

Systemes d'exploitation

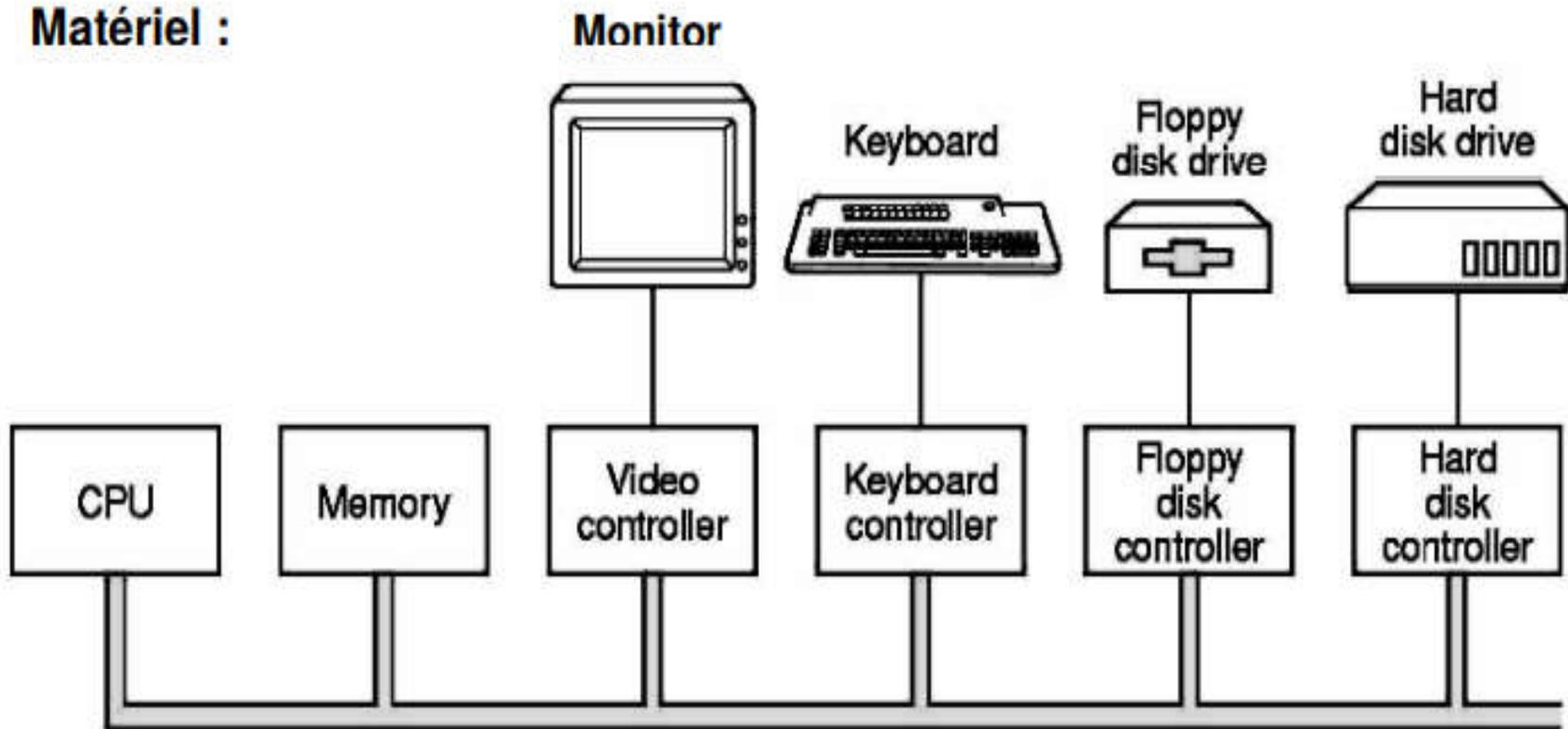


Introduction au systèmes d'exploitation

Introduction

- Ordinateur: matériel + logiciel.

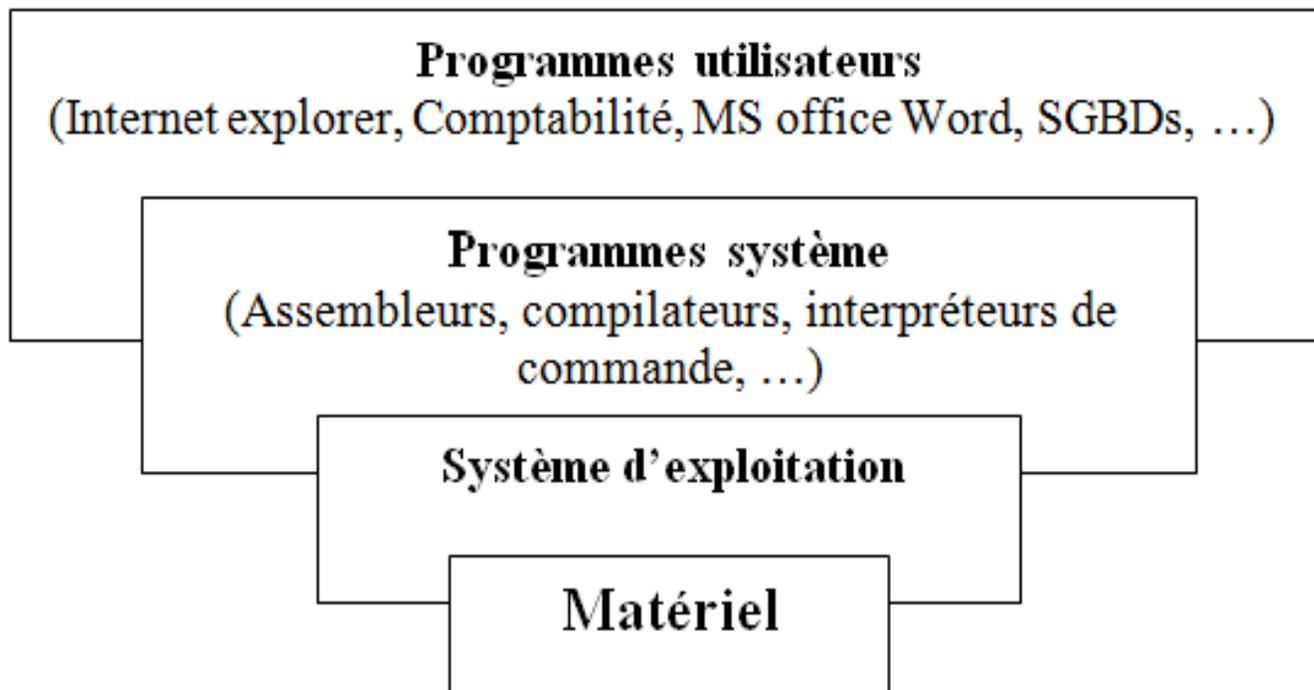
Matériel :



Bus

Introduction

- **Logiciel:**



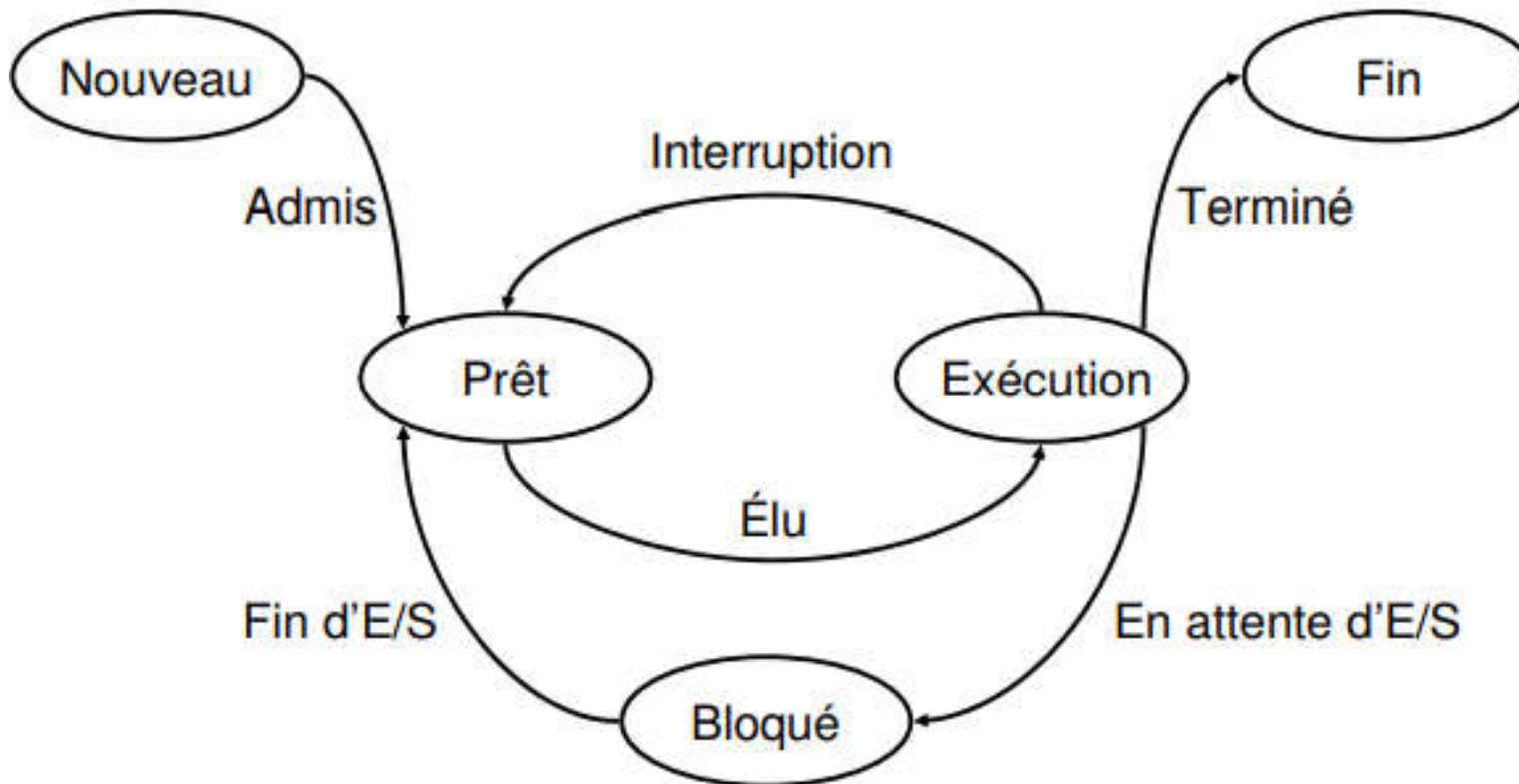
Les couches d'un système informatique

Introduction

- Système d'exploitation: ensemble de programmes qui agissent comme un intermédiaire entre l'utilisateur et l'ordinateur.
- l'utilisateur à une vue simplifiée de l'ordinateur lui permettant de l'utiliser plus facilement.
- **But :**
Développer des applications sans se soucier des détails de fonctionnement et de gestion du matériel.

Introduction

- Processus : Programme en cours d'exécution



Fonctions d'un SE

- Gestion du processeur et des processus
- Gestion de la mémoire
- Gestion des périphériques
- Gestion des fichiers
- Protection et sécurité

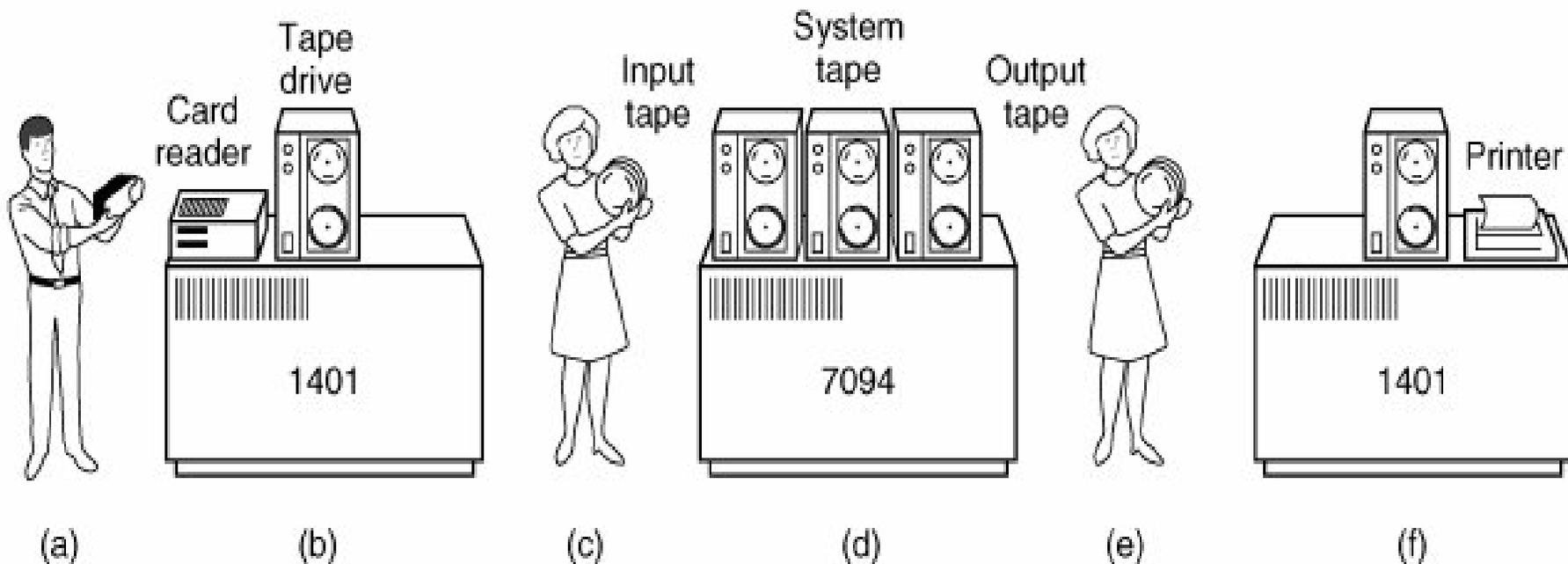
Évolution des systèmes d'exploitation

- **Machines portes ouvertes (nues) : 1945-1955**
- machines énormes, couteuses, très peu fiables et peu rapides.
- Pas de système d'exploitation: le programmeur s'occupe de la programmation, du chargement et du lancement des programmes.
- Programmation en langage machine.
- Mode d'exploitation : Monoprogrammation
- Début des année 50 : Introduction des cartes perforées

Évolution des systèmes d'exploitation

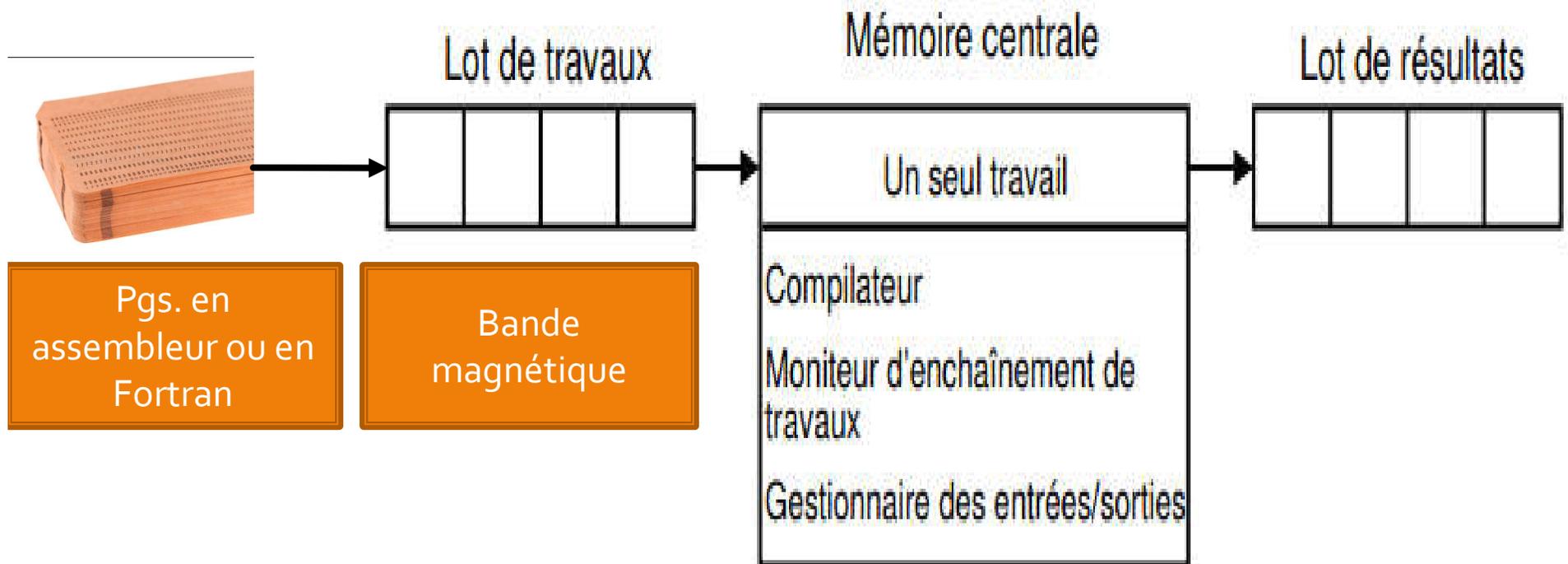
- **Traitement par lots : 1955-1965**

Lot: ensembles de programmes écrits dans un même langage et font appel au même compilateur.



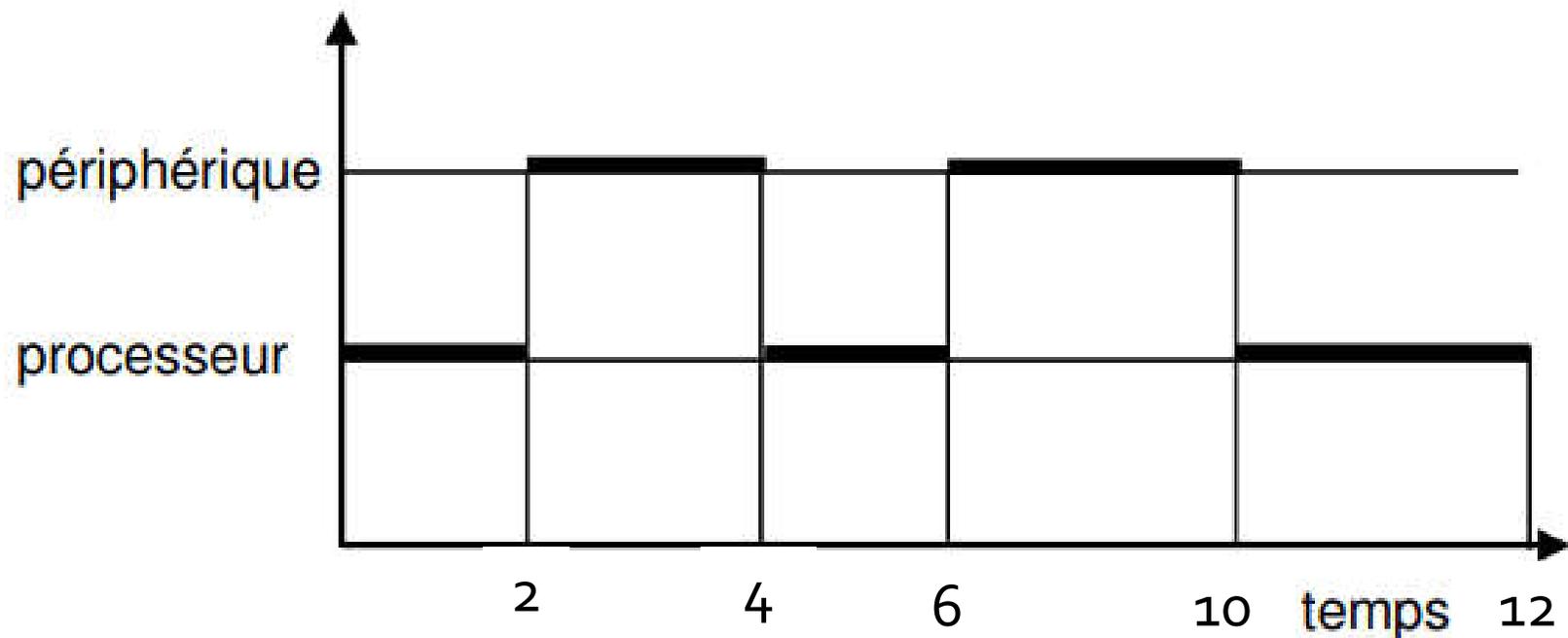
Évolution des systèmes d'exploitation

- Traitement par lots : 1955-1965



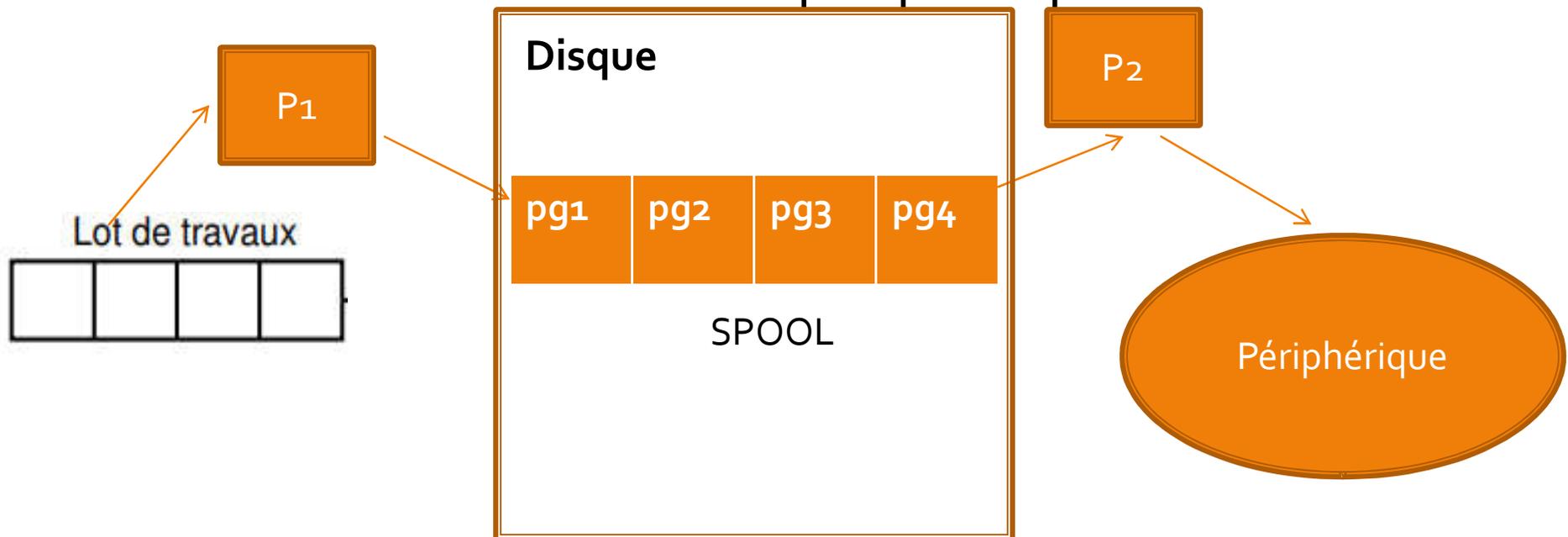
Évolution des systèmes d'exploitation

- **Traitement par lots : 1955-1965**
- Mode d'exploitation: Monoprogrammation (un seul programme en mémoire à la fois)
- Inconvénient: Processeur inutilisé pendant les E/S



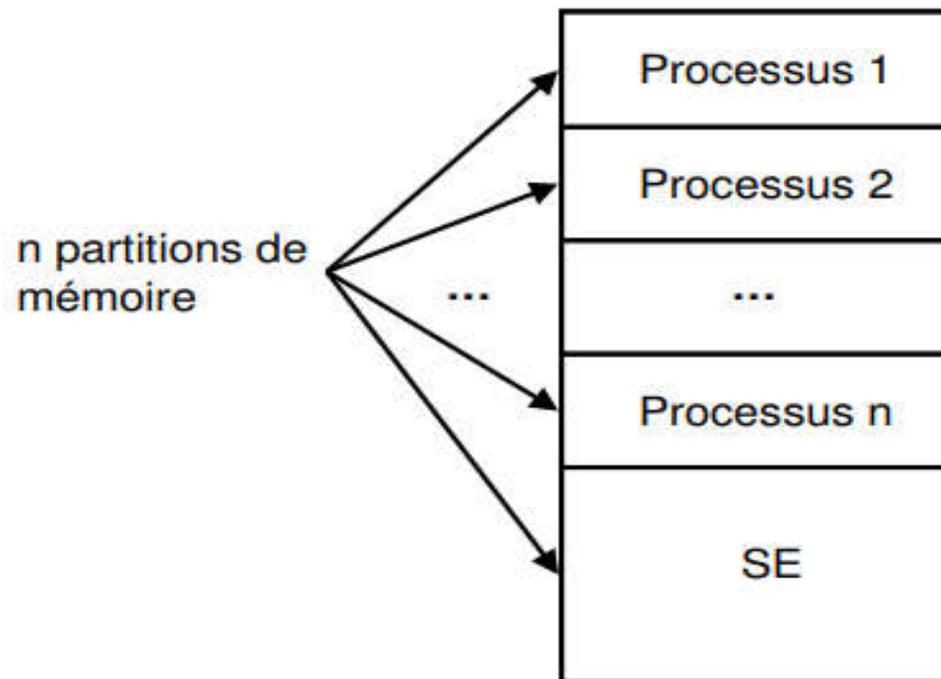
Évolution des systèmes d'exploitation

- Multiprogrammation et traitement par lots : 1965-1980
- SPOOLING (Simultaneous Peripheral Operation On Line)
 - Simultanéité de transferts de programmes dans le disque et transfert de fichiers vers les périphériques.



Évolution des systèmes d'exploitation

- Spooling:
 - Multiprogrammation: Plusieurs P.gs en mémoire.
 - Simultanéité calcul et E/S.



Évolution des systèmes d'exploitation

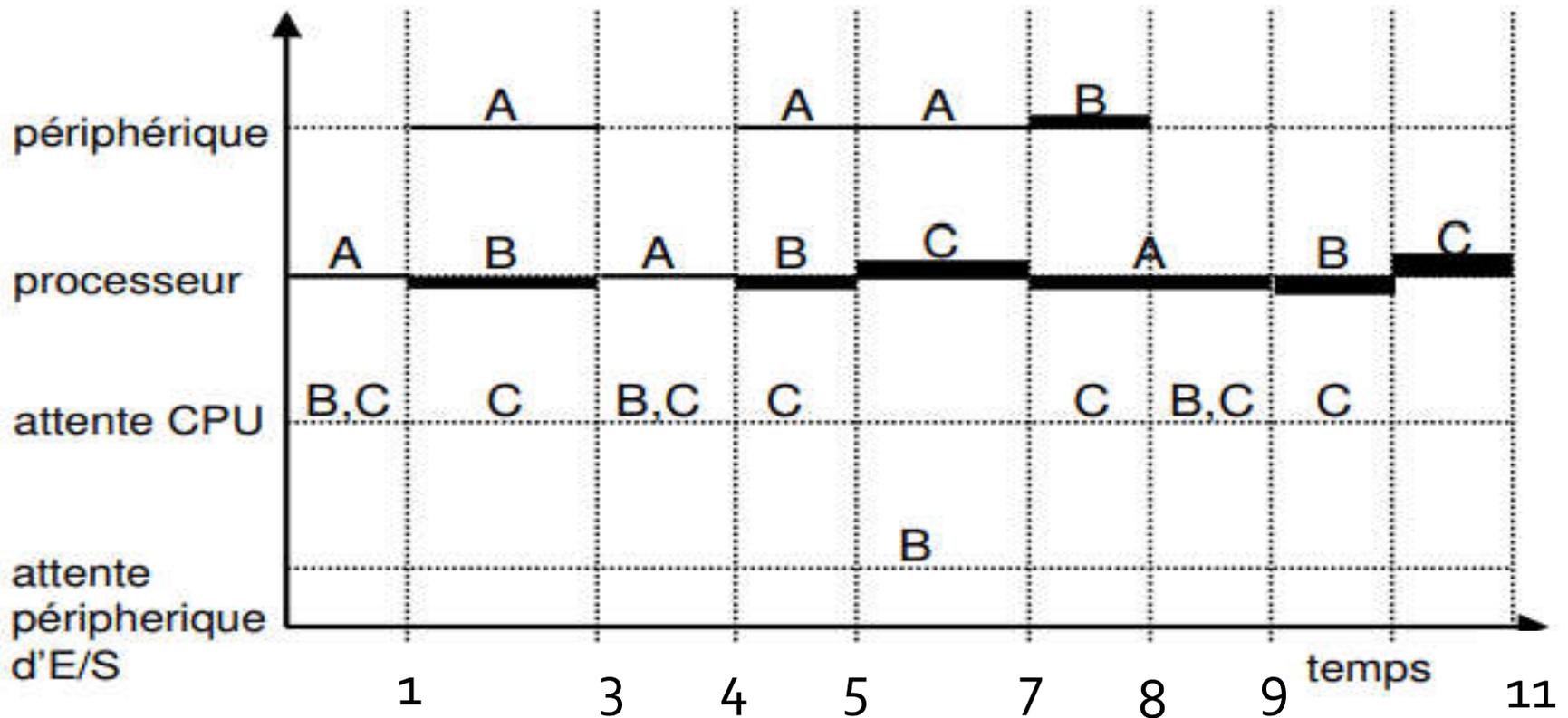
- **Multiprogrammation sans interruption de programme :**
 - Le processeur est accordé au programme prioritaire qui le garde jusqu'à ce qu'il se termine ou fasse une E/S.
 - Pendant qu'un processus est en attente ou s'il s'est terminé, le contrôle est repris par le SE qui accorde le processeur au programme le plus prioritaire qui demande à faire du calcul.

Évolution des systèmes d'exploitation

- **Multiprogrammation avec interruption de programme:**
- Lorsqu'une opération d'E/S est terminée le SE reprend la main et donne l'UC au programme le plus prioritaire qui la demande.
- A tout moment le déroulement d'un programme peut être interrompu au profit d'un autre plus prioritaire: Une interruption est alors produite.

Évolution des systèmes d'exploitation

- Exemple : Priorité croissante (C,B,A)

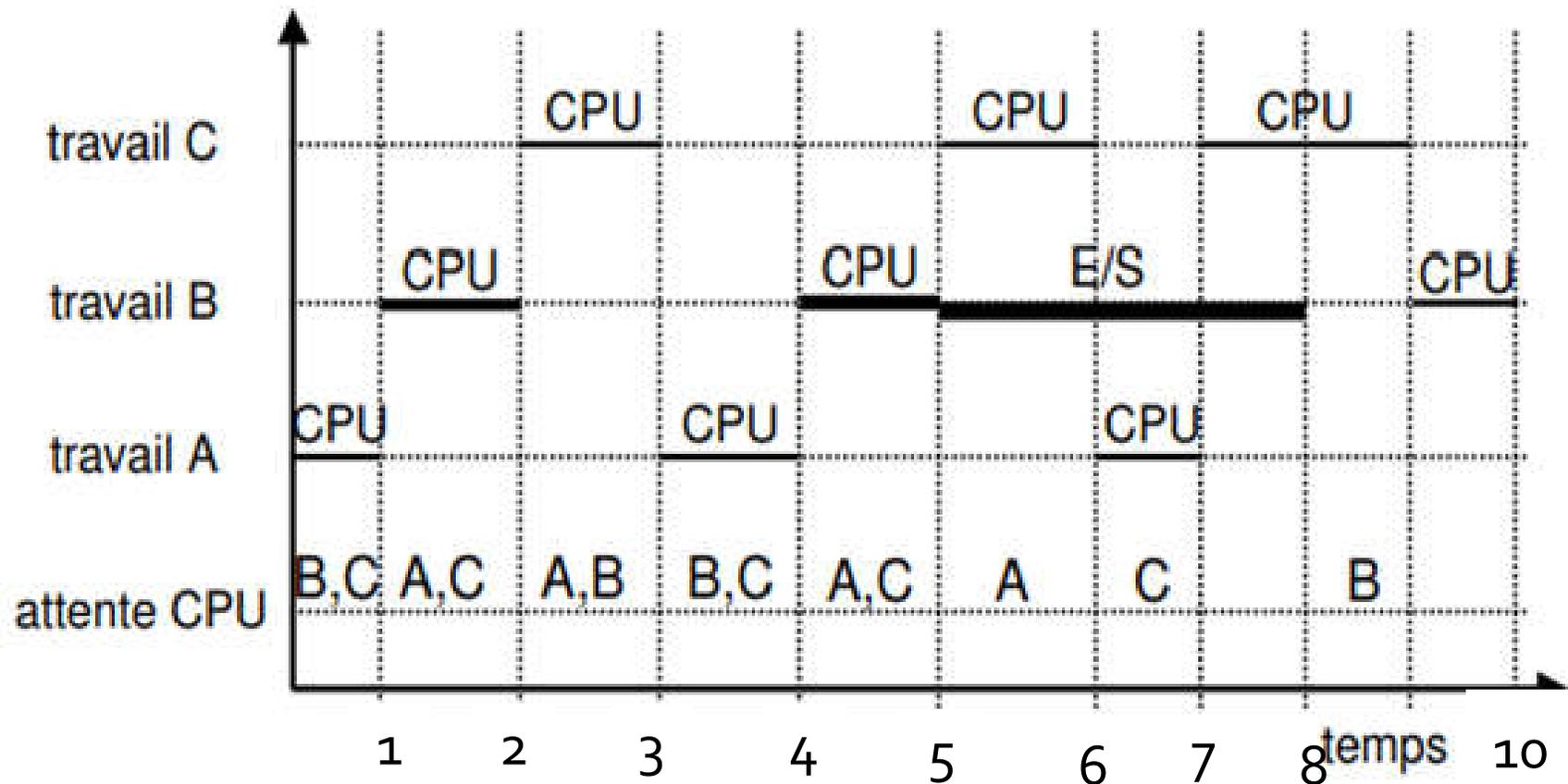


Évolution des systèmes d'exploitation

- **Temps partagé:** plusieurs utilisateurs peuvent se connecter à la machine par l'intermédiaire de leurs terminaux et travailler en même temps.
- Le processeur est alloué, à tour de rôle, pendant un certain temps à chacun des travaux en attente d'exécution.
- ce mode d'exploitation donne l'impression que les programmes s'exécutent en parallèle.

Évolution des systèmes d'exploitation

- Exemple: temps partagé ordre ABC



Évolution des systèmes d'exploitation

- Le temps réel: Bien que le temps partagé ait permis un temps de réponse approprié aux utilisateurs, il existe des domaines (industriel, médical, militaire) dans lesquels, l'ordinateur doit réagir immédiatement à un changement des données d'entrée.
- Quand un dispositif d'entrée veut envoyer une donnée, il active sa ligne d'interruption qui sera détectée par le processeur. Le processeur interrompt son travail et exécute un programme particulier associé à cette interruption

Évolution des systèmes d'exploitation

- (1980-1990) L'informatique personnelle – Les systèmes distribués
- Informatique personnelle:
Micro-ordinateurs (PC) ont connu un développement. Le premier PC, conçu par IBM a été équipé du système MS-DOS de Microsoft. Ce système est l'ancêtre de l'actuel Windows. D'autres constructeurs ont développé d'autres architectures et l'ont équipé de variétés de SEs comme Motorola, Apple ou Digital Research.

Évolution des systèmes d'exploitation

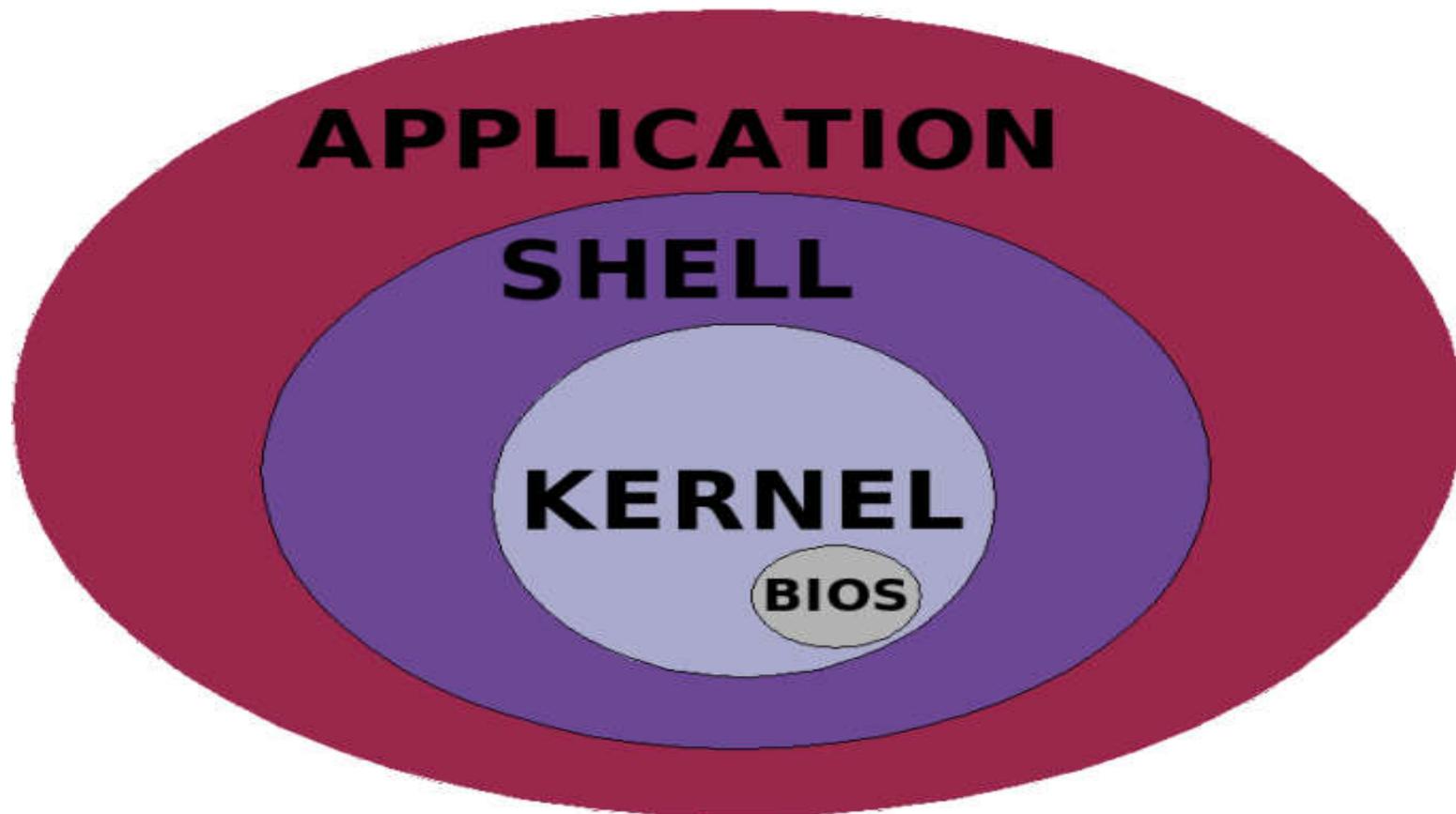
- **Système d'exploitation réseau**
- Un système distribué est un ensemble de machines autonomes connectées par un réseau, et équipées d'un logiciel dédié à la coordination des activités du système ainsi qu'au partage de ses ressources
- Un système distribué est un système qui s'exécute sur un ensemble de machines sans mémoire partagée, mais que pourtant l'utilisateur voit comme une seule et unique machine

Évolution des systèmes d'exploitation

- **La miniaturisation – Les systèmes embarqués:1990 à nos jours:** Téléphones portables, Téléviseurs, machines à laver, équipement médical, etc .
-Équipements électroniques autonomes conçus pour une tâche bien précise doté d'un SE et de processeurs.

Exemple: Un système à domicile peut contrôler la température, l'éclairage, l'alarme et même la machine à café. Bientôt, le réfrigérateur pourra commander du lait par internet quand il s'aperçoit qu'il n'en reste plus assez.

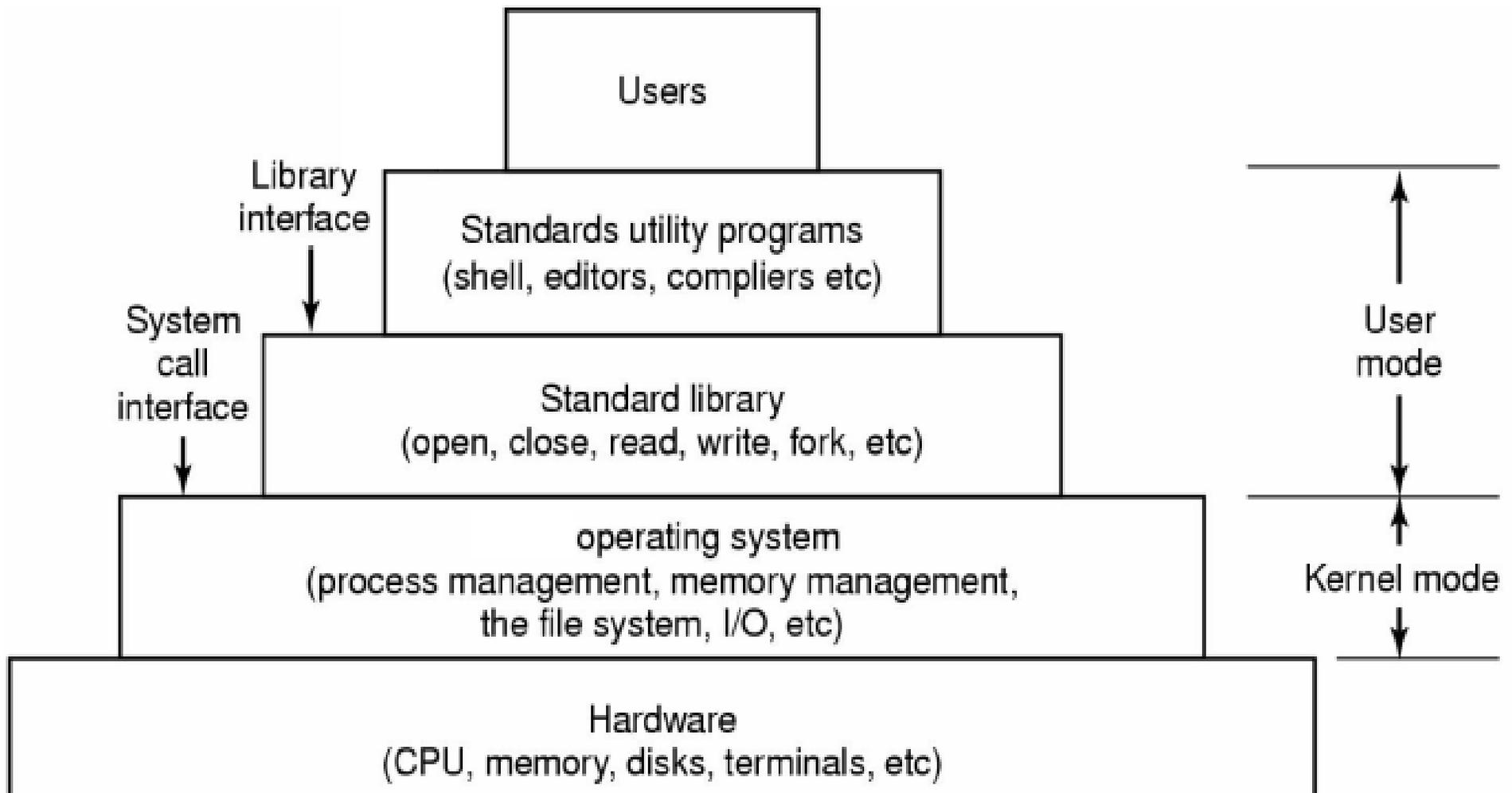
Organisation d'un système d'exploitation



Interface avec le matériel

- Chaque composant (processeurs, mémoires et périphériques) de l'ordinateur a son propre code (câblé ou logiciel) qui assure son fonctionnement et les interactions avec les autres.
- Le système d'exploitation gère et coordonne l'ensemble de ces composants au moyen de signaux (interruptions).
- Les interruptions permettent au SE de reprendre le contrôle:
 - Interruptions matérielles :
 - Horloges (pour gérer l'allocation des processeurs)
 - Périphériques
 - Interruptions logicielles :
 - Erreurs arithmétiques (division par zéro)
 - Données non disponibles en mémoire (défaut de page)
 - Appels système (invocation du système d'exploitation).

Interactions utilisateur/systeme

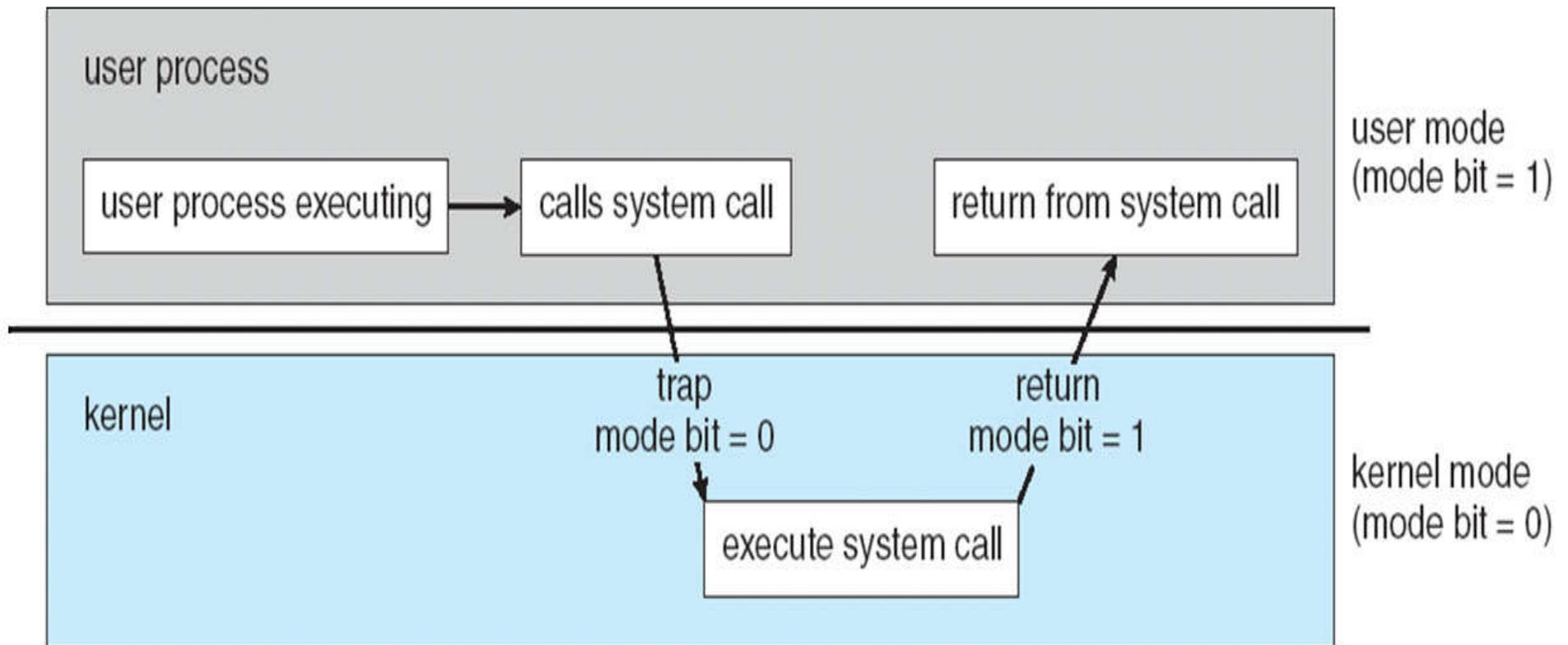


Appels système

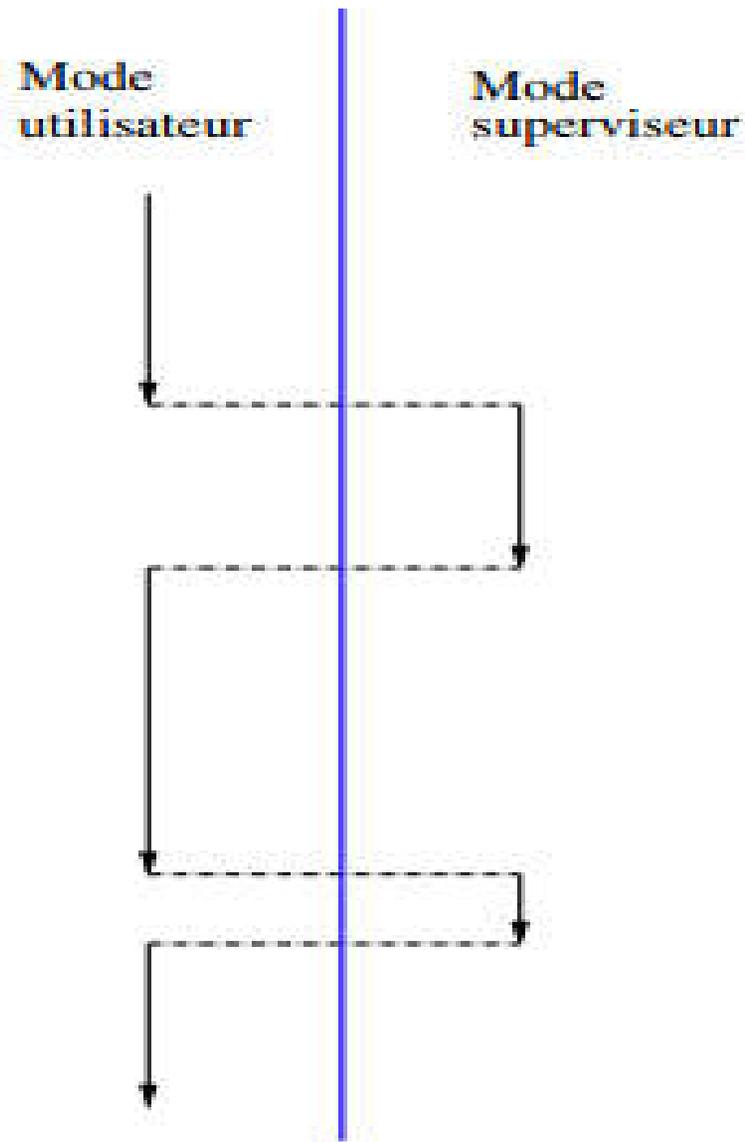
- les processeurs ont deux modes de fonctionnement:
 - Le **mode superviseur** (noyau, privilégié ou maître): pour le système d'exploitation, où toutes les instructions sont autorisées.
 - Le **mode utilisateur** (esclave): pour les programmes des utilisateurs et les utilitaires, où certaines instructions ne sont pas permises.
- Un **appel système** consiste en une interruption logicielle qui a pour rôle d'activer le système d'exploitation. Il a pour but: changer de mode d'exécution pour passer du mode utilisateur au mode maître.

Appels système

- Les processeurs sont dotés d'un **bit de mode**: Ce mode de fonctionnement assurent la protection du système d'exploitation contre les intrusions et les erreurs.



Appel système	Description
fork	Créer un processus
waitpid	Attendre la terminaison d'un processus
wait	
execve	Exécuter un autre programme
exit	Terminer l'exécution
open	Créer ou ouvrir un fichier
close	Fermer un fichier
read	Lecture de données
write	Écriture de données
lseek	Pointeur dans un fichier
stat	Obtenir l'état des attributs
mkdir	Créer un répertoire
rmdir	Éliminer un répertoire
link	Liens vers un fichier
unlink	Éliminer un fichier
mount	Monter un système de fichiers
umount	Démonter un système de fichiers
chdir	Changer de répertoire
chmod	Changer les permissions d'accès
kill	Signaux
time	Obtenir le temps



Approche basée sur des appels système

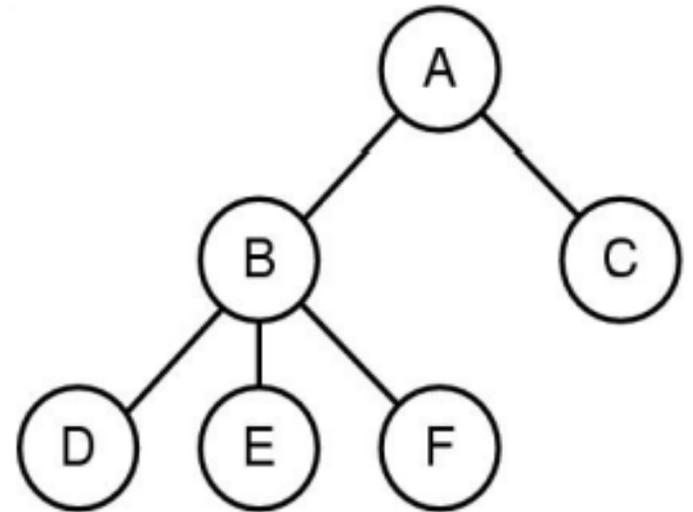
Alternance de modes d'exécution d'un processus

Appels système

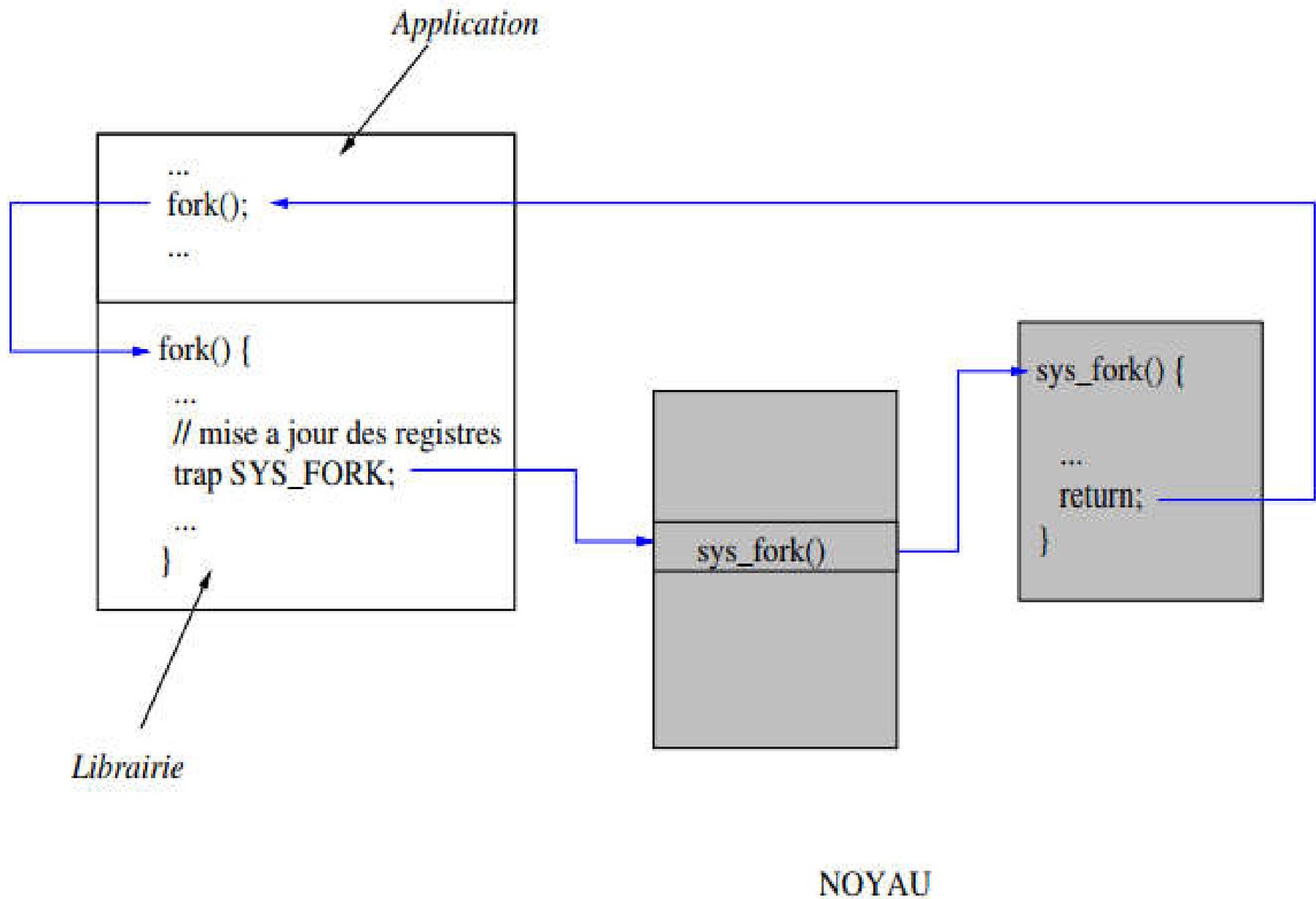
- Les appels systèmes permettent:
 - la création,
 - la synchronisation
 - l'arrêt des processus.
 - la communication interprocessus
 - gestion de fichiers et répertoires

Appels système

- Un processus peut créer un ou plusieurs processus fils qui, à leur tour, peuvent créer des processus fils (structure arborescente): commande `Fork()`



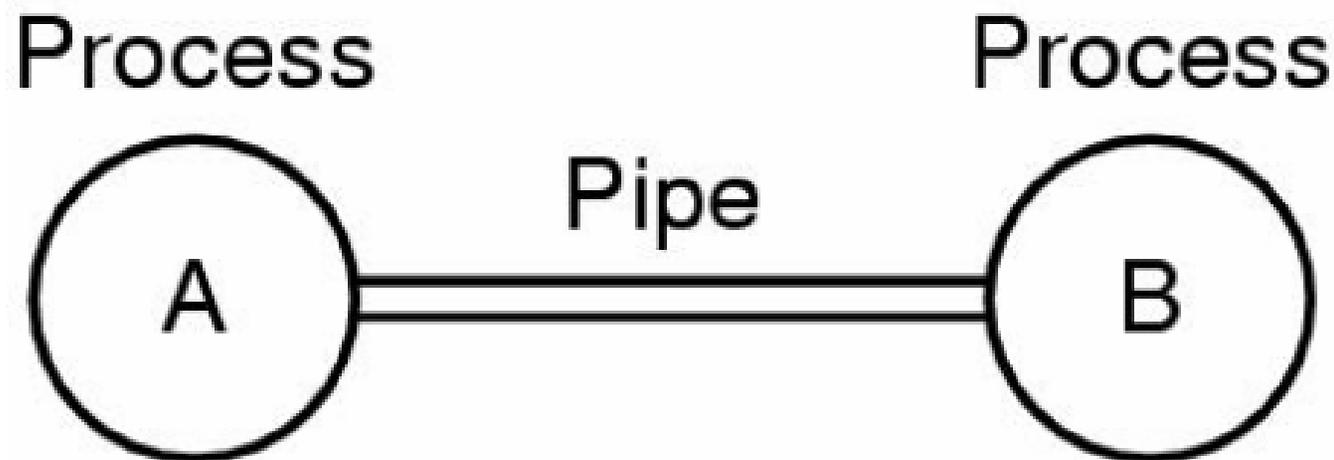
- Un processus peut être partitionné en plusieurs threads concurrents partageant un même environnement d'exécution. Les threads sont un moyen de raffiner et de diviser le travail normalement associé à un processus.



Étapes d'un appel système

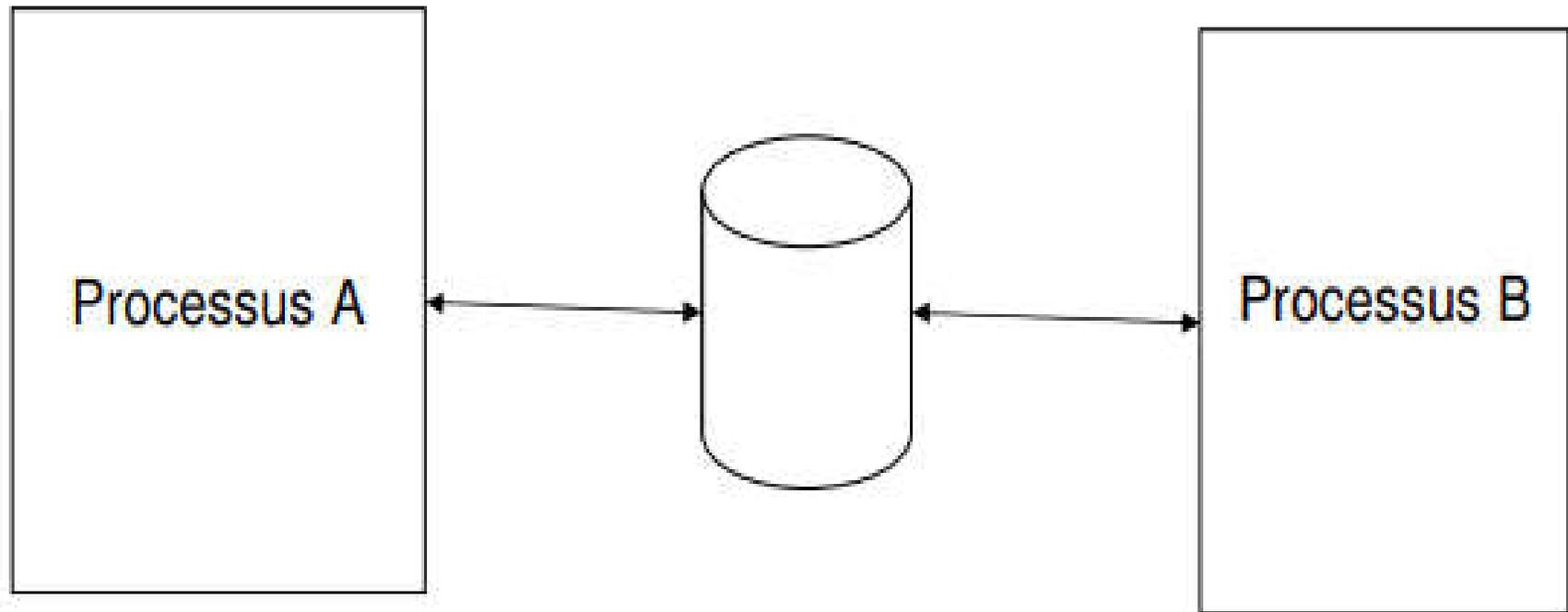
Appels système: Communication interprocessus

- - Segments de données partagés;
- Fichiers;
- Signaux;
- Messages -> Tubes de communication (pipe)



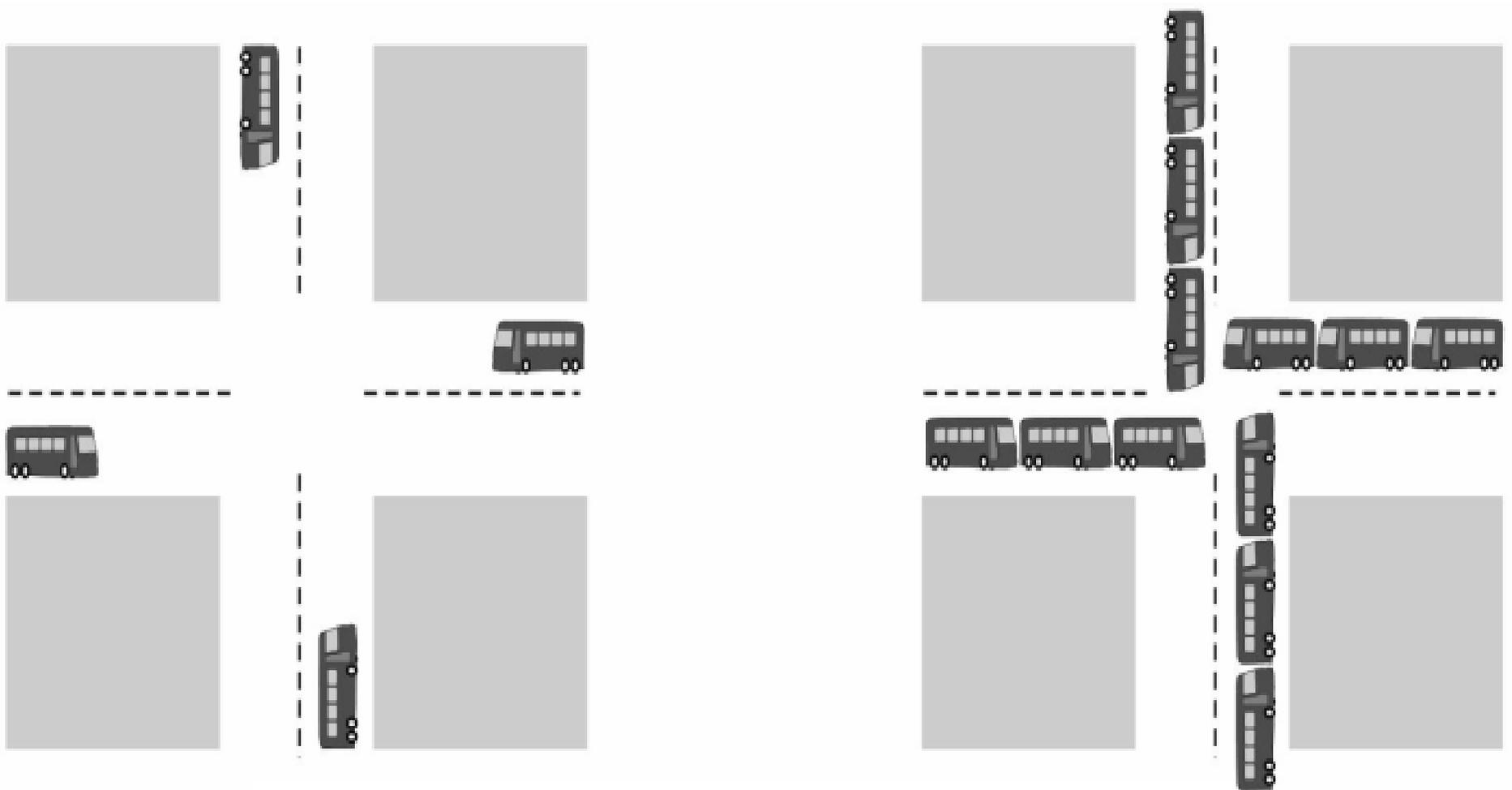
Appels système: Synchronisation de processus

- Éviter l'accès simultané lecture/écriture ou écriture/écriture à une même donnée.



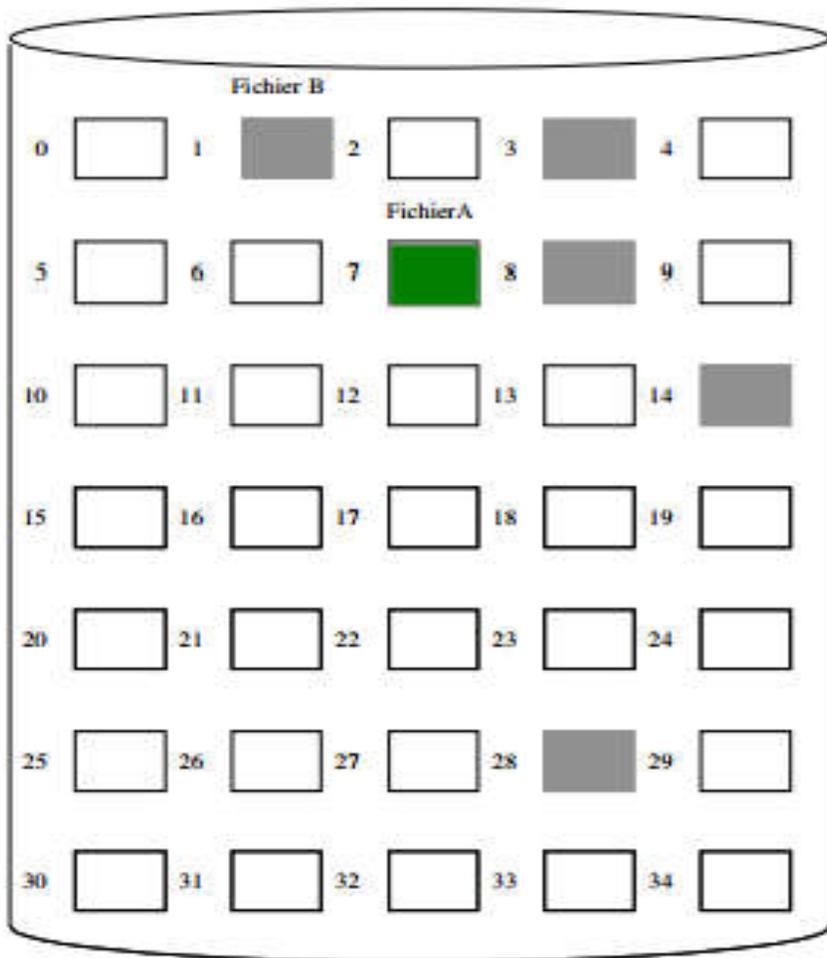
Appels système: Synchronisation de processus

- Partage de ressources:



Appels système: gestion de fichiers

- Un fichier est un ensemble de blocs sur le disque



Nom du fichier	Blocs
Fichier A	7
...	...
Fichier B	1, 8, 3, 14, 28
...	...

Appels système: gestion de fichiers

- Les appels systèmes permettent de créer des fichiers, de les supprimer, de les ouvrir, de les lire, de les modifier et de récupérer leurs attributs...

File management

Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information

Appels système: Répertoires

- Les fichiers sont regroupés dans des répertoires. Un répertoire peut contenir soit des fichiers, soit d'autres répertoires (structure arborescente).
 - L'accès au fichier se fait en spécifiant le chemin d'accès (la liste des répertoires à traverser).
 - Un chemin d'accès est absolu si le point de départ est le répertoire racine.
 - Un chemin d'accès est relatif si le point de départ est le répertoire courant.

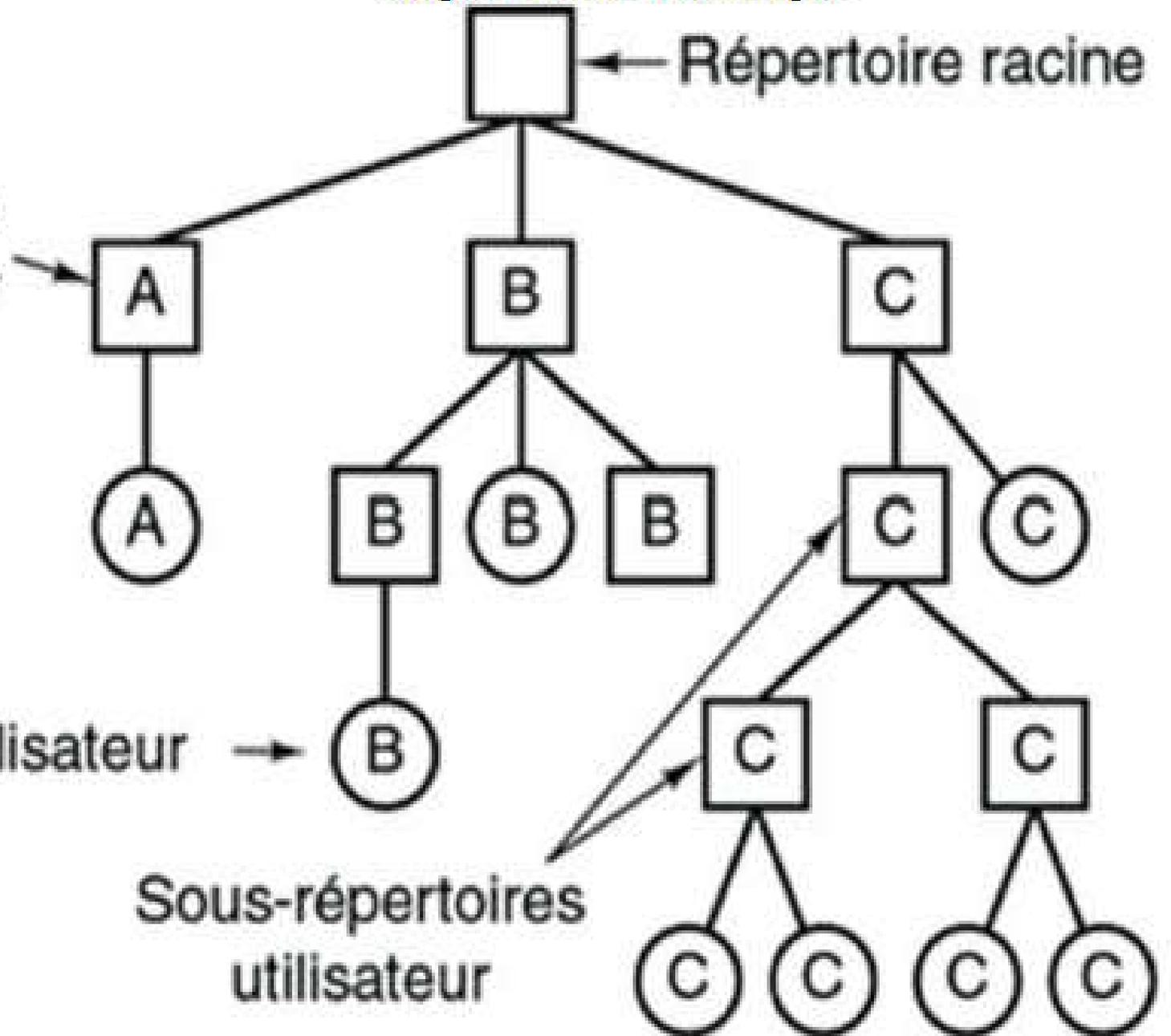
Répertoire hiérarchique

← Répertoire racine

Répertoire utilisateur →

Fichier utilisateur →

Sous-répertoires utilisateur



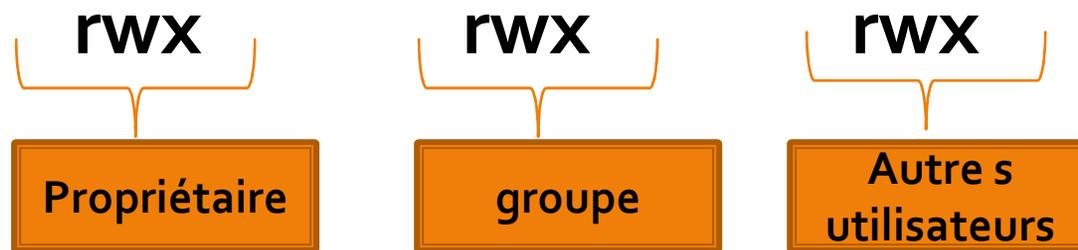
Appels système: Répertoires

Directory and file system management

Call	Description
<code>s = mkdir(name, mode)</code>	Create a new directory
<code>s = rmdir(name)</code>	Remove an empty directory
<code>s = link(name1, name2)</code>	Create a new entry, name2, pointing to name1
<code>s = unlink(name)</code>	Remove a directory entry
<code>s = mount(special, name, flag)</code>	Mount a file system
<code>s = umount(special)</code>	Unmount a file system

Droits d'accès: Répertoires et fichiers

- A chaque fichier ou répertoire sont associés des droits d'accès: autorisation ou inhibition de la lecture noté **r**, de l'écriture noté **w**, de l'exécution noté **x**.



- exemple:

droits = $110101000_2 = 650_8$

affiché sous la forme : rw-r-x---

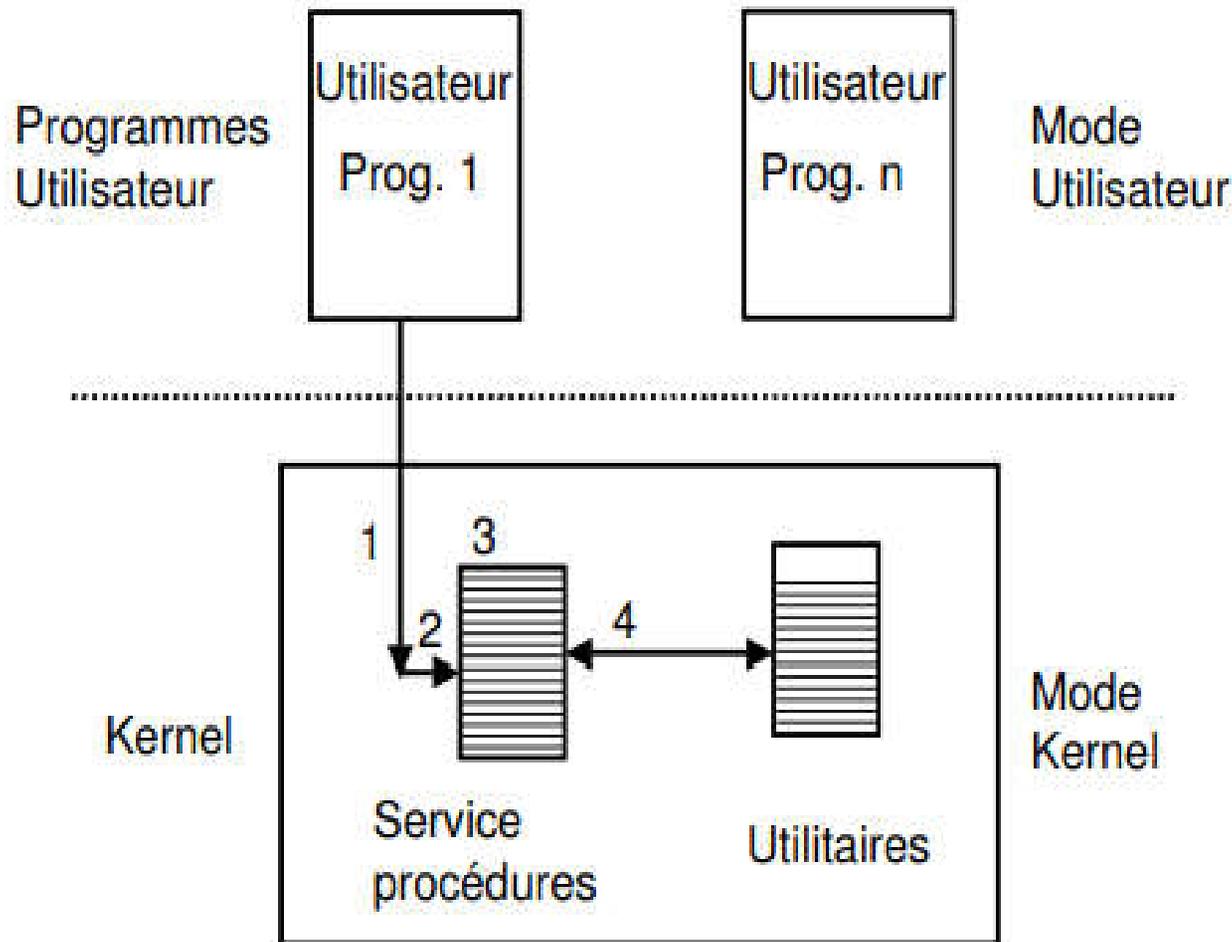
droits du propriétaire : Lire, Ecrire

droits des membres du groupe : Lire, Exécuter

droits des autres : aucun

Structure d'un système d'exploitation

- Structure monolithique: ensemble de procédures de même niveau: une **procédure principale** qui appelle la **procédure de service** requise qui fait appel à son tour à , des **procédures utilitaires** qui assistent les procédures de service (ex: la recherche de données des programmes utilisateur)

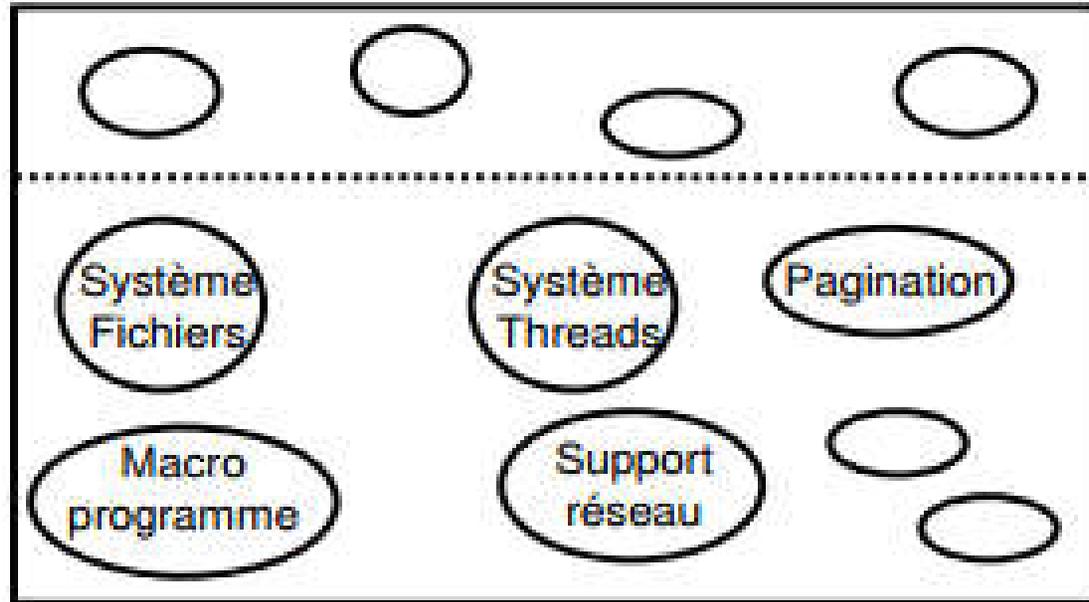


Structure monolithique. (1) Appel système (Mode utilisateur \mapsto Mode kernel). (2) Vérification de paramètres. (3) Appel de la procédure de service. (4) Procédure de service appel utilitaires, et puis retourner au mode utilisateur.

Structure d'un système d'exploitation

- Micro noyau: Architecture plus moderne que la structure monolithique.
- Différence entre les deux architectures se situe dans l'implantation de leurs architectures respectives en mode superviseur (kernel mode) et en mode usager (user mode).
- L'architecture monolithique met en œuvre tous les services du SE dans le domaine du mode superviseur.
- l'architecture micro-kernel fait une division entre les services du SE: services «haut-niveau» implantés dans le domaine de l'utilisateur et «bas-niveau» implantés dans l'espace du mode superviseur.

**Mode
utilisateur**



Processus
utilisateur

Processus
système

**Mode
Kernel**

Support pour primitives :
processus, mémoire
objets, ports, messages

Micro
Kernel

Matériel

Structure de micro-kernel.