

LES SYSTÈMES D'EXPLOITATION

INTRODUCTION AUX SYSTÈMES D'EXPLOITATION

🎓 3A - Cursus Ingénieurs - Dominante Informatique et Numérique
🏛️ CentraleSupélec - Université Paris-Saclay - 2025/2026



Idir AIT SADOUNE
idir.aitsadoune@centralesupelec.fr

PLAN

- Architecture des ordinateurs
- Qu'est-ce qu'un système d'exploitation
- Rôles du système d'exploitation
- Structure d'un Système d'exploitation
- Synthèse

[Retour au plan](#) - [Retour à l'accueil](#)

PLAN

- Architecture des ordinateurs
- Qu'est-ce qu'un système d'exploitation
- Rôles du système d'exploitation
- Structure d'un Système d'exploitation
- Synthèse

[Retour au plan](#) - [Retour à l'accueil](#)

L'INFORMATIQUE

- L'informatique est la science du **traitement automatique de l'information**.
- Le traitement automatique de l'information s'effectue avec des **programmes informatiques** exécutés par des **machines**
 - **les programmes (software)** décrivent le traitement à réaliser,
 - **les machines (hardware)** exécutent **les programmes**.



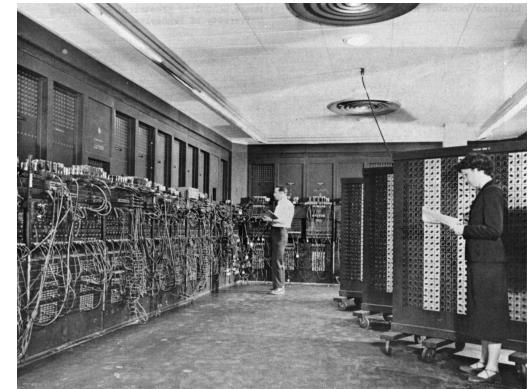
LA NOTION D'ORDINATEUR



- L'ordinateur désigne **un équipement informatique** permettant de traiter des informations en **exécutant des instructions**.
 - ➡ On lui donne **des instructions** (programme/logiciel)
 - ➡ On lui donne **des données** (information)
 - ➡ Il **transforme** les données

ENIAC - 1946

- Construit de 1943 à 1946 par John Mauchley et John Eckert à l'université de Pennsylvanie
- Premier ordinateur entièrement électronique (utilise des tubes à vide).
- Programmé pour résoudre tous les problèmes calculatoires.



HP 3000 - 1972

- Le **mini-ordinateur** a été une innovation des années **1970**.
- L'intégration de **circuits intégrés à grande échelle** conduit au développement des **micro-processeurs**.



APPLE II - 1977

- Un des premiers **ordinateurs personnels** à **micro-processeur** fabriqué à grande échelle
- Conçu par **Steve Wozniak**, commercialisé le **10 juin 1977** par **Apple**



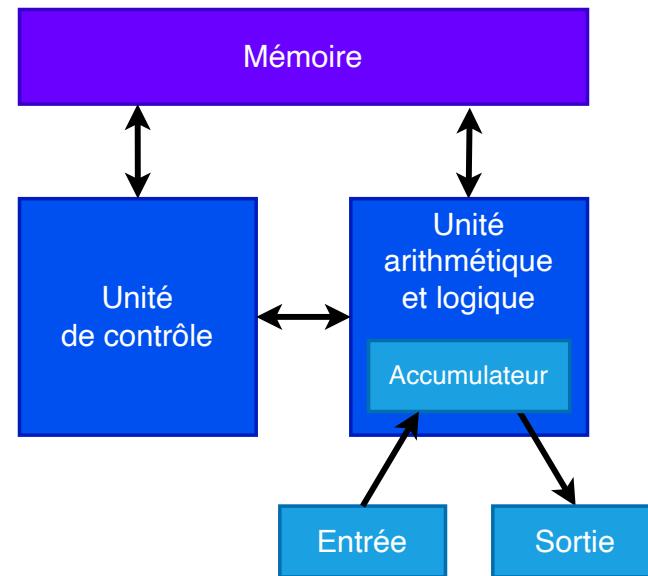
LES ORDINATEURS D'AUJOURD'HUI

- System on a Chip (**SOC**) : un système complet embarqué dans une puce (**circuit intégré**).
- Un **circuit intégré** peut comprendre :
 - un ou plusieurs microprocesseurs
 - de la mémoire
 - des périphériques d'interface
 - ou tout autre composant

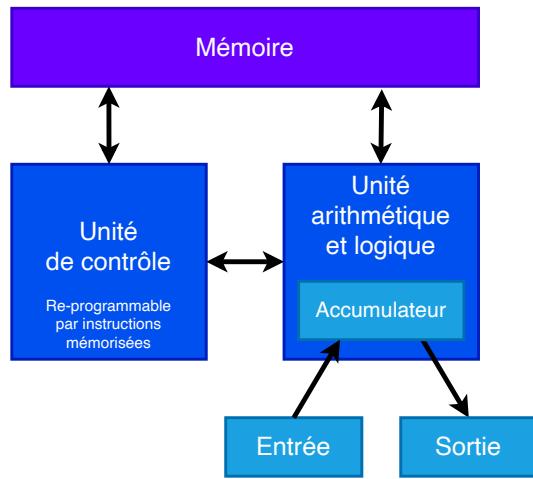


L'ARCHITECTURE DE VON NEUMANN

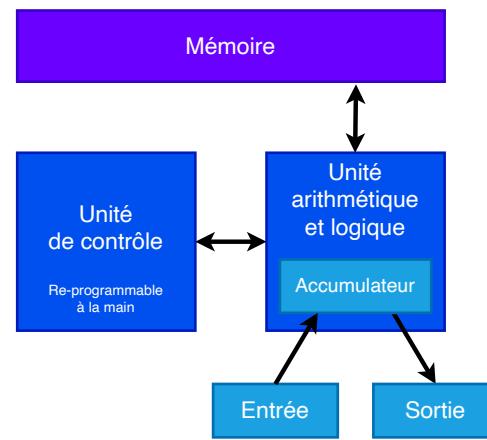
- L'architecture de von Neumann :
un **modèle** pour un ordinateur avec
une **mémoire unique** pour conserver
 - les instructions
 - et les données



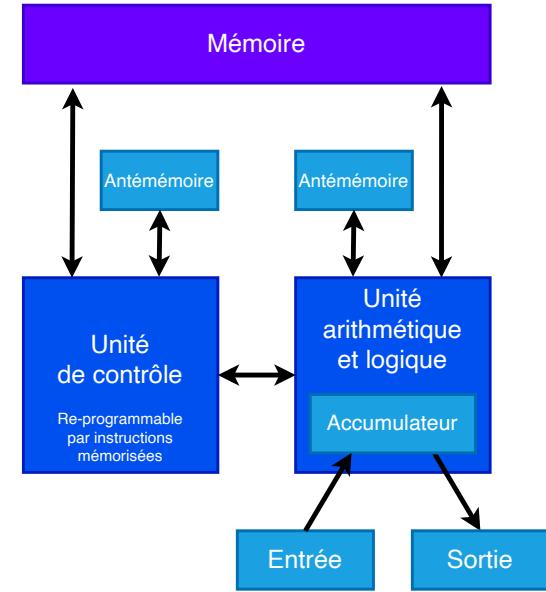
VON NEUMANN / L'ORDINATEUR MODERNE



Von Neumann



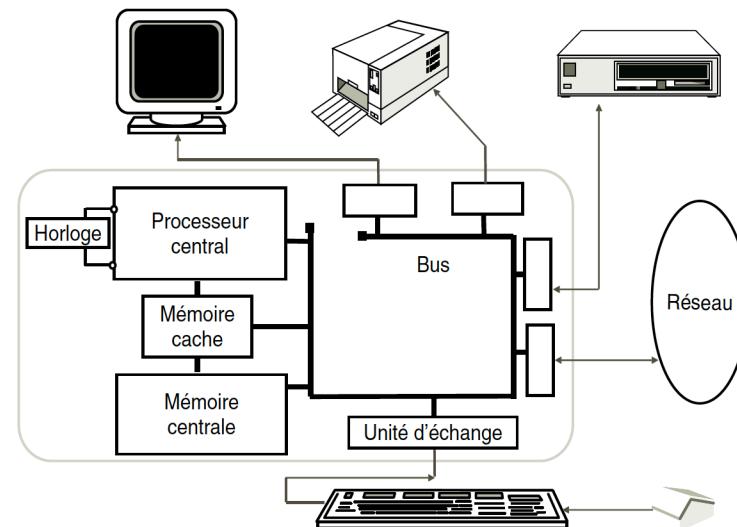
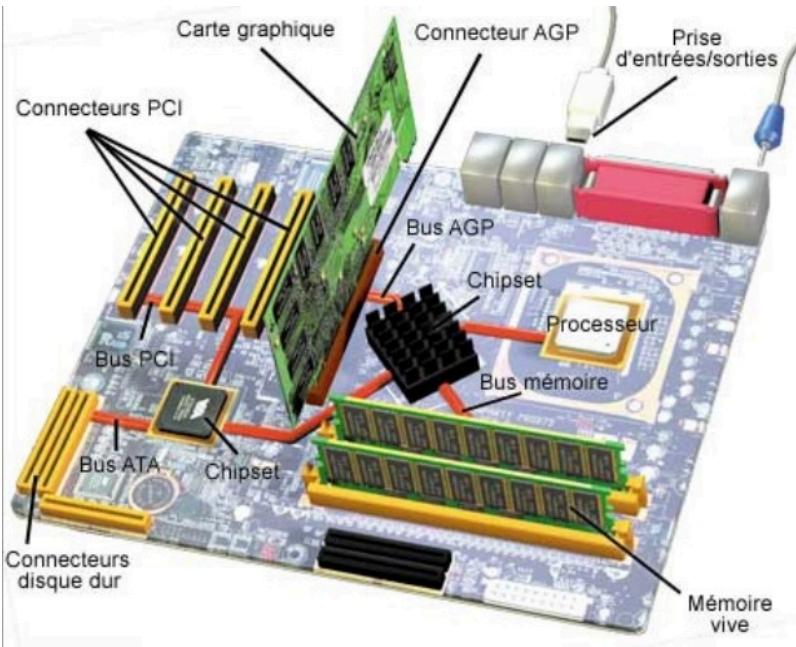
ENIAC



Ordinateur personnel

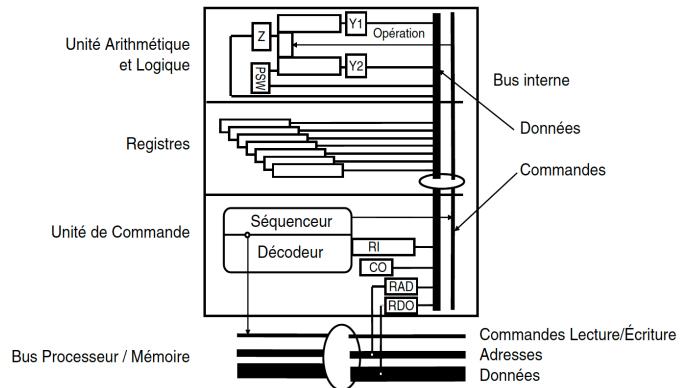
source : [la thèse d'Alexandre Brunet](#)

STRUCTURE GÉNÉRALE D'UN ORDINATEUR



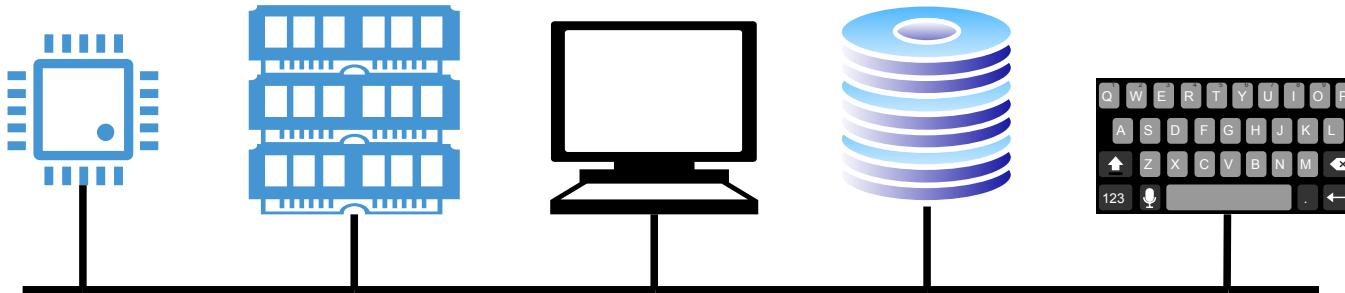
L'ARCHITECTURE D'UN MICROPROCESSEUR

- Le microprocesseur (**CPU**) exécute les instructions machines placées en mémoire centrale.
- Le **CPU** est constitué de quatre parties
 1. l'unité arithmétique et logique (**UAL**),
 2. les registres,
 3. l'unité de commande,
 4. le bus de communication interne.



LE FONCTIONNEMENT DE L'ORDINATEUR

Comment fonctionne un ordinateur ?



Tout cela n'est que des fils électriques ...
... qu'on allume et qu'on éteint.

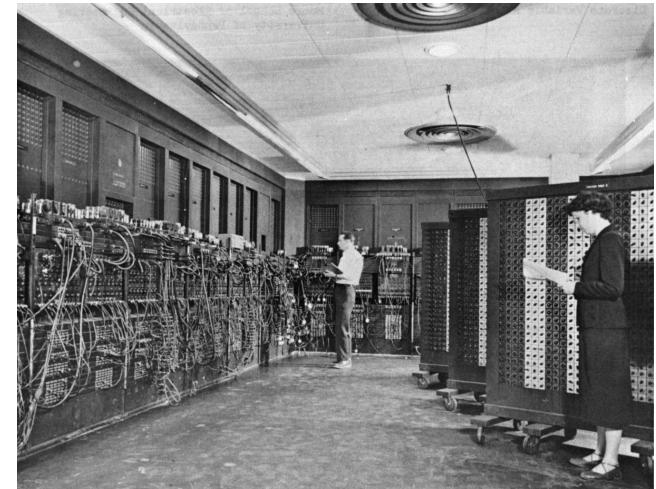
PLAN

- Architecture des ordinateurs
- Qu'est-ce qu'un système d'exploitation
- Rôles du système d'exploitation
- Structure d'un Système d'exploitation
- Synthèse

[Retour au plan](#) - [Retour à l'accueil](#)

AUTREFOIS : ENIAC

- Premier ordinateur **entièrement électronique** :
 - 18 000 tubes à vide
 - 1 500 relais
 - 20 registres de 10 chiffres décimaux
 - programmé à l'aide de 6 000 commutateurs
- **La programmation** se faisait directement **en langage machine**
- **Un seul programme** à la fois pouvait s'exécuté.
- L'absence d'un OS obligeait le programmeur à **charger manuellement le programme**

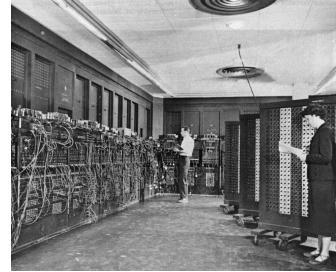


AUTREFOIS : IBM RAMAC 305



- Premier ordinateur à disque dur (l'**IBM 350**) commercialisé en **septembre 1956** par **IBM**.
- Composé des éléments suivants : unité de traitement, imprimante, console, alimentation, disque dur, mémoire 5Mo.
- L'**unité de traitement** est basée sur un tambour magnétique sur lequel est stocké le programme.
- Un opérateur programme à l'aide de **cartes perforées** et inscrit les données sur le tambour.

AUTOMATISER LES TÂCHES



- Comment **automatiser les tâches** des opérateurs et des programmeurs ?
- Écrire un **programme informatique** qui:
 - ➡ décide qui fait quoi et à quel moment
 - ➡ fait le lien entre les applications et le matériel

DÉFINITION

“Un système d'exploitation est un ensemble de programmes réalisant l'interface entre le matériel et les utilisateurs.”

- gère la partie **matérielle**
- sert de socle pour les **applications**

Applications



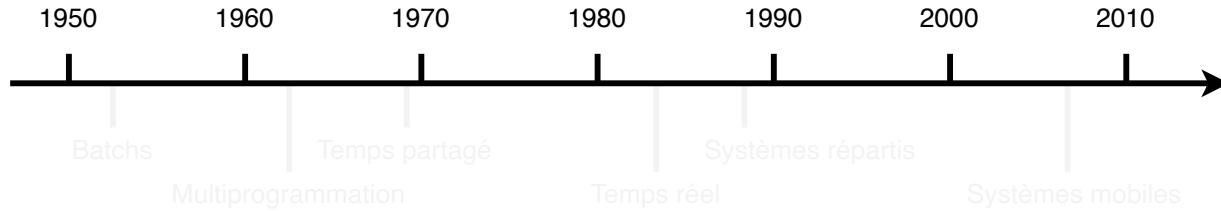
Système d'exploitation



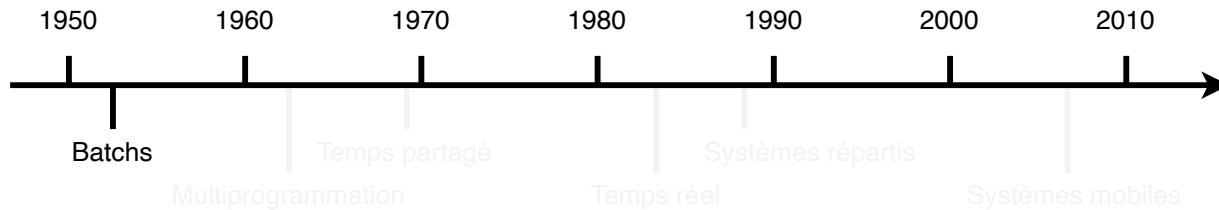
Matériel



HISTORIQUE/TYPES DES OS

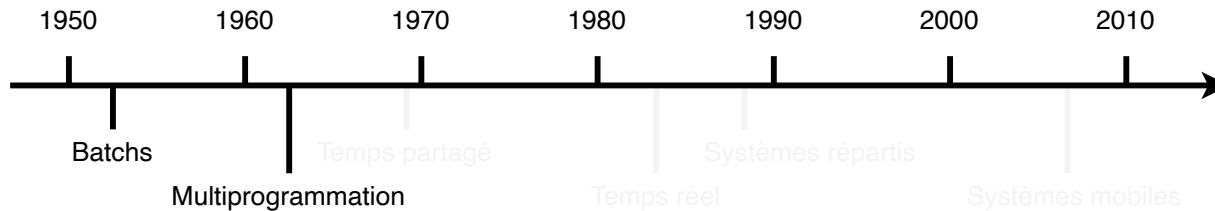


HISTORIQUE/TYPES DES OS



- **Les systèmes batch** sont basés sur deux programmes :
 1. **le chargeur** : charger les programmes dans la mémoire centrale depuis les cartes perforées ou le dérouleur de bandes.
 2. **le moniteur d'enchaînement de traitements** : permettre l'enchaînement des travaux soumis à la place de l'opérateur.
- **Les systèmes batch** automatisent les tâches de préparation des travaux et exploitent efficacement le processeur.

HISTORIQUE/TYPES DES OS

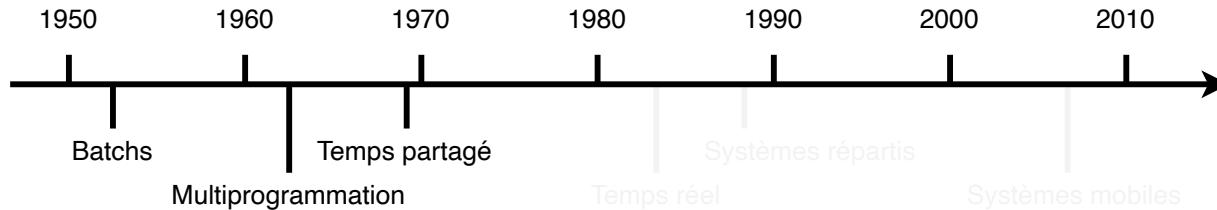


Utiliser plusieurs composants en parallèle, ce qui nécessite:

- **Gestion de la priorité** (*quel processus peut accéder à la ressource*)
➡ ordonnancement
- **Mémoire partagée** (*gérer des informations de plusieurs processus*)
➡ adressage et mémoire

Exemple : MULTICS

HISTORIQUE/TYPES DES OS

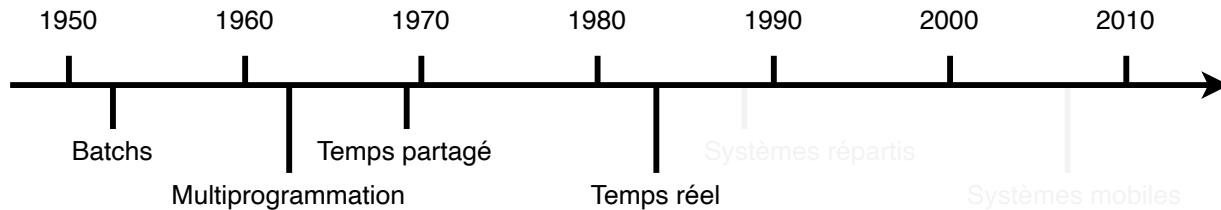


Plusieurs processus actifs alternant sur le processeur

- Gestion des interruptions
- Cycle de vie du processus
- Synchronisation de processus et programmation concurrente

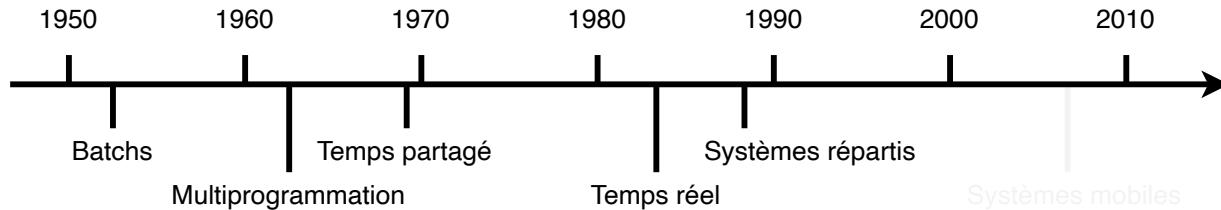
Exemple : UNICS ou UNIX

HISTORIQUE/TYPES DES OS



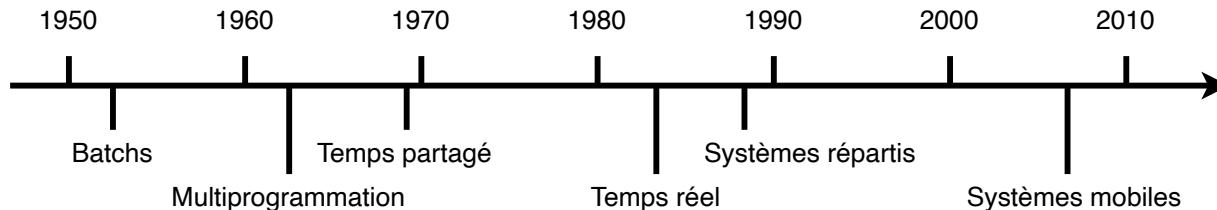
- **Gestion des délais:** contrainte de **temps de réponse**
 - ➡ les processus doivent répondre vite
- Apparition des **micro-ordinateurs**
 - CP/M → IBM PC (**MSDOS**)
- Apparition des **interfaces graphiques**
 - Xerox → Apple Macintosh 1984, Windows 95, Linux 1991

HISTORIQUE/TYPES DES OS



- **Les ordinateurs communiquent** pour échanger des données !
 - Arpanet (1967) conçu par la DARPA
 - E-mail (1972) avec Ray Tomlinson
 - TCP/IP(1972)
 - Clients-Serveur années 80 → NFS - Network File System (Sun, 1984)
 - Arpanet ouvert fin 80 → Web début 90 (CERN , Tim Berners-Lee)

HISTORIQUE/TYPES DES OS



- **Les ordinateurs de poche** existent depuis **les années 80**
 - **1986** : sortie des **PDA** → **PalmOS**
 - **2007** : sortie des **smartphones** → **android OS**
 - **2007** : sortie de l'**iPhone** → **iOS**

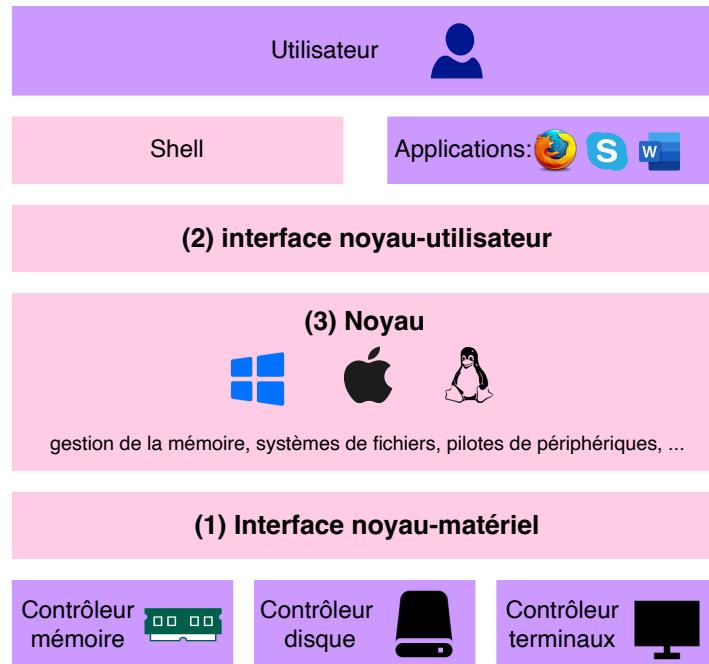
PLAN

- Architecture des ordinateurs
- Qu'est-ce qu'un système d'exploitation
- Rôles du système d'exploitation
- Structure d'un Système d'exploitation
- Synthèse

[Retour au plan](#) - [Retour à l'accueil](#)

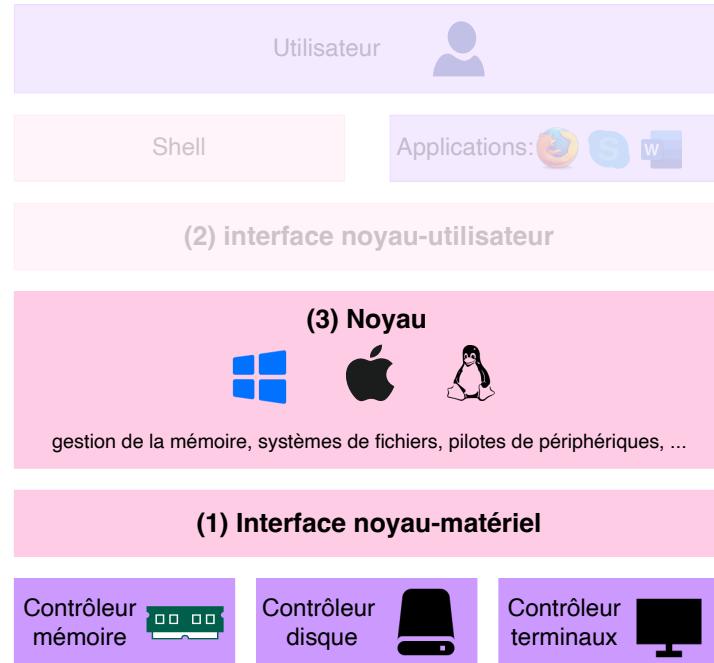
RÔLES DU SYSTÈME D'EXPLOITATION

1. **L'interface noyau-matériel** prend en charge la gestion et le partage des ressources de la machine.
2. **L'interface noyau-utilisateur** construit une machine virtuelle plus facile d'emploi et plus conviviale.
3. **Le noyau** assure plusieurs grandes fonctionnalités.



INTERFACE NOYAU-MATÉRIEL

- Gérer l'accès et le partage des ressources matérielles (arbitrage).
 - processeur
 - mémoire centrale
 - périphériques
 - ...
- Cet arbitrage doit assurer:
 - l'équité d'accès aux ressources
 - la protection de l'accès aux ressources
 - la cohérence des états des ressources

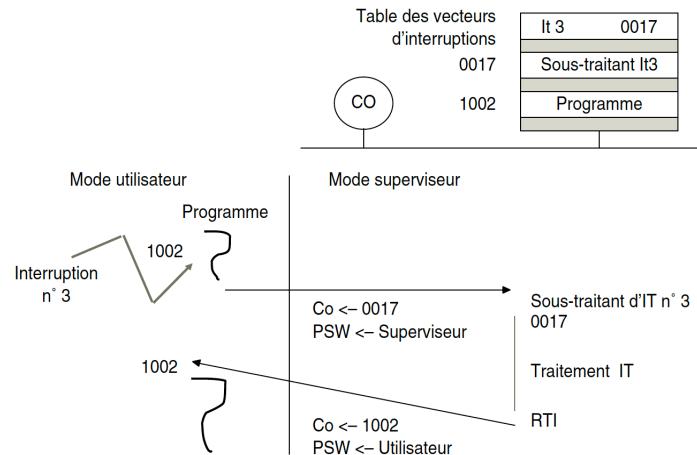


INTERRUPTION - IRQ

- L'OS s'interface avec la couche matérielle, par le biais du mécanisme des interruptions (Interrupt ReQuest ou IRQ).
 - ➡ prendre connaissance des événements survenant sur le matérielle
- L'IRQ est un signal (code) permettant à un dispositif externe d'interrompre le processeur pour lancer un traitement particulier.
 - à chaque code correspond une routine de traitement de l'OS.
 - les adresses des routines sont dans une table placée en mémoire (la table des vecteurs d'interruptions).
 - les routines d'interruptions sont chargées en mémoire au moment du chargement de l'OS et exécutées en mode superviseur.

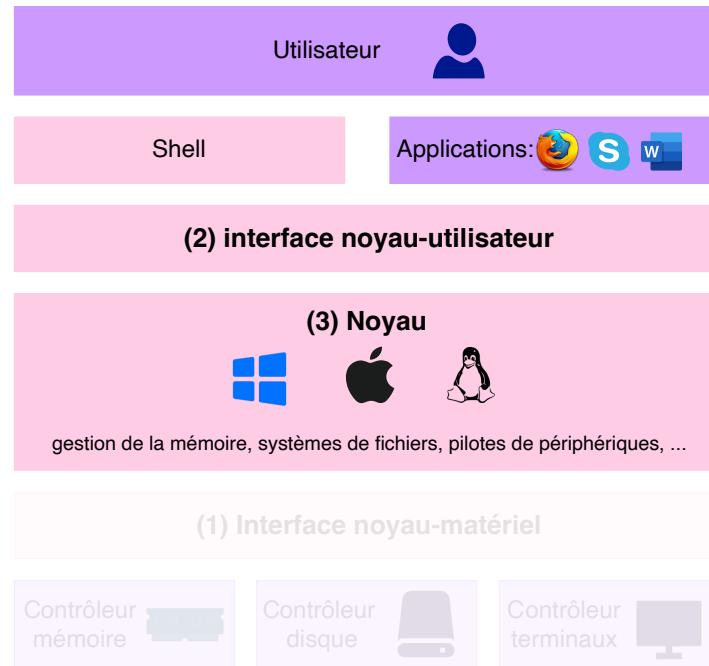
PRISE EN COMPTE D'UNE IRQ

- Enregistrer → **pile de l'OS**
 - l'adresse d'**instruction interrompue**
 - l'**état du processeur** (registres)
- Passer en **mode superviseur**
- Charger la **routine** correspondant à **l'interruption**



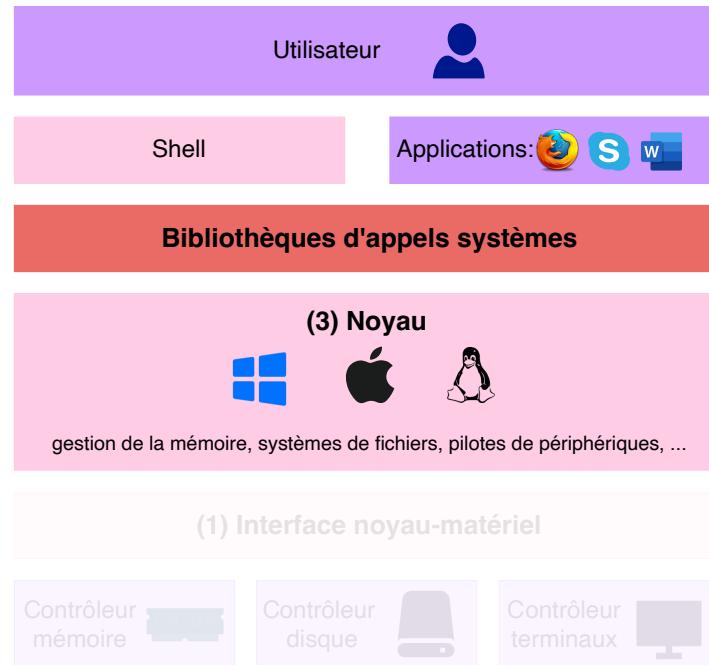
INTERFACE NOYAU-UTILISATEUR

- Présenter une **interface** entre le hardware et les applications.
➡ une **interface simplifiée et unifiée**.
- Présenter au-dessus de la machine physique, **une machine virtuelle** plus simple et plus conviviale.
- Créer **l'illusion de vrais ressources physiques** (processeur, mémoire, périphérique ...).



LES APPELS SYSTÈMES

- Fournir une **interface d'accès** aux ressources matérielles.
 - ➡ par le biais de **fonctions prédéfinies** (**appels/routines systèmes**).
 - ➡ les points d'entrées aux **fonctionnalités de l'OS**.



EXEMPLES D'APPELS SYSTÈMES

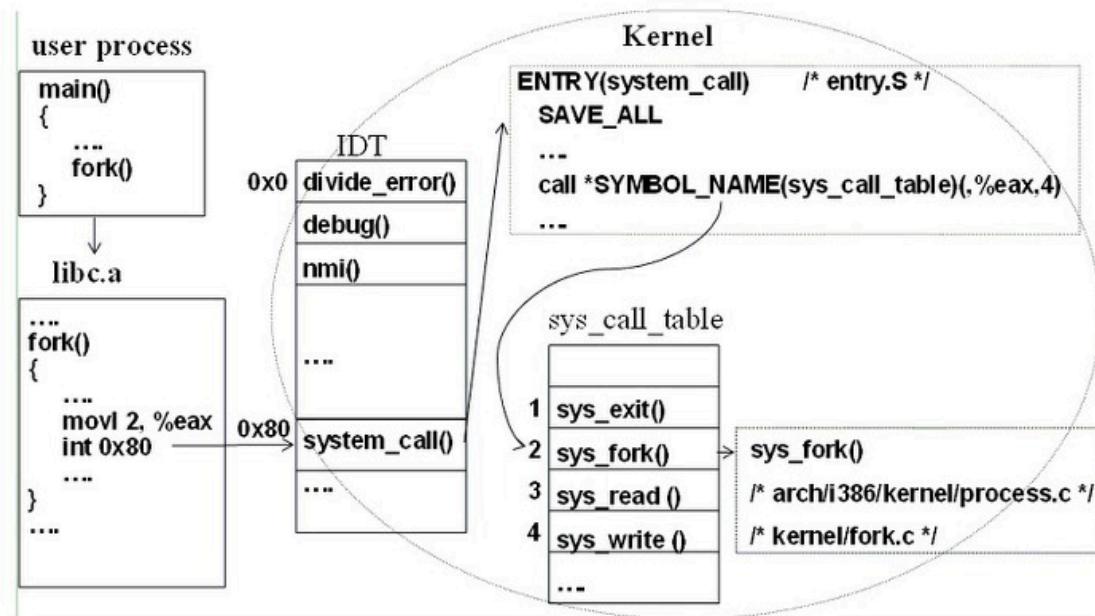
- **Contrôle de processus**

- `sys_fork` : créer un processus
- `sys_wait` : attendre la terminaison d'un processus
- `sys_exit` : terminer l'exécution d'un processus
- `sys_kill` : Envoyer un signal à un processus

- **Gestion des fichiers**

- `sys_open/sys_close` : ouvrir/fermer un fichier
- `sys_read/sys_write` : lire/écrire des données dans un fichier
- `sys_mkdir/sys_rmdir` : créer/supprimer un répertoire

L'APPEL SYSTÈME `fork()`

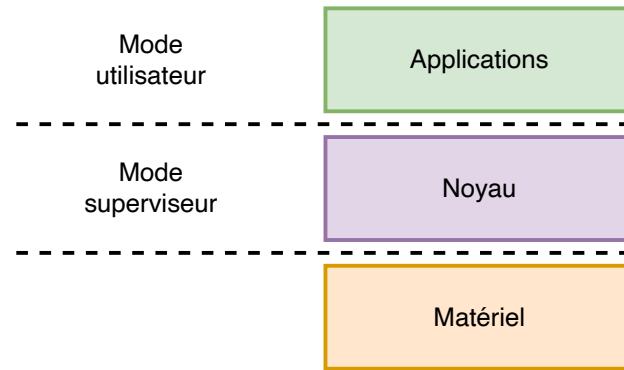


EXAMPLE SOUS UNIX

- L'instruction `os.chdir(path)` permet de changer le répertoire courant d'un programme **Python** en cours d'exécution.
- La commande `cd path` permet de changer le répertoire courant depuis l'**interpréteur de commandes (Shell)**.
- Les deux exécutent la routine système `sys_chdir`.

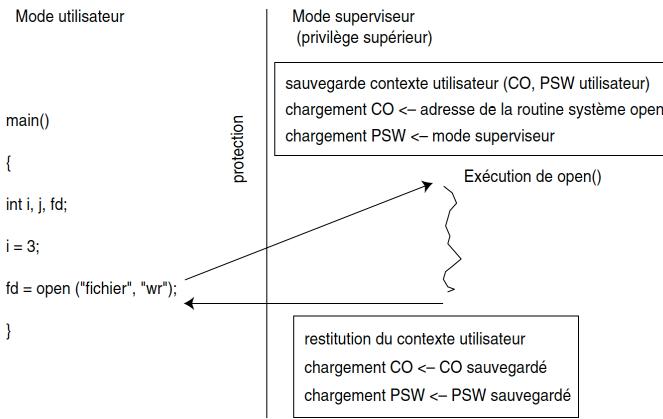
MODES D'EXÉCUTIONS

- Un programme utilisateur s'exécute dans un mode utilisateur :
 - un Jeu d'instructions restreint pour protéger la machine.
 - ex. manipulation des IRQs interdite.
- L'OS s'exécute dans un mode privilégié (mode superviseur):
 - aucune restriction de droits n'existe.



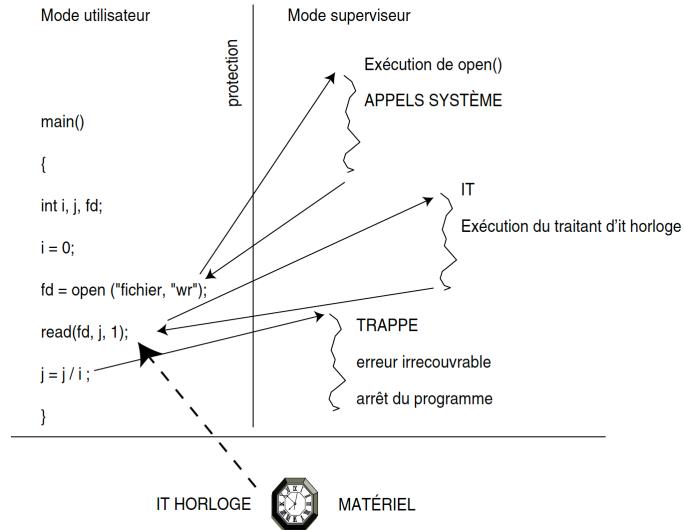
COMMUTATIONS DE CONTEXTE

- A l'appel d'une fonction du noyau,
il y a passage au **mode superviseur**
(commutation de contexte).
- A la fin de l'exécution de la fonction
du noyau, le programme repasse
au **mode utilisateur**.
 - commutation de contexte avec **restauration du contexte utilisateur**.
 - reprise de l'exécution du programme utilisateur



LES CAUSES DE COMMUTATIONS DE CONTEXTE

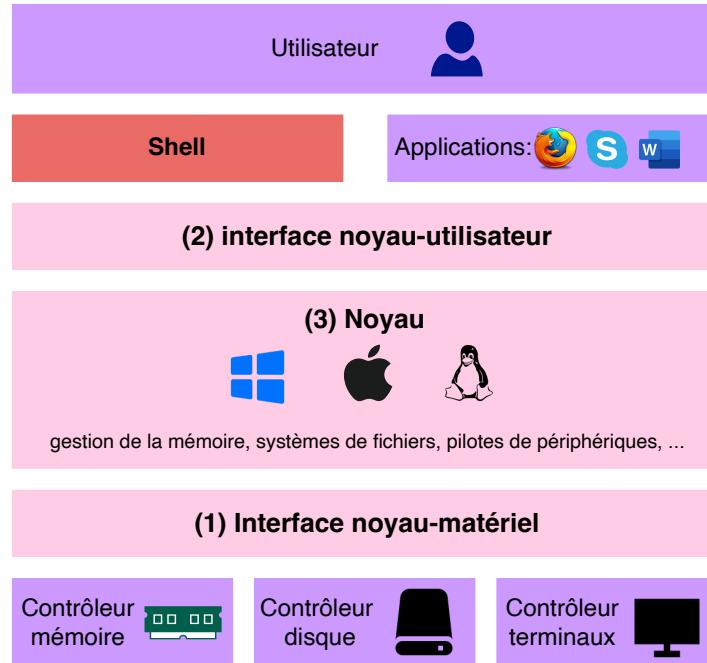
1. appelle d'une **fonction système**.
2. exécute une opération illicite (**trappe** ou **exception**).
3. prise en compte d'une **interruption matérielle**.



Trappes et appels systèmes sont parfois qualifiés d'**interruptions logicielles** par opposition aux interruptions matérielles.

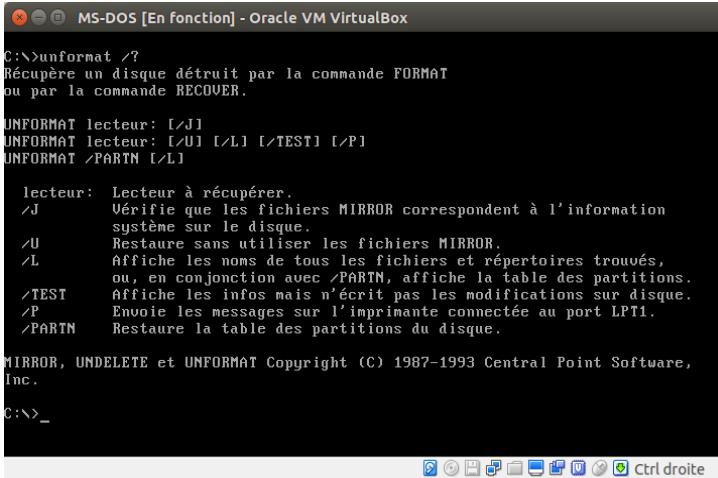
INTERPRÉTEUR DE COMMANDE (SHELL)

- **Langage de commandes :**
l'interface de niveau utilisateur avec le système d'exploitation.
- **Interpréteur de commandes :**
exécuter **des commandes de l'utilisateur** en appellant **la routine système** appropriée.



INTERPRÉTEUR DE COMMANDE (SHELL)

- Chaque système d'exploitation a son propre **langage de commandes** :
 - **MSDOS/Unix** : console + clavier
 - **Mac/Windows** : souris + clavier
 - **iOS/Android** : boutons + écran tactile



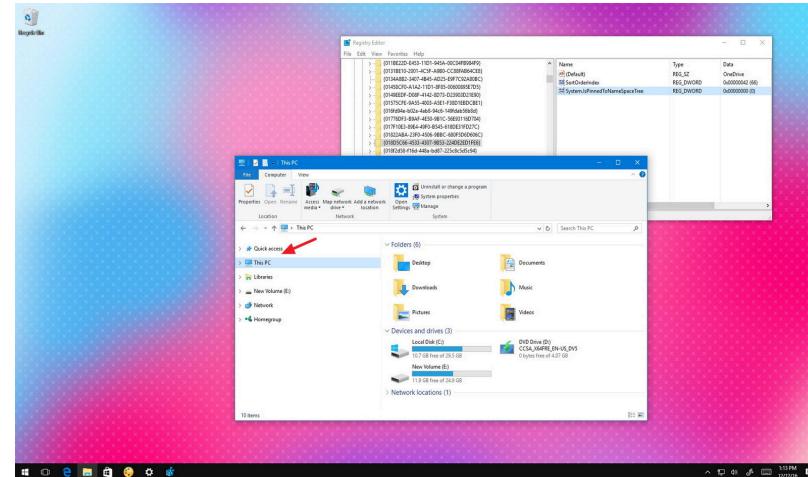
```
C:\>unformat /?
Récupère un disque détruit par la commande FORMAT
ou par la commande RECOVER.

UNFORMAT lecteur: [/J]
UNFORMAT lecteur: [/U] [/L] [/TEST] [/P]
UNFORMAT /PARTN [/L]

lecteur: Lecteur à récupérer.
/J Vérifie que les fichiers MIRROR correspondent à l'information
      système sur le disque.
/U Restaure sans utiliser les fichiers MIRROR.
/L Affiche les noms de tous les fichiers et répertoires trouvés,
      ou, en conjonction avec /PARTN, affiche la table des partitions.
/TEST Affiche les infos mais n'écrit pas les modifications sur disque.
/P Envoie les messages sur l'imprimante connectée au port LPT1.
/PARTN Restaure la table des partitions du disque.

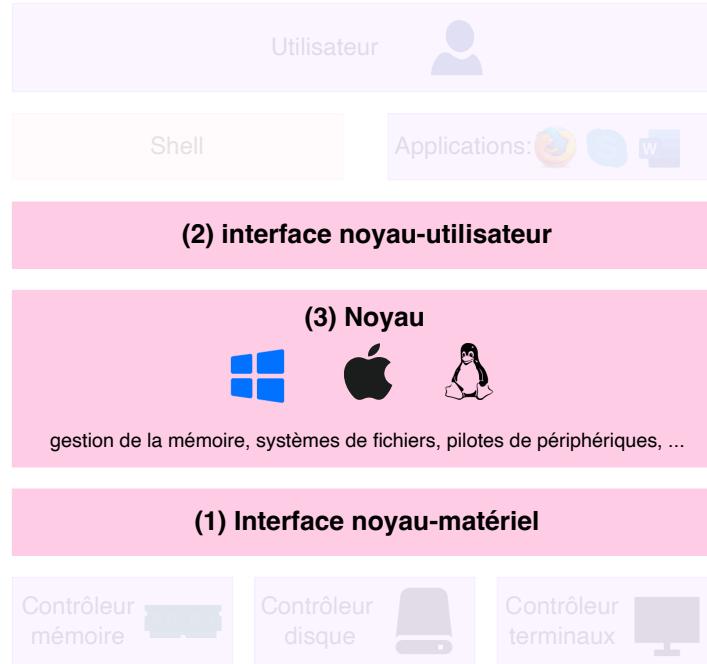
MIRROR, UNDELETE et UNFORMAT Copyright (C) 1987-1993 Central Point Software,
Inc.

C:\>_
```



NOYAU D'UN SYSTÈME D'EXPLOITATION

- **Gestion des entrées/sorties (I/O)**
 - contrôleurs, pilotes, ...
- **Gestion des processus**
 - ordonnancement, synchronisation, ...
- **Gestion mémoire**
 - allocation, gestion des espaces, ...
- **Gestion du stockage secondaire**
 - système de fichiers, ...
- **Gestion de la sécurité**



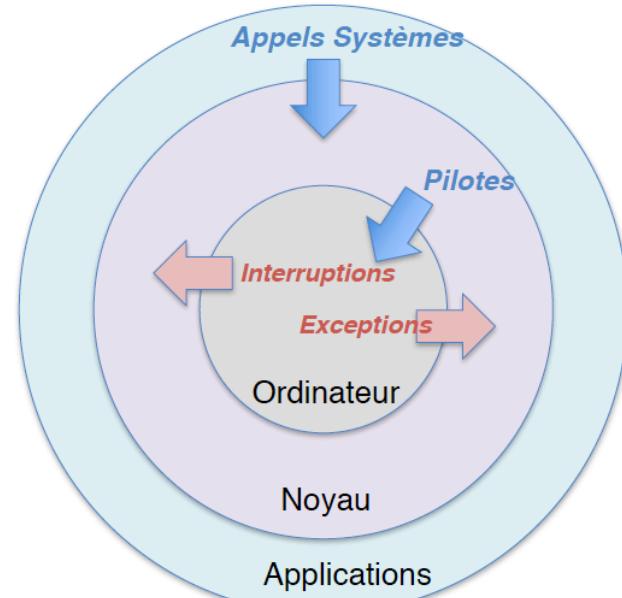
PLAN

- Architecture des ordinateurs
- Qu'est-ce qu'un système d'exploitation
- Rôles du système d'exploitation
- Structure d'un Système d'exploitation
- Synthèse

[Retour au plan](#) - [Retour à l'accueil](#)

ORGANISATION GÉNÉRALE DE L'OS

- **Interruptions** : évènements produits par le matériel.
- **Exceptions** : évènements générés par le processeur.
- **Pilotes (drivers)** : applications contrôlant les périphériques.
- **Noyau (kernel)** : application rendant des services généraux.
- **Appels Systèmes** : demandes de services.

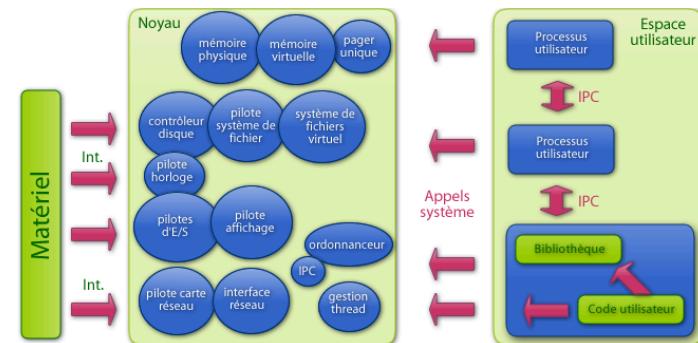


STRUCTURE DES OS

- Comment organiser les différentes fonctions d'un **OS** ?
 - ➡ Qu'est-ce qui est dans le **noyau** (en **mode Superviseur**) ?
 - ➡ Comment interagissent les **différents composants** ?

NOYAUX MONOLITHIQUES

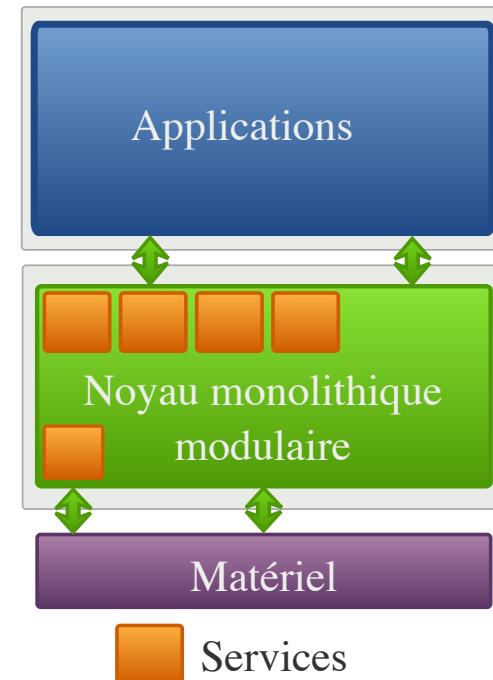
- L'ensemble des fonctions/pilotes sont regroupés dans **un seul bloc**.
- **Ex.** anciennes versions de **Linux** ou certains **vieux Unix**.
- Les OS **monolithiques** sont rapides mais délicats à maintenir.



source : <https://fr.wikipedia.org>

NOYAUX MONOLITHIQUES MODULAIRES

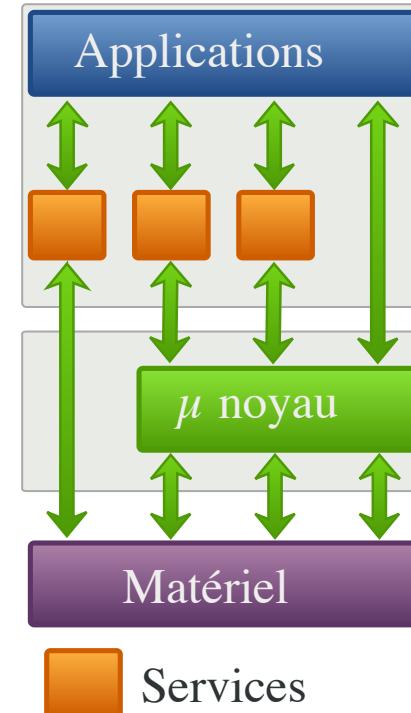
- Seules **les parties fondamentales** de l'OS sont regroupées dans **un bloc unique**.
- Les autres fonctions (**ex.** les pilotes) sont regroupées dans des **modules séparés**.
- **Ex.** Linux ou **Solaris**.
- Les OS **monolithiques modulaires** ne sont pas faciles à concevoir (dépendances multiples).



source : <https://fr.wikipedia.org>

SYSTÈMES À MICRO-NOYAUX

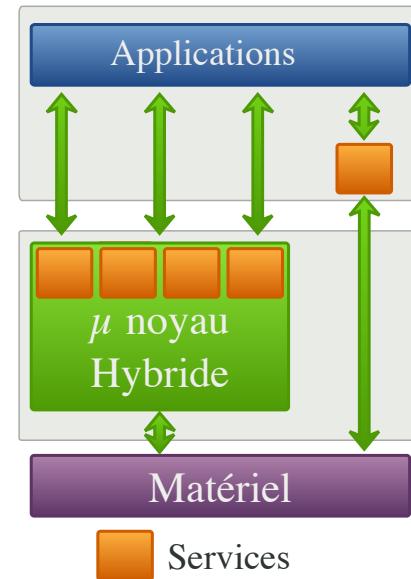
- **Minimiser les fonctionnalités dépendantes** du noyau en plaçant des services l'extérieur.
- **Éloigner les services « à risque »** des parties critiques de l'OS regroupées dans le noyau.
- **Ex. Mach** de **Mac OS X**.
- Les OS à **micro-noyaux** pur sont trop lents.



source : <https://fr.wikipedia.org>

SYSTÈMES À NOYAUX HYBRIDES

- Combiner des noyaux **monolithiques** et des **micro-noyaux** pour avoir les avantages des deux.
- Ex. **XNU** de **Mac OS X**.



source : <https://fr.wikipedia.org>

Les OS avec un micro-noyau étendus en fonctionnalités par d'autres composants sont les plus utilisés.

CHARGEMENT D'UN OS

- L'**OS** est le **premier programme exécuté** lors de la mise en marche de l'ordinateur, après l'amorçage (**boot**).
- Le **boot** (**bootstrap**) désigne les **étapes successives du démarrage**.

LES ÉTAPES DU BOOT

1. le **POST** test - **Power On Self Test**

- après un start ou un reset, le processeur charge **les premières instructions** à partir de la **ROM** du **BIOS** situées à l'adresse **FFFF0**.
- des instructions de **branchement vers un programme** du **BIOS** qui **initialise et teste les fonctions vitales** du hardware

2. le chargement du **MBR** - **Master Boot Record**

- si le **POST** réussit, il consultera la **RAM CMOS** pour identifier le **disque système** dont le premier secteur est appelé **MBR**.
- le code du **MBR** teste la table de partition pour charger la partition contenant le secteur d'amorçage avec l'**IPL** - **Initial Program Load**.
 - ➡ l'**IPL** charge l'**OS** ou le **bootmanager** en RAM.
 - ➡ l'**OS** est lancé

PLAN

- Architecture des ordinateurs
- Qu'est-ce qu'un système d'exploitation
- Rôles du système d'exploitation
- Structure d'un Système d'exploitation
- Synthèse

[Retour au plan](#) - [Retour à l'accueil](#)

SYNTHÈSE

- Un système d'exploitation est un ensemble de programmes réalisant l'interface entre le matériel et les utilisateurs.
- Les deux objectifs principaux de cette interface sont :
 1. construire au-dessus du matériel d'une machine virtuelle plus facile d'emploi et plus conviviale (accessible par des **appels système**);
 2. prendre en charge de la gestion de plus en plus complexe des ressources et le partage de celles-ci (gestion basée sur les **interruptions**).
- Les fonctionnalités du système d'exploitation sont accessibles par le biais des commandes ou des appels système.

SYNTHÈSE

- Le **mode superviseur** est le mode d'exécution du noyau du système d'exploitation.
- Le passage du **mode utilisateur** vers le mode superviseur peut être provoqué par un appel système, une trappe, ou par une **IRQ**.
- Il s'accompagne d'une commutation de contexte qui consiste à :
 - sauvegarder le contexte utilisateur
 - changer le mode d'exécution
 - restituer le contexte utilisateur

MERCI

[Version PDF des slides](#)

[Retour à l'accueil](#) - [Retour au plan](#)