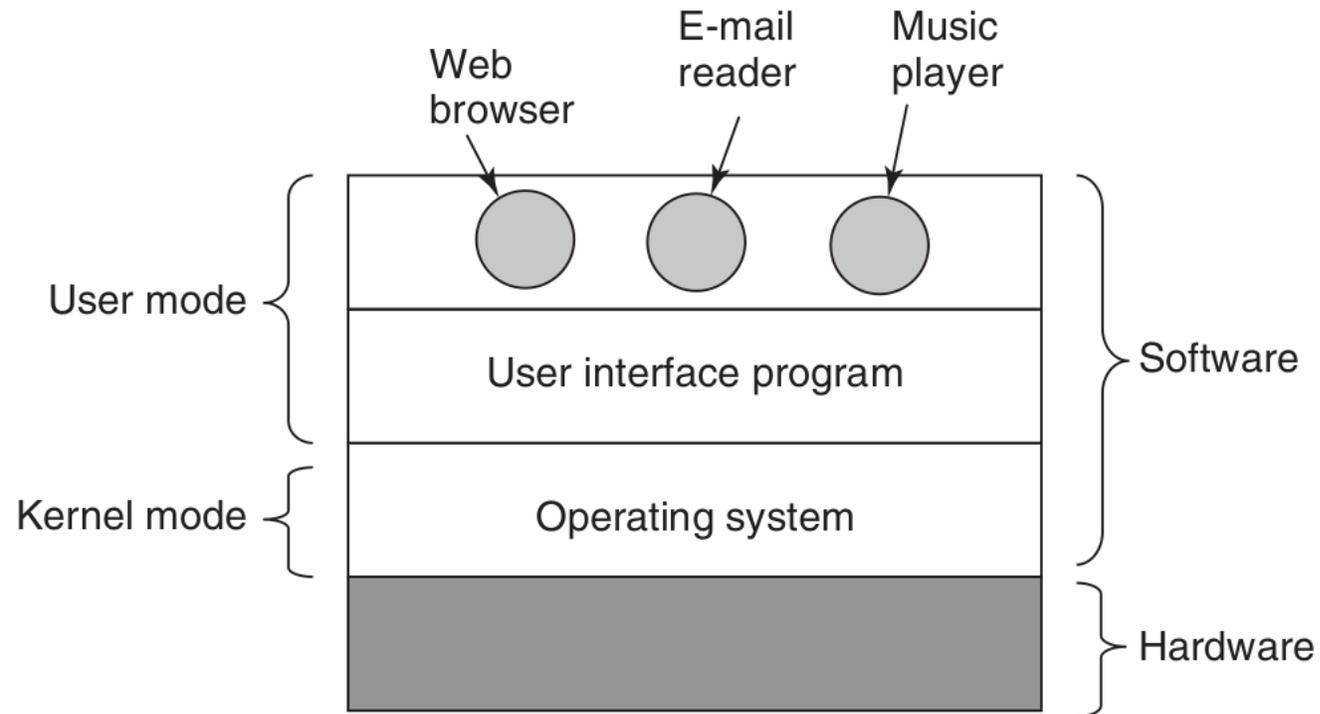
# Architecture d'un système de traitement

- Un système de traitement interagit avec le monde extérieur par l'intermédiaire de **dispositifs d'entrée/sortie**
  - Écran
  - Clavier
  - Souris
  - Réseau
  - Capteurs
- Communiquer avec l'unité centrale via le bus



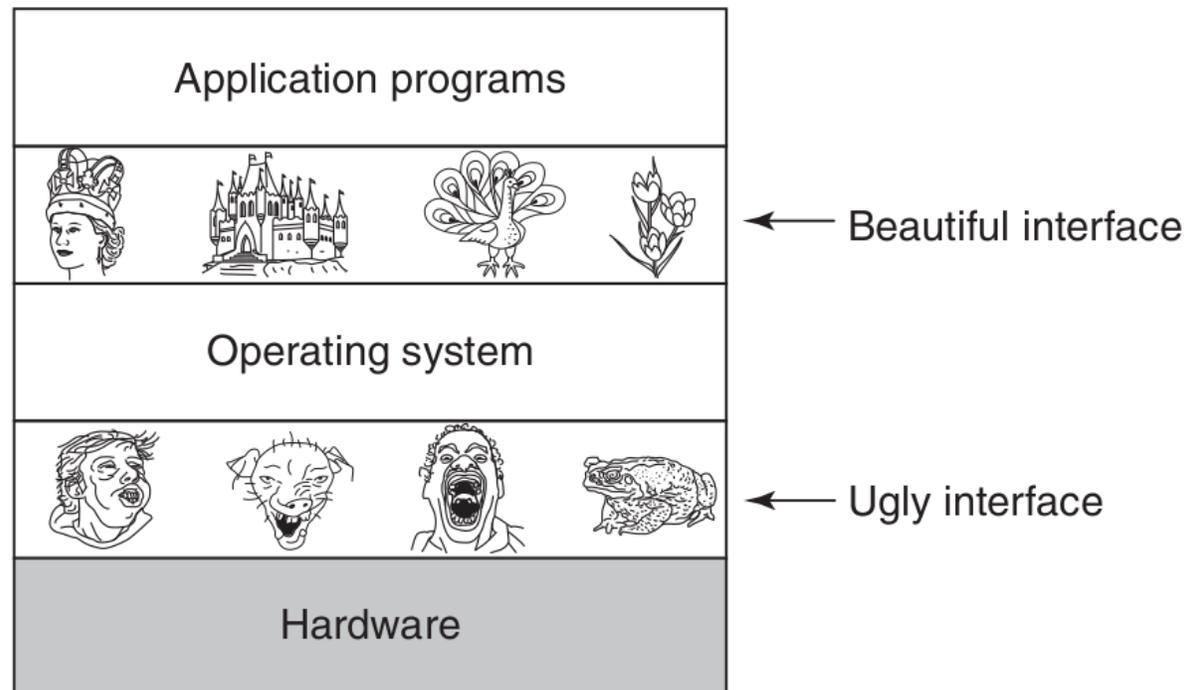
# Définition du système d'exploitation

- La tâche des systèmes d'exploitation est :
  - Interagir avec l'**hardware**
  - Fournir à l'utilisateur un **modèle informatique plus simple**



# Définition du système d'exploitation

- D'une certaine manière, les systèmes d'exploitation rendent agréable ce qui a une interface désagréable...

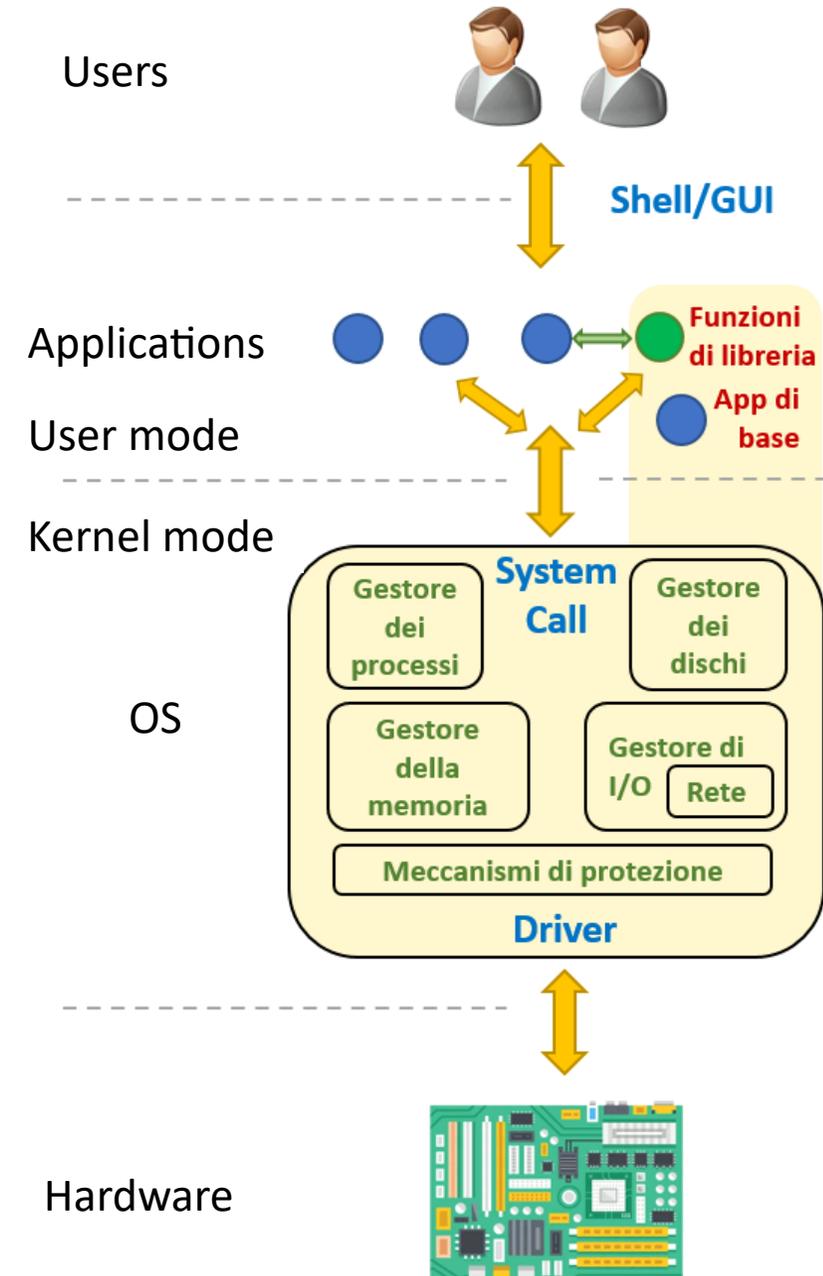


# Définition du système d'exploitation

- Le système d'exploitation s'interface avec les dispositifs matériels
  - Via des modules logiciels appelés **Drivers**
- Il offre des services aux applications
  - Via des API appelées **Appels système**
- Organisé en modules

# Les composants d'un système d'exploitation

- Un système de traitement se compose de plusieurs modules
  - Qui offrent des services aux utilisateurs
  - Mais ils interagissent également les uns avec les autres



# Les composants d'un système d'exploitation

- **Process Manager:**

- Créer et gérer des processus
  - Un processus est un programme en cours d'exécution.
- Il trouve les ressources CPU et mémoire nécessaires à l'exécution.

- **Memory Manager:**

- La mémoire d'un ordinateur est un vecteur unique
- Le système d'exploitation gère :
  - L'allocation de la mémoire aux processus et le partage de la mémoire entre les programmes.

- **Disk Manager:**

- Le stockage sur disque est persistant
- Le système d'exploitation organise le stockage sur disque dans une structure arborescente de répertoires et de fichiers.

# Les composants d'un système d'exploitation

- **Gestionnaire de périphériques d'I/O:**

- Les périphériques d'E/S ne sont pas gérés directement par les applications
  - Ce serait compliqué et créerait des problèmes
- Le système d'exploitation utilise des pilotes pour piloter les périphériques
- La connexion réseau représente un cas particulier d'entrée/sortie
- Il bénéficie d'un traitement spécial dans les OS modernes

- **Gestion des mécanismes de protection:**

- Dans les systèmes d'exploitation, il y a de nombreux utilisateurs avec des autorisations différentes pour accéder aux ressources.
  - Pour accéder aux fichiers, aux dispositifs, à la configuration, etc...
- Le système d'exploitation met en œuvre des politiques d'accès

# Les composants d'un système d'exploitation

- Autres composants:
  - **Synchronisation.** Permet aux processus
    - De communiquer les uns avec les autres
    - Pour se synchroniser entre eux
  - **Virtualisation.** Pour prendre en charge la création de machines virtuelles ou similaires (par exemple, des conteneurs)
    - Si le système d'exploitation offre des capacités de virtualisation, les machines virtuelles seront beaucoup plus rapides car elles peuvent avoir un accès direct à (certaines) ressources

# Définitions relatives aux systèmes d'exploitation

- **Kernel (Noyau) :**
  - Le cœur de l'OS
  - Comprend tous les modules vus ci-dessus
    - Gère les opérations fondamentales et de niveau inférieur
  - Module logiciel fonctionnant en permanence
    - Avec des privilèges spéciaux
    - Toutes les autres applications (utilisateur ou système) ont des privilèges mineurs.
      - Ils s'appuient sur le noyau
  - Différents types de noyaux
    - Monolithique : le noyau est un programme unique qui exécute tout le code nécessaire.
      - Les plus courants
    - Micro-noyaux : ils essaient de déléguer le plus de fonctionnalités possible aux applications.
    - A couche : le noyau (et les programmes) sont organisés en une hiérarchie de processus aux privilèges croissants.

# Définitions relatives aux systèmes d'exploitation

- **Processus :**

- Programme de course
  - Avec certains privilèges et ressources
- Accède à l'unité centrale à tour de rôle avec d'autres programmes
  - Le système d'exploitation doit permettre de préserver son état lorsqu'il est en pause.
    - Nécessite la sauvegarde des registres du CPU
  - Chaque processus est identifié par un numéro appelé **PID**
- Un processus peut créer d'autres processus appelés enfants
  - On dit que les processus sont organisés dans un **Arbre des processus**

- **Thread :**

- Un processus regroupe les ressources d'un programme en cours
- Un thread individue un flux d'exécution
- Un processus peut avoir *un ou plusieurs* flux d'exécution en son sein.
- Identifié par un identifiant appelé **TID**

# Définitions relatives aux systèmes d'exploitation

- **Appel système :**

- Il s'agit de fonctions mises à la disposition des applications par le système d'exploitation
- Ils offrent des services d'OS pour créer des processus, accéder aux disques, etc...
  - Le noyau "sert" les demandes d'appel système des programmes
- A ne pas confondre avec les **Fonctions de la bibliothèque**
  - Qui sont des modules logiciels qui exécutent des tâches communes pour plusieurs applications
  - Il s'agit de logiciels communs avec les mêmes privilèges d'application
  - Peut (mais pas toujours) appeler les appels système

# Définitions relatives aux systèmes d'exploitation

- Appels système :
  - Les appels système sont des fonctions C. Ils ont des aspects différents
  - *POSIX* (Linux)

```
int read (int fd, void *buffer, size_t nbytes);
```

- *Win32/Win64 API*

```
BOOL ReadFile (  
    HANDLE fileHandle,  
    LPVOID dataBuffer,  
    DWORD numberOfBytesToRead,  
    LPDWORD numberOfBytesRead,  
    LPOVERLAPPED overlappedDataStructure  
);
```

# Définitions relatives aux systèmes d'exploitation

- **Appels système :**

- Exemples d'appels système courants :

- Gestion des processus : **fork**, **exec**, **wait**, **kill**
    - Gestion des fichiers : **open**, **close**, **read**, **write**

- **Note** : les systèmes d'exploitation proposent également de nombreuses fonctions de bibliothèque.

- Certaines fonctions de bibliothèque ne nécessitent pas d'appel système
    - Calculer la racine carrée : **double sqrt(double arg)** n'a pas besoin d'un appel système
    - Écriture à l'écran : **int printf(char \*format, arg list ...)**. Formate la chaîne de caractères à écrire, puis appelle l'appel système **write**

# Définitions relatives aux systèmes d'exploitation

- **Mode Kernel/Utilisateur :**

- **Mode noyau :** le noyau a un accès complet au matériel.

- Peut accéder à chaque emplacement de mémoire et registre de périphérique d'entrée/sortie.
    - Uniquement du code de haute qualité. Les erreurs sont potentiellement destructives

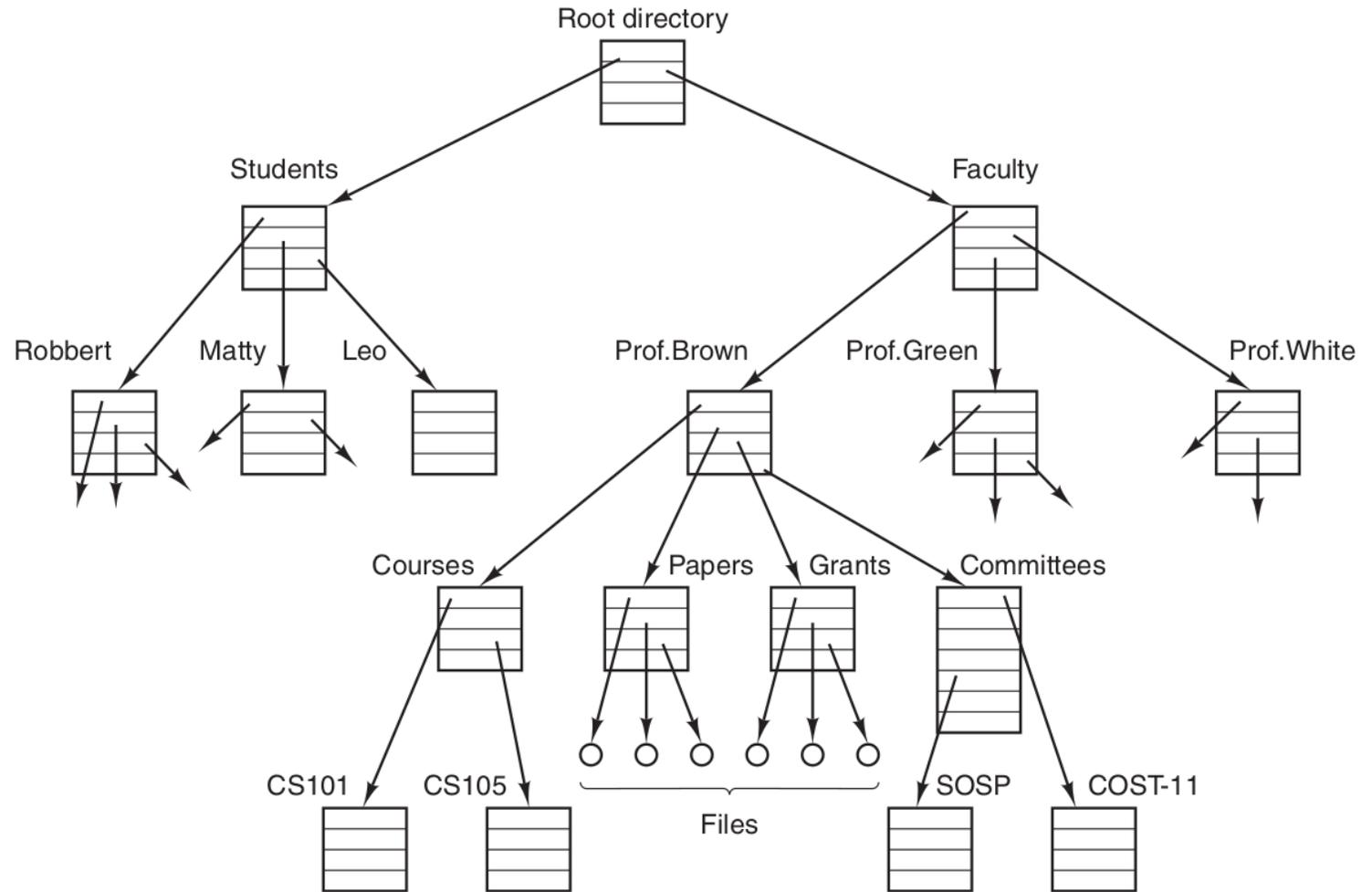
- **Mode utilisateur :** les applications ont des possibilités limitées.

- Ils accèdent à un espace d'adressage limité (mémoire virtuelle)
    - Impossible d'accéder directement aux périphériques d'I/O
    - Besoin d'une coopération avec le CPU pour mettre en œuvre une fonctionnalité spécifique
    - Les fonctions de la bibliothèque sont exécutées en mode utilisateur

# Définitions relatives aux systèmes d'exploitation

- **Systeme de fichiers :**

- Structure qui comprend un ensemble de fichiers et de dossiers.
- Organisé selon un arbre



# Définitions relatives aux systèmes d'exploitation

- **Organisation du parcours :**

- **Répertoire racine :** la racine de tous les répertoires. Elle est *identifiée* avec / dans Linux
- **Répertoire de travail:** répertoire où un processus est lancé
- **Chemin absolu :** *début*e par / et identifie un chemin à partir du *répertoire racine*
- **Chemin relatif :** *ne commence pas* par / mais par un nom. Identifie un chemin relatif au *répertoire de travail* du processus

# Définitions relatives aux systèmes d'exploitation

- **Bootloader :**

- Le code qui charge le noyau en mémoire lorsque le système est allumé
- Contenu de la ROM/EEPROM

- **Login :**

- Authentification d'un utilisateur dans le système, généralement par un nom d'utilisateur et un mot de passe

- **Shell :**

- Programme qui lit les commandes du clavier, les exécute et imprime la sortie
- Méthode traditionnelle de connexion
- Ne fait pas partie du noyau
  - Lorsqu'un système est démarré, le noyau lance toujours un shell ou l'interface graphique

# Histoire et types de systèmes d'exploitation

# Sujets

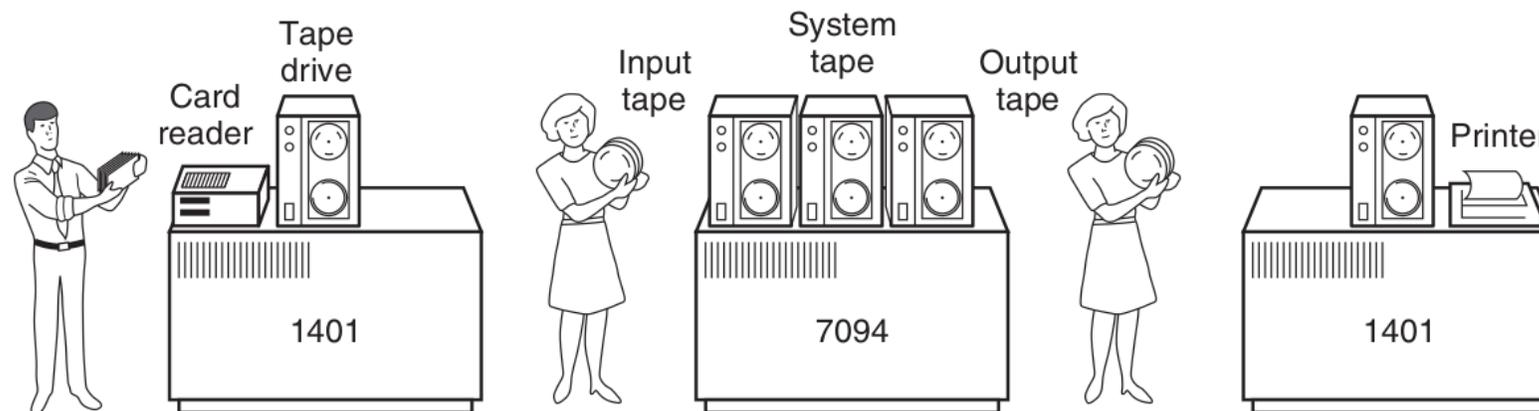
1. Histoire des systèmes d'exploitation
2. Types de systèmes d'exploitation
3. Linux

# Histoire des systèmes d'exploitation

- **Condition nécessaire** pour un système d'exploitation : avoir un système de traitement.
- Premier ordinateur conçu par Charles Babbage dans la première moitié du 19e siècle.
  - Purement mécanique.
  - Il n'a jamais été mis en œuvre
- Les premiers ordinateurs ont été construits dans les années 1940.
  - Programmé directement en langage machine
  - Aucun système d'exploitation. L'ordinateur exécutait un seul programme à la fois

# Histoire des systèmes d'exploitation

- Elaborateurs de transistors (1955-1965)
  - Mainframes conservés sur place et par des techniciens spécialisés
  - Programmes écrits sur des cartes ou des bandes magnétiques
  - Premiers langages de programmation (par exemple, FORTRAN)
  - Les programmes ont été exécutés de manière séquentielle
  - Le système d'exploitation n'avait pour tâche que d'exécuter les programmes en séquence



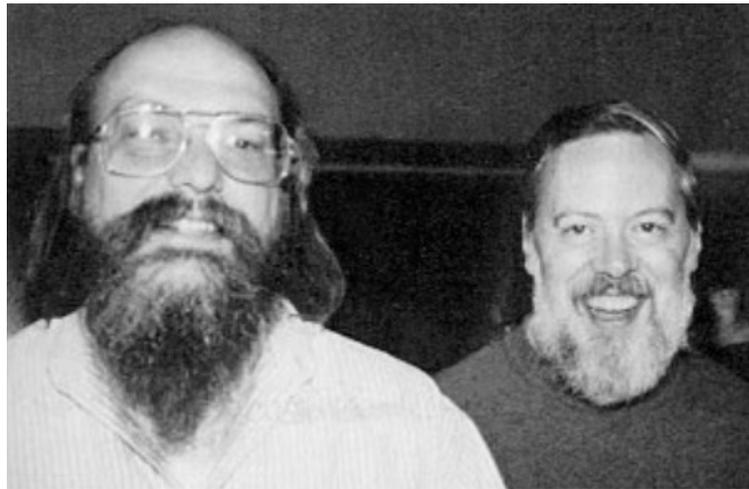
# Histoire des systèmes d'exploitation

- Circuits intégrés (1965-1980)
  - Des performances plus élevées. Prix et taille inférieurs
  - IBM crée la gamme d'ordinateurs IBM 360, avec son système d'exploitation OS/360
    - Introduction de la multiprogrammation : plusieurs processus fonctionnant simultanément
    - Encore utilisé aujourd'hui dans certains domaines
- Le MIT développe **MULTICS** en collaboration avec des partenaires industriels.
  - Conçu pour les cadres principaux très puissants
  - Permet l'utilisation par des centaines d'utilisateurs
  - Peu de succès



# Histoire des systèmes d'exploitation

- De MULTICS est né Unix dans les Bell Labs, en tant que projet personnel de Ken Thompson, Dennis Ritchie et d'autres
- C'est d'**UNIX** qu'est née toute la famille **Linux**



# Histoire des systèmes d'exploitation

- Ordinateurs personnels (de 1980 à aujourd'hui)
  - Grâce au développement des circuits intégrés, il était possible de produire des ordinateurs à bas prix
  - Le début des années 1980 a vu la naissance de Microsoft/DOS, initialement conçu pour les ordinateurs IBM équipés de CPU Intel (famille 8086)
  - Apple invente un système d'exploitation avec une GUI très réussie, dont Windows « s'inspirera »

# Histoire des systèmes d'exploitation

- Windows et Apple Mac OS poursuivent le développement jusqu'à présent.
- À partir des années 1990, l'essor des téléphones cellulaires (et smartphones après) entraîne l'émergence de systèmes d'exploitation dédiés. Ils sont nés :
  - Symbian : mort en 2011
  - Windows Mobile : mort en ???
  - Android
  - Mac OS

# Types de systèmes d'exploitation

- Différentes variétés de SO. Certains sont encore en vie, d'autres sont morts et enterrés
- **OS pour mainframes**
  - Pour les gros processeurs dans les grandes entreprises
  - Prise en charge de nombreux utilisateurs et ressources
  - En déclin au profit d'un système d'exploitation général (Linux)
  - Il existe toujours OS/390, descendant de l'OS/360 d'IBM

# Types de systèmes d'exploitation

- **SO pour PC**

- Ce sont les plus utilisés
- Basé sur une interface graphique
- Conçu pour un utilisateur unique (non expert)
- Exemples : Windows, MacOS

- **SO pour les serveurs**

- Pour les professionnels
- Souvent en mode "shell" seulement
- Ce sont des variantes de celles pour PC
- Exemples : Linux, Windows Server

# Types de systèmes d'exploitation

- **SO pour Smartphone ou tablette**
  - Entrée par interface graphique et tactile
  - Exemples : Android, MacOS
- **SO intégrés**
  - Pour les appareils ménagers, les véhicules
  - Ne pas accepter de programmes externes
- **SO pour les capteurs**
  - Sur des appareils aux ressources très limitées
  - Très léger et simple

# Types de systèmes d'exploitation

- **SO temps réel**

- Pour les applications spéciales où le temps est un facteur essentiel
  - Procédés industriels, aéronautique, automobile
- Certaines tâches doivent être accomplies en grande partie dans un délai donné
  - Une conception du système beaucoup plus compliquée

- **SO pour les smartcards**

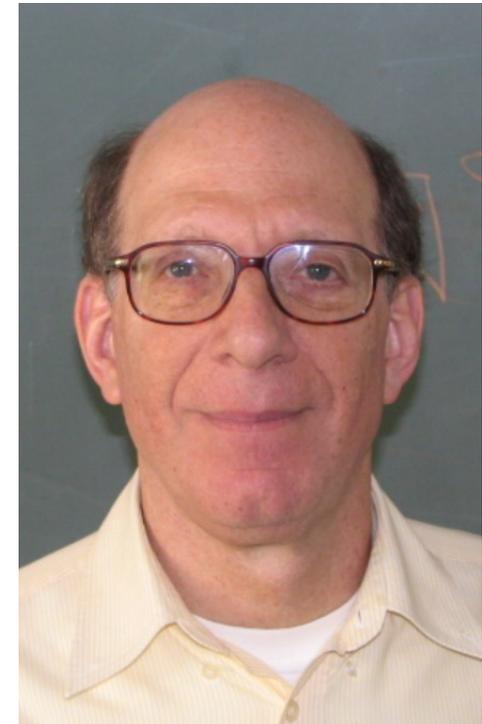
- Les cartes à puce (par exemple, les distributeurs automatiques de billets) possèdent un système de traitement et un système d'exploitation.
- Faible consommation et exigences de sécurité

# Unix

- Nous avons dit que Unix est né dans les années 1970 de MULTICS
- Plusieurs variantes sont nées dans les années 1980 et ont été normalisées
  - Norme ISO C - 1972
  - Standard Posix - 1988
- Toutes les versions étaient payantes, sous le contrôle d'AT&T (Bell)
  - Le code était fermé, très long et complexe

# Minix

- Créé par Andrew Stuart Tanenbaum
  - Un des auteurs du livre recommandé pour ce cours
- C'est un **clone** d'UNIX :
  - Open-Source
  - Un micro-noyau
  - Conçu pour l'éducation
  - Ne convient pas pour être un véritable OS



# Linux

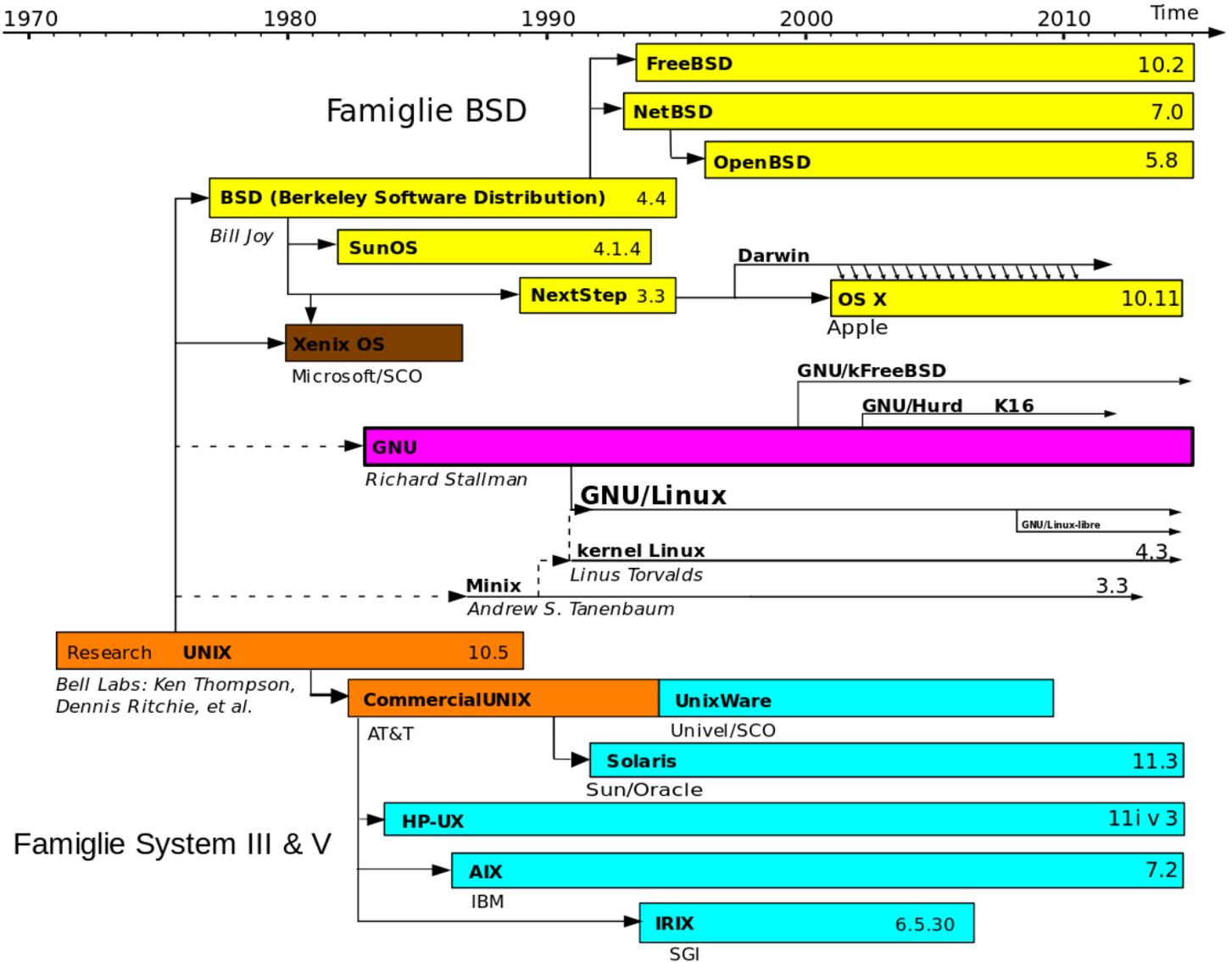
- En 1991, Linus Torvald a créé le noyau **Linux** :
  - Développé à partir de Minix
  - Objectif : Être un véritable système d'exploitation (pas seulement à des fins éducatives)
  - De nombreuses distributions : Ubuntu, Debian, Fedora, etc



# En résumé

- Unix : système d'exploitation développé dans les années 1980 chez AT&T
- Linux : un noyau de type Unix développé par Linus Torvald depuis 1991
- GNU : Système d'exploitation de type Unix (noyau exclu). Il peut fonctionner avec différents noyaux

# Les fils d'UNIX



# Une question pour compléter le cours

- Linux est-il un système d'exploitation ?
- Et POSIX ?