



ARCHITECTURES DES SYSTÈMES D'EXPLOITATION



Présentation de l'Enseignant- Formateur :

M. Ndiamé CAMARA

Ingénieur en Informatique industrielle

Administrateur Systèmes

Présentation des séquences

Séquence 4 : ARCHITECTURE DES SYSTEMES D'EXPLOITATION

- Expliquer les composantes des systèmes d'exploitation et leur interaction
- Sélectionner le SE adéquat par rapport aux besoins

DUREE: 4 heures



Prérequis et Consignes

Prérequis :

- Architecture des ordinateur
- Historique de l'informatique

Consignes pour l'apprenant :

- Télécharger le document de présentation du cours
- Faire l'évaluation formative
- Animer le forum de partage
- Effectuer les projets de classe



Architectures des systèmes d'exploitation

Modes d'exécution des processus et appels systèmes

Noyau à architecture monolithique

Architecture à micronoyau

Architecture à micronoyau hybride

Architecture à micronoyau temps réel

Architecture en (6) couches des systèmes d'exploitation

Architecture Client-serveur

Compilation des programmes et édition des liens

Architectures des systèmes d'exploitation

Modes d'exécution des processus et appels systèmes

Mode superviseur

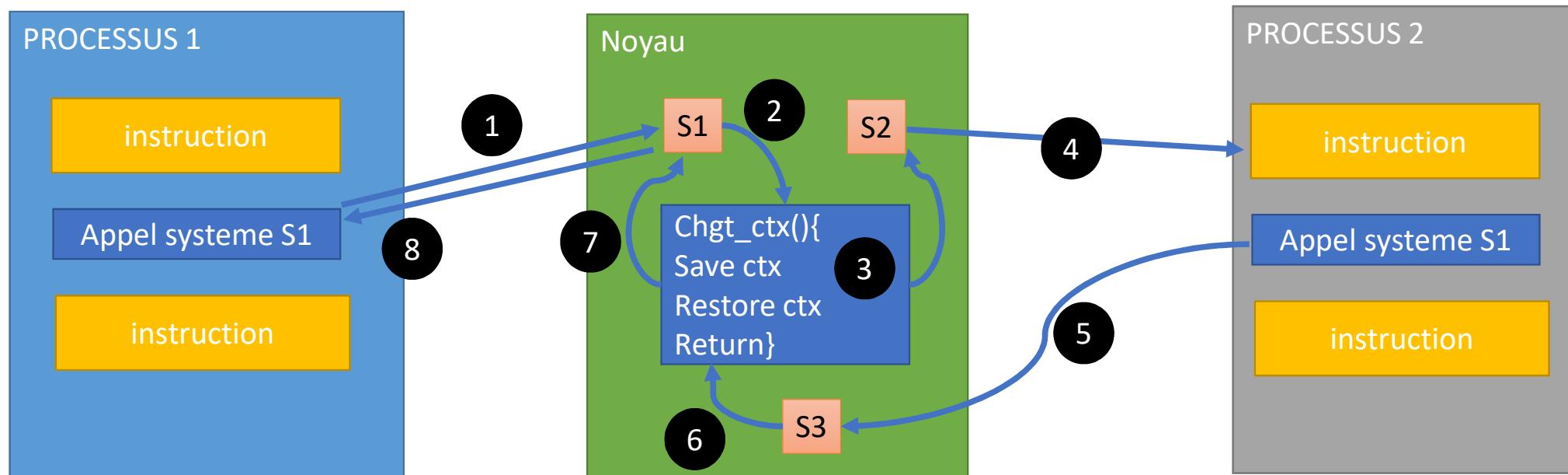
- Le mode protégé ou superviseur (également appelé mode noyau) est réservé à l'exécution des **primitives** du système d'exploitation.
- L'accès aux différentes ressources de l'ordinateur leur est réservé.

Mode utilisateur

- Le mode utilisateur dans lequel les programmes utilisateurs sont exécutés.
- Dans ce mode certaines instructions sont interdites. Ainsi, les programmes sont contraints à effectuer des **appels systèmes** aux primitives pour certaines opérations.

Architectures des systèmes d'exploitation

Modes d'exécution des processus et appels systèmes





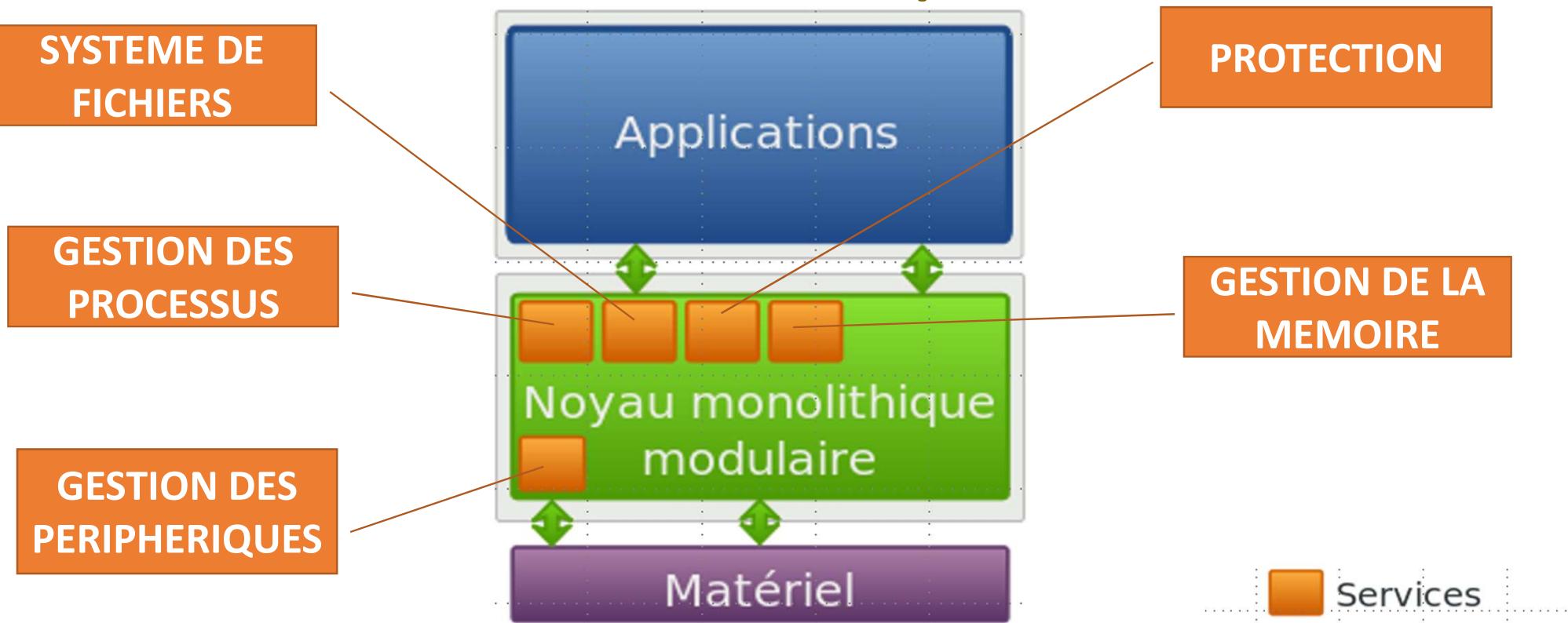
Architectures des systèmes d'exploitation

Noyau du système d'exploitation

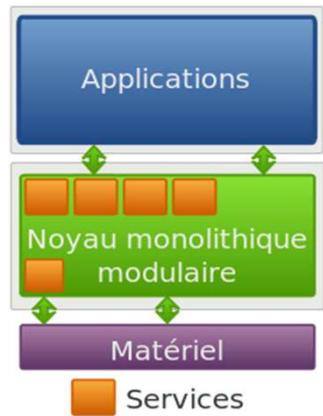
- Le noyau est la partie centrale du système d'exploitation
- Il regroupe les processus les plus critiques du système d'exploitation
 - La gestion des processus
 - La gestion de la RAM
 - La gestion du système de fichier
 - La gestion de la sécurité
 - Dans certains cas, la gestion des périphériques

Architectures des systèmes d'exploitation

Architecture monolithique



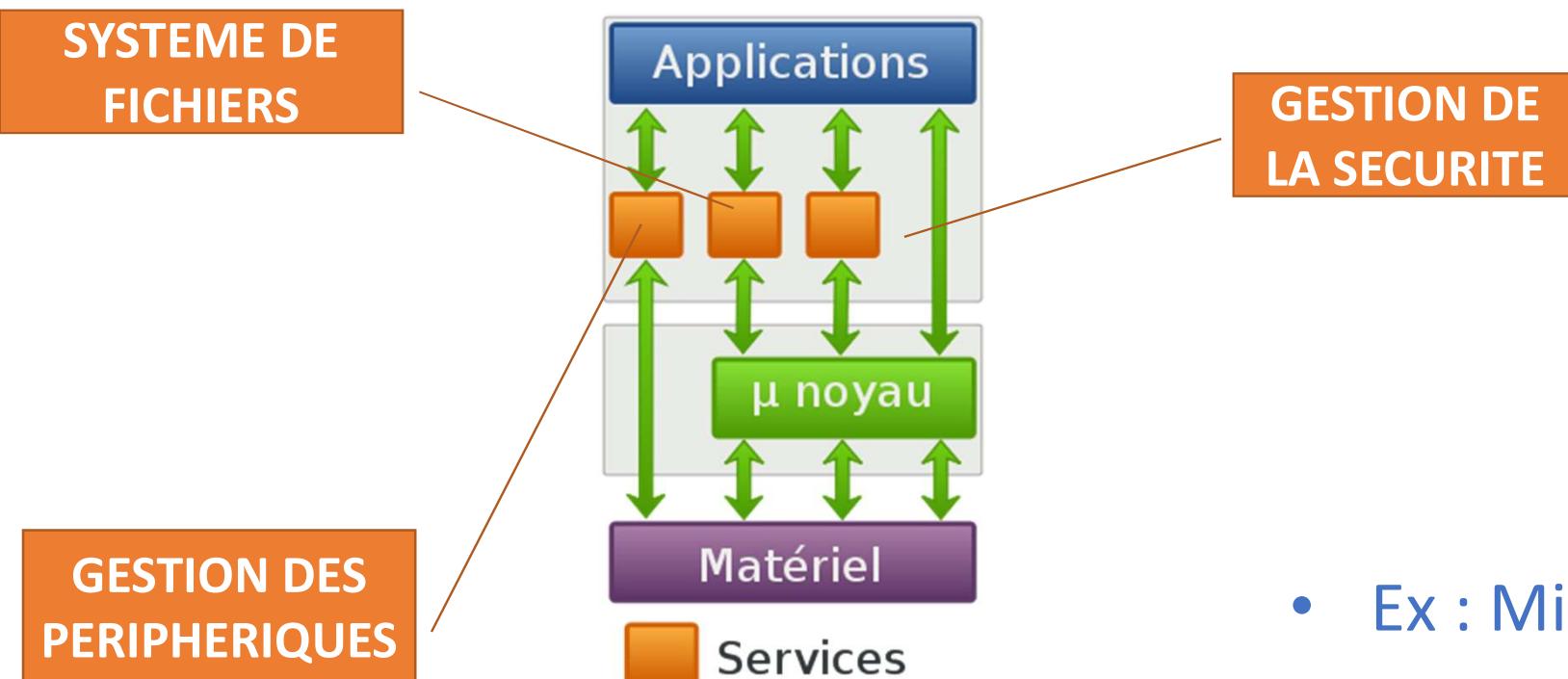
Architecture monolithique



- Le système d'exploitation est constitué d'un « gros » noyau et des programmes systèmes.
- une collection de fonctions chacune pouvant appeler l'autre à tout moment.
- Le noyau s'exécute en mode superviseur et les applications en mode utilisateur.
- Fiabilité et Tolérance aux Pannes : la défaillance dans une partie du noyau peut affecter l'ensemble du système, car tous les composants partagent le même espace d'adressage.

Architectures des systèmes d'exploitation

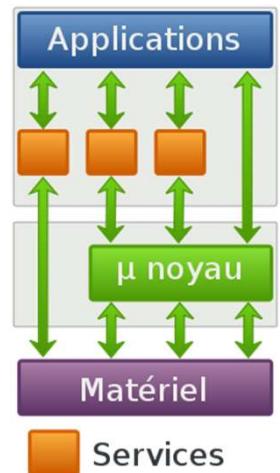
Architecture à micronoyau



- Ex : Minix, QNX

Architectures des systèmes d'exploitation

Architecture à micronoyau



- Le S.E. est constitué d'un noyau de petite taille.
- Le noyau s'exécute en mode superviseur et les programmes en mode utilisateurs.
- **Isolation des Services** : Chaque service du noyau fonctionne dans son propre espace d'adressage protégé. Cela signifie que si un module plante ou rencontre un problème, il n'affectera pas nécessairement le reste du système..
- **Performances** : L'architecture à micronoyau peut, dans certains cas, offrir des performances supérieures, car les services peuvent être optimisés indépendamment les uns des autres.



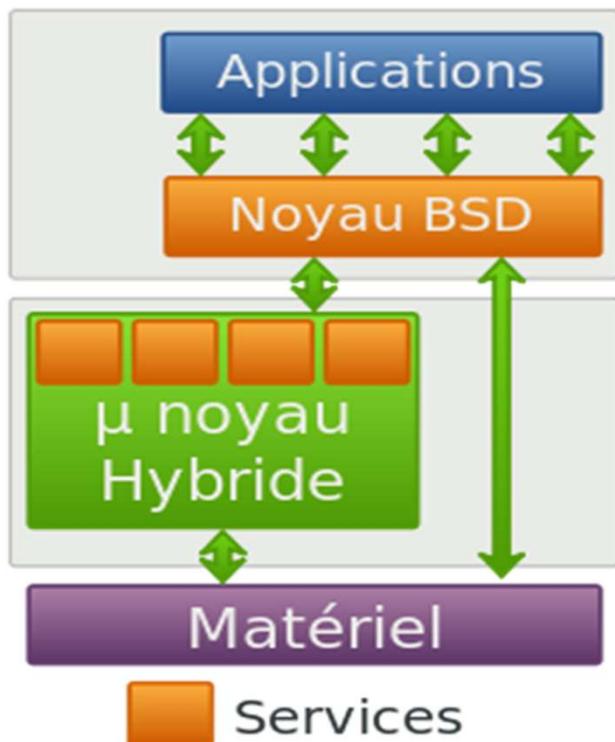
Architectures des systèmes d'exploitation

Architecture à micronoyau

- Les modules du noyau **communiquent** entre eux en utilisant des mécanismes spécifiques, tels que des appels de procédure à distance (**RPC**) ou des messages interprocessus (**IPC**).
- **Extensibilité** : Ajout de nouveaux services ou mise à jour des parties spécifiques du noyau sans affecter l'ensemble du système.
- **Fiabilité et Tolérance aux Pannes** : Comme chaque composant fonctionne de manière isolée, une défaillance dans un module n'affectera pas nécessairement le fonctionnement des autres parties du système. Cela peut conduire à un système plus robuste et tolérant aux pannes

Architectures des systèmes d'exploitation

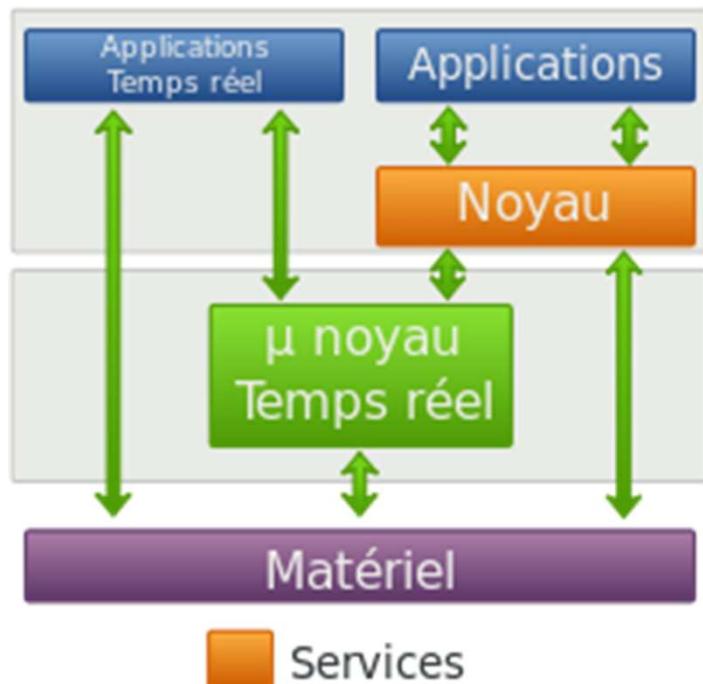
Architecture à micronoyau hybride



- Un **micronoyau hybride** combine des caractéristiques des architectures de **micronoyau** et de **noyau monolithique**. Cette approche cherche à tirer parti des avantages de la modularité offerts par les micronoyaux tout en conservant certaines caractéristiques du noyau monolithique pour optimiser les performances.

Architectures des systèmes d'exploitation

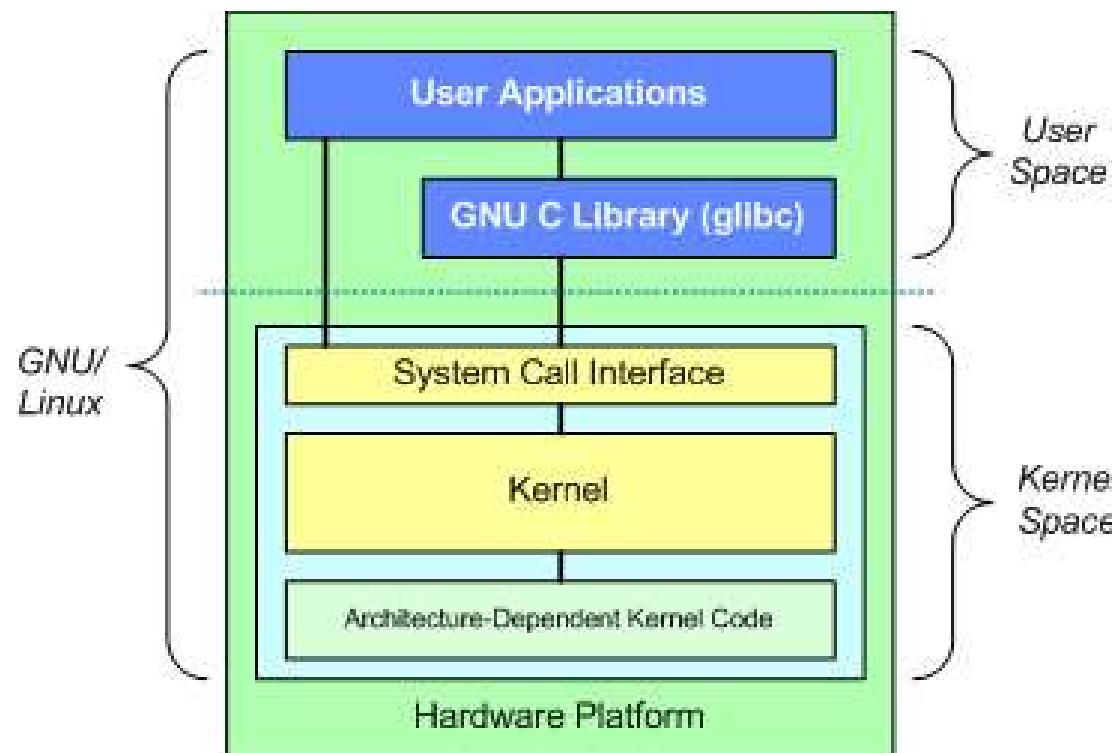
Architecture à micronoyau temps réel



- L'utilisation de micronoyaux temps réel est courante dans les systèmes embarqués, les systèmes de contrôle industriel, les systèmes de navigation automobile, les équipements médicaux, etc., où la réactivité et la prédictibilité sont essentielles, critiques.

Architectures des systèmes d'exploitation

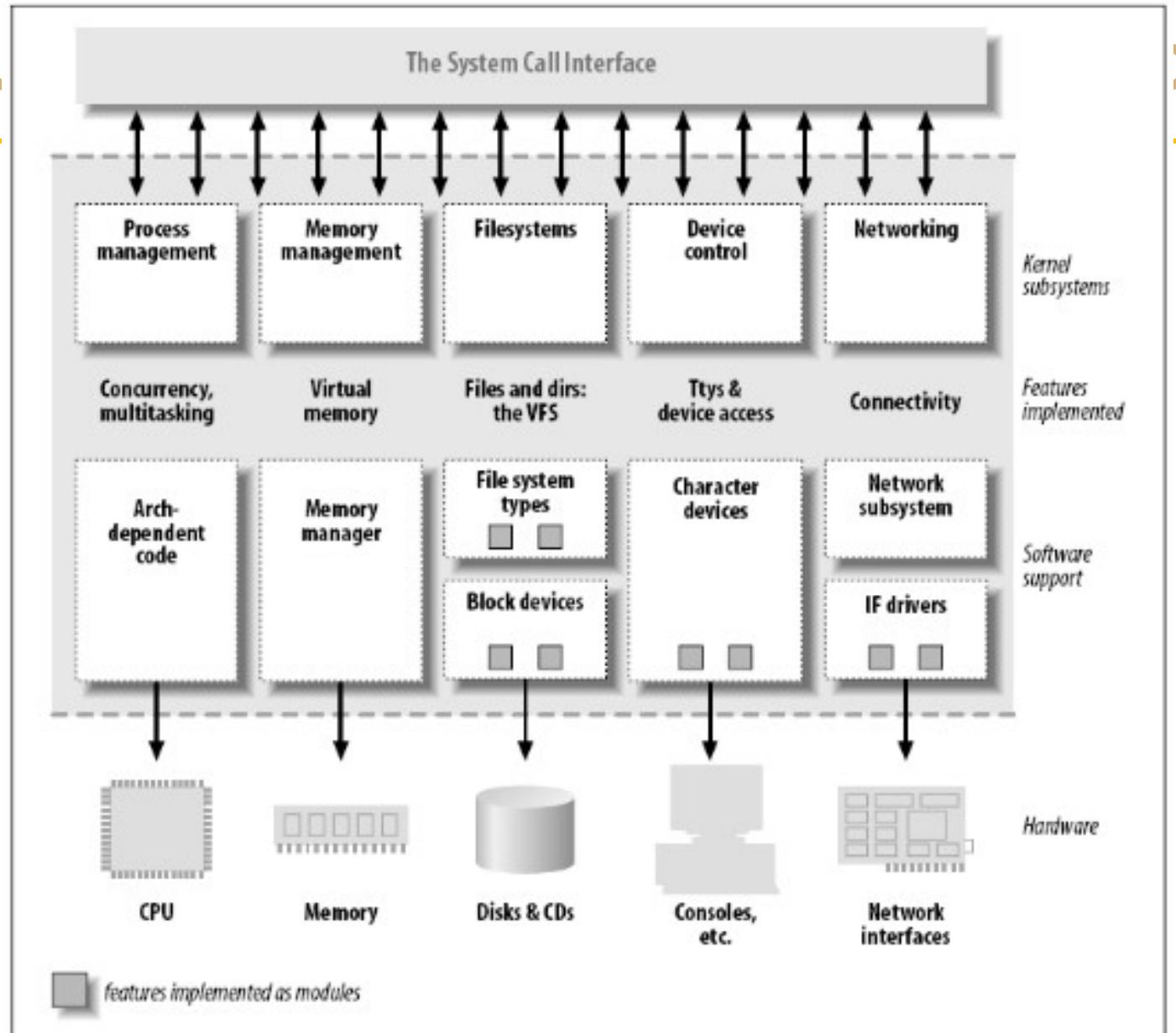
Exemple de noyau : le kernel de linux





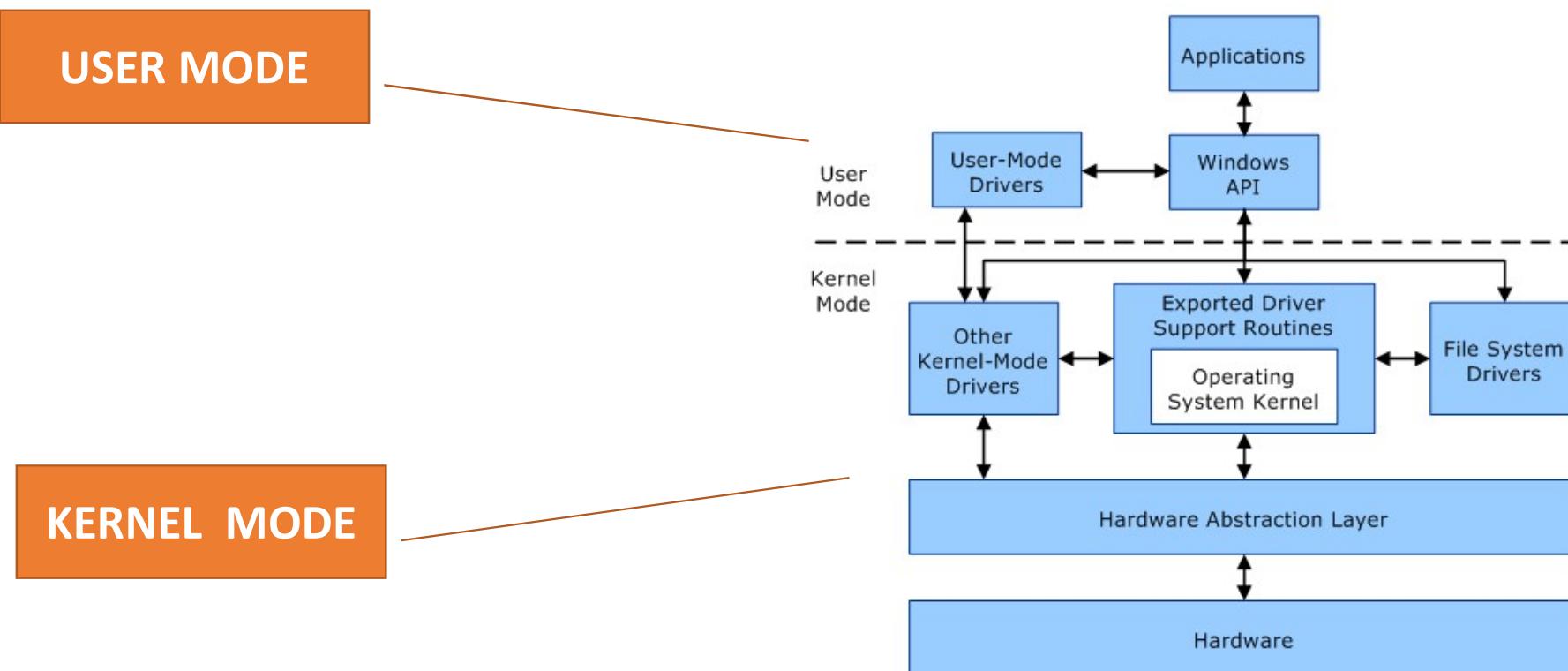
Architecture

Exemple de noyau : le kernel de linux



Architectures des systèmes d'exploitation

Exemple de noyau : le kernel de Windows 10

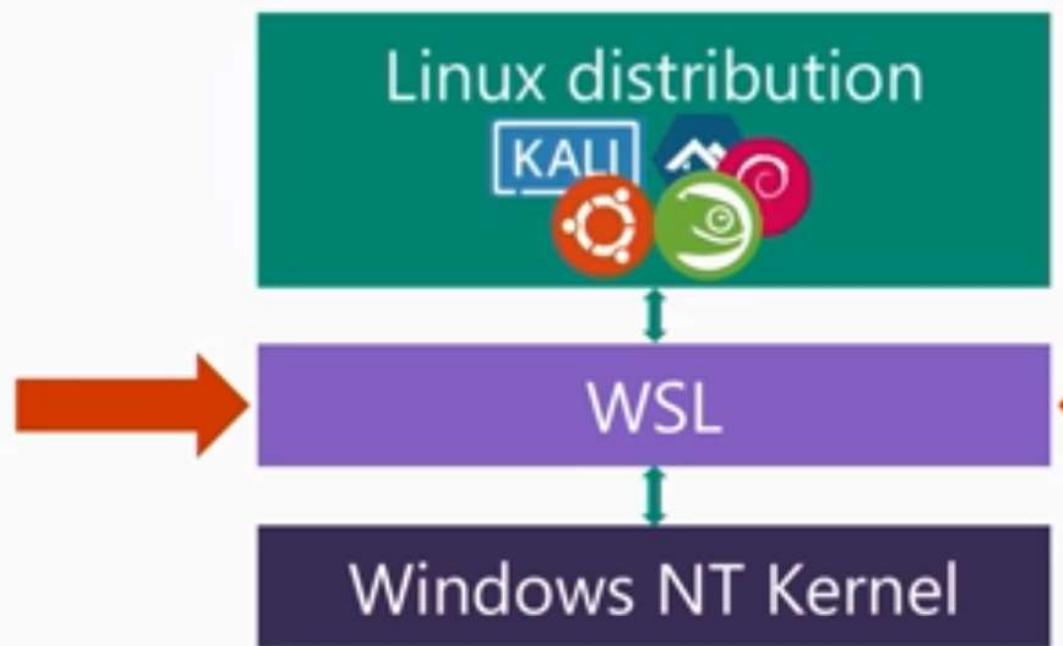




Architectures des systèmes d'exploitation

Architecture WSL : noyau Linux incorporé dans Windows

WSL architecture



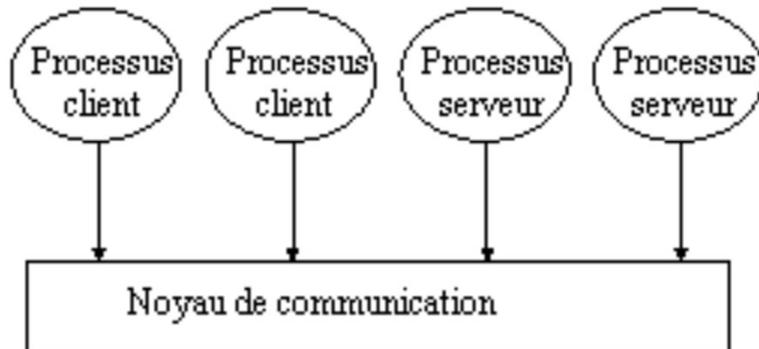
- **WSL (Windows System for Linux)**
- Compilateurs linux dans un environnement windows.
- Exécution des shells (CLI) linux bash, zsh, fish
-

<https://docs.microsoft.com/en-us/windows/wsl/install-win10>

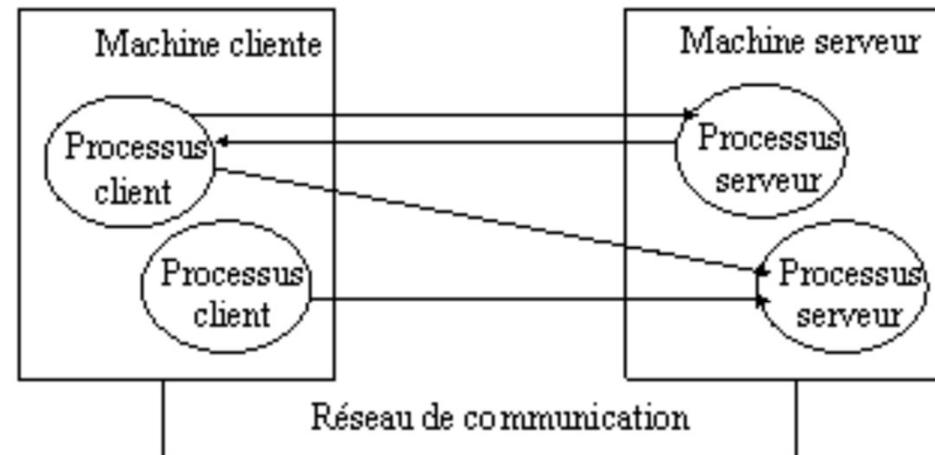
<https://ubuntu.com/wsl>

Architectures des systèmes d'exploitation

Architecture Client-serveur



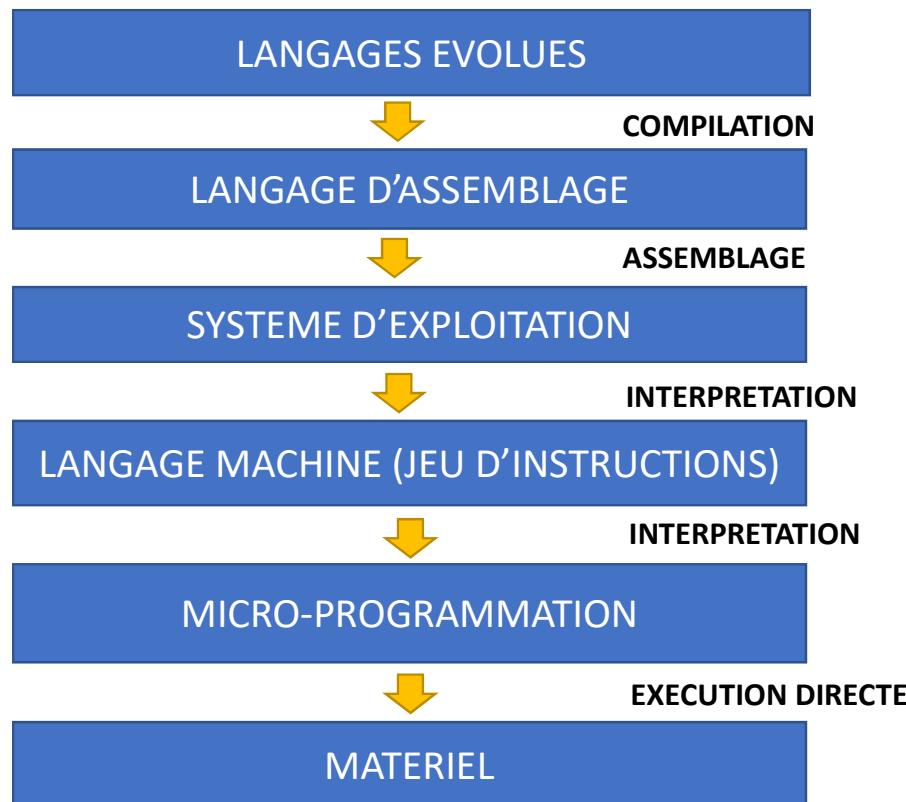
a. Sur une même machine



b. Entre deux machines distinctes

Architectures des systèmes d'exploitation

Architecture en couches des systèmes d'exploitation

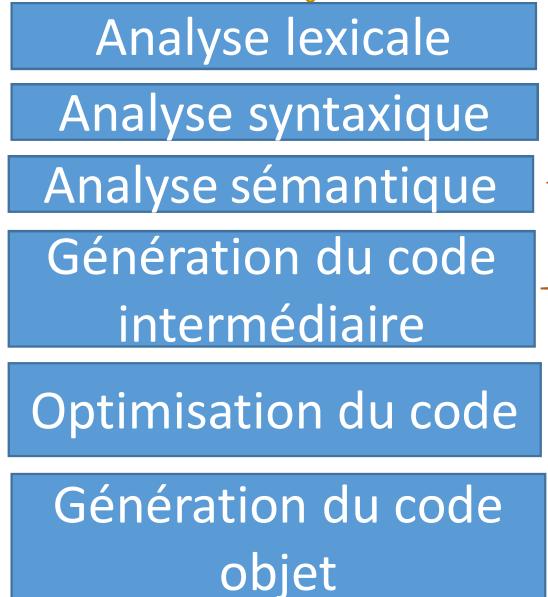


- Modèle dans lequel chaque couche fournit des services à la couche suivante
- Langage machine = suite de “0” et de “1”
- Langages évolués proches du langages humain
- Lorsque vous prorammez en Langage C, C++, VB.., vous devez **compiler** votre code source pour obtenir un code objet (assembleur)

Architectures des systèmes d'exploitation

Compilation

Programme en code source



Lecture / identification des éléments constitutifs du programme (variables, noms de procédures...)

Vérifie si la liste fournie précédemment est correcte et bien agencée et génère l'arbre syntaxique

Identifie les opérateurs et les operands et vérifie s'ils sont autorisés par le compilateur.

Production d'un code objet à partir de l'arbre syntaxique

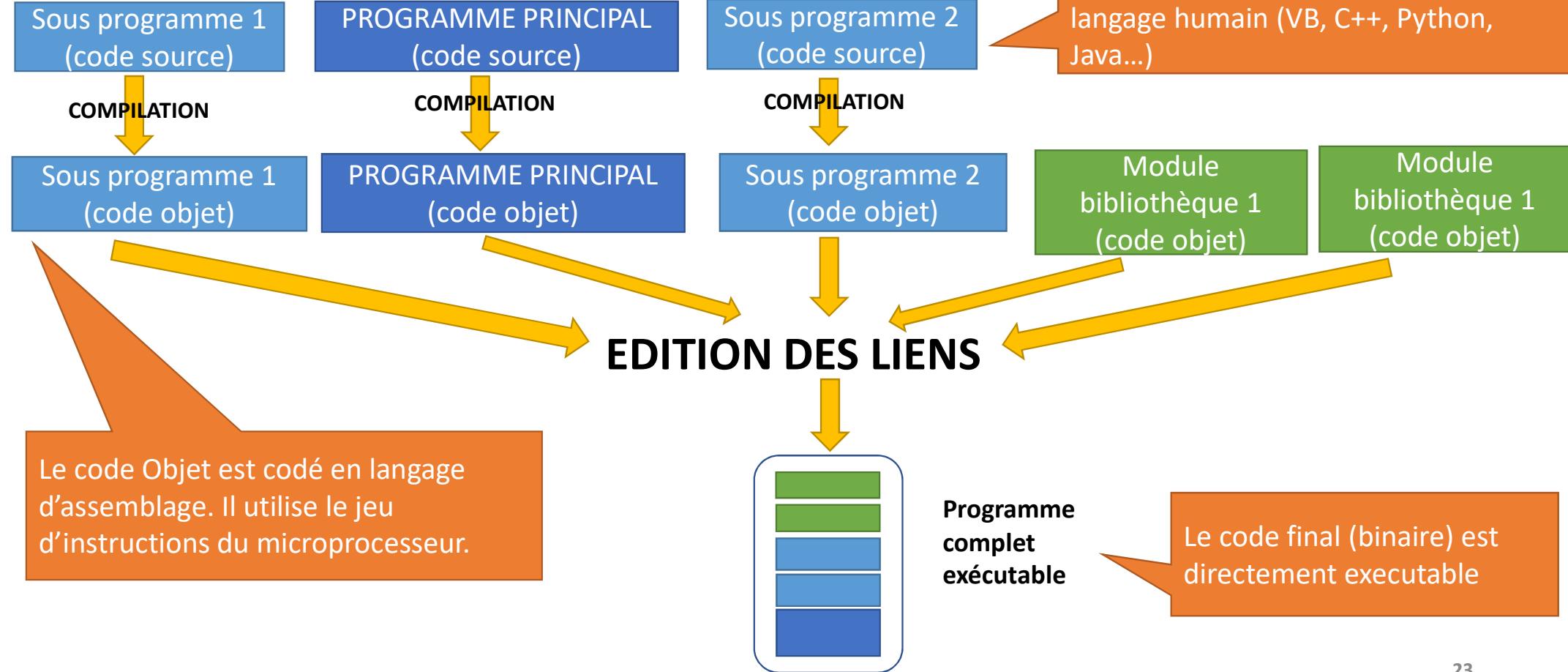
Optimisation par rapport à l'ordinateur cible (taille du code, organisation en mémoire...)

Le code final du programme est générée.

Programme en code objet

Architectures des systèmes d'exploitation

Compilation et édition des liens



A BIENTOT