

SYSTÈMES D'EXPLOITATION

1^{ère} Licence Sciences de l'Informatique

Dr. Jarray Ridha

Année Universitaire 2024/2025

Bibliographie SE

- Andrew Tanenbaum, Les systèmes d'exploitation, conception et mise en œuvre Interdictions.
- Joffroy Beauquier, Béatrice Bérard, Les systèmes d'exploitation, concepts et algorithmes, Ediscience.
- J.M. Rifflet, La programmation sous Unix, 3e éd., McGraw-Hill, 1993.
- Jean-Michel Levy, Unix & Linux - Utilisation et Administration, Ed Pearson Education, 450 pages, 08/2004, ISBN 2-7440-7049-1.

SYSTÈMES D'EXPLOITATION

Partie 1: Introduction aux systèmes d'exploitation

Dr. Jarray Ridha



Plan général

1

Rappels sur l'architecture des ordinateurs

2

Aperçu général sur les systèmes d'exploitation



Rappels sur l'architecture des ordinateurs

- 1 Définition d'un ordinateur
- 2 Mémoire et caractéristiques
- 3 Types des mémoires
- 4 Le processeur
- 5 Les bus
- 6 Exécution d'un programme

Définitions

Un ordinateur est une machine électronique qui traite une **information** fournie par un organe d'entrée suivant un programme et délivre une information sur un organe de sortie

Définitions

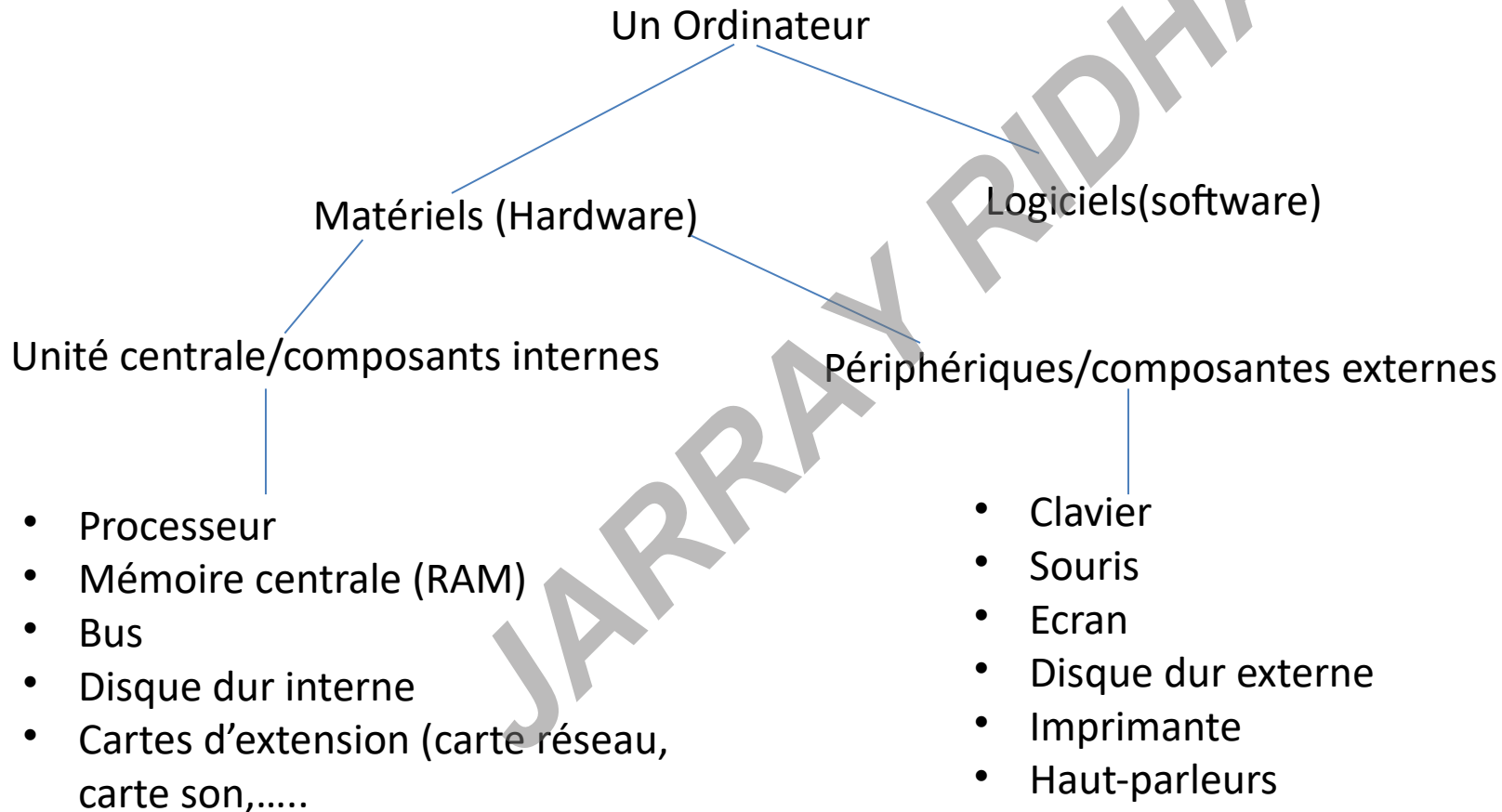
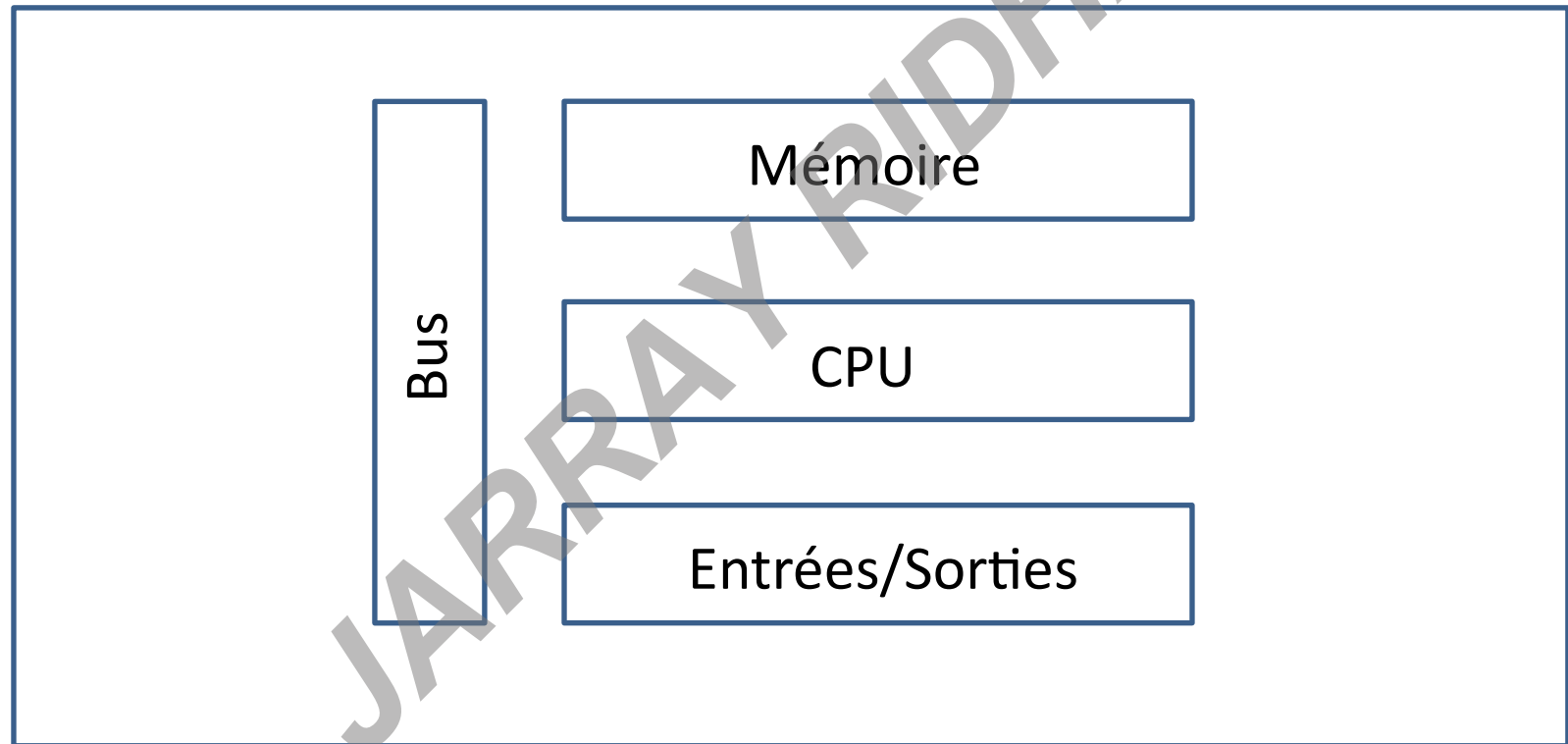


Schéma simplifié de l'architecture des ordinateurs



Mémoire

En informatique , la mémoire est un dispositif capable:

- d'enregistrer une information,
- de la conserver (memoriser),
- et de la restituer (possibilité de la lire ou la récupérer par la suite)

Caractéristiques des mémoires

- La capacité,
- La volatilité
- Le mode d'accès
- Le temps d'accès

Caractéristiques

La capacité d'une mémoire :

- Est la quantité d'informations qu'on peut enregistrer dans cette mémoire
- En informatique, l'information est numérisée, c'est-à-dire limitée à des valeurs discrètes , précisément binaires
- L'élément de base pour la représentation de l'information est le bit
- Le bit est un chiffre binaire qui ne peut prendre que deux valeurs , notées par convention 0 et 1
- Le mot <bit> est la contraction des mots anglais *binary digit*, qui signifient <chiffre binaire>

Caractéristiques

La capacité d'une mémoire :

- C'est le nombre total de bits pouvant être enregistrés dans cette mémoire
- La capacité peut s'exprimer en:
- Bit: un bit est l'unité de base pour la mesure de la quantité de l'information
- Octet: 1 octet = 8 bits
- Kilooctet(KO): 1 kilooctet=1024 octets =
- Mégaoctet(MO): 1 Mégaoctet=1024 KO=
- Gigaoctet(GO): 1 Gigaoctet=1024 MO=
- Téraoctet(TO): 1 Téraoctet=1024 GO=

Caractéristiques

La volatilité:

- Si une mémoire perd son contenu (les informations) en cas de coupure de courant alors la mémoire est dite volatile
- Si une mémoire conserve son contenu lorsque la source d'alimentation est coupée alors la mémoire est dite non volatile (mémoire permanente ou stable)

Caractéristiques

Le mode d'accès à l'information (lecture /écriture)

- Sur une mémoire , on peut effectuer l'opération de lecture (récupérer/ restituer une information à partir de la mémoire et l'opération de l'écriture (enregistrer une nouvelle information ou modifier une information déjà existante dans la mémoire)
- Il existe des mémoires qui offrent les deux modes lecture/écriture, ces mémoires s'appellent mémoires vives
- Il existe des mémoires qui offrent uniquement la possibilité de la lecture (ce n'est pas possible de modifier le contenu), ces mémoires s'appellent mémoires mortes

Caractéristiques

Le temps d'accès

- C'est le temps nécessaire pour effectuer une opération de lecture ou d'écriture
- Par exemple pour l'opération de lecture, le temps d'accès est le temps qui sépare la demande de la lecture de la disponibilité de l'information
- Le temps d'accès est un critère important pour déterminer les performances d'une mémoire ainsi que les performances d'une machine

Types des mémoires

- Un ordinateur est composé par plusieurs types de mémoire:
 - La mémoire principale : la mémoire vive et la mémoire morte
 - Les registres
 - La mémoire de masse
 - La mémoire cache

Mémoire morte

- La mémoire morte (appelée aussi ROM: Read Only Memory) caractérisée par:
 - Un accès uniquement en lecture: le contenu d'une mémoire ROM est fixé à la construction et ne peut plus être changé
 - Sa non volatilité
 - La mémoire morte est utilisé pour stocker les informations nécessaires au démarrage d'un ordinateur (exemple le BIOS qui , entre autres, démarre le système d'exploitation)

Mémoire vive

- La mémoire vive (appelée aussi RAM: Random Access Memory) caractérisée par:
 - Un accès en lecture/écriture
 - Sa volatilité
 - Possibilité d'un accès arbitraire/direct aux données, c'est-à-dire un accès à n'importe quelle donnée en utilisant son adresse
 - Les ROM jouissent de la même caractéristique d'accès direct, mais contrairement aux RAM ne sont pas volatiles

Mémoire vive

- La mémoire vive est utilisée pour stocker temporairement les programmes (instructions + données) à exécuter par le processeur
- Il existe deux grandes familles des mémoires vives : les mémoires statiques (SRAM) et les mémoires dynamiques (DRAM)
- Les mémoires DRAM possèdent une capacité beaucoup plus grande que les mémoires SRAM, elles sont plus simples et moins chers, mais avec un temps d'accès beaucoup plus long

Représentation des informations dans la RAM

- La mémoire vive peut être vue comme une grille où chaque case mémorise un bit
- La mémoire est formée d'un ensemble de mots de même longueur
- Un mot est constitué de 1 ou plusieurs octets consécutifs
- Chaque mot est repérée par un numéro appelé adresse
- Une adresse k bit permet de repérer un mot de p bits parmi

Représentation des informations dans la RAM

- Si le mot est constitué de 2 octets,
- La capacité de la mémoire bits = octets
- Exemple :
K=8, capacité = = 512 octets
K=16, capacité = = 128 O

JARRAY RIDHA

Les registres

- Des mémoires de petite taille (quelques octets) et de meilleur temps d'accès
- Exemples : les registres du processeur (utilisés pour le stockage de données nécessaires pour le fonctionnement du processeur)

La mémoire de masse

- Une mémoire de masse (appelée aussi mémoire auxiliaire) est une mémoire :
 - ✓ de grande capacité
 - ✓ Non volatile
 - ✓ Caractérisée par un accès en lecture/écriture

Les mémoires de masse servent d'éléments de stockage permanent et utilisent pour cela :

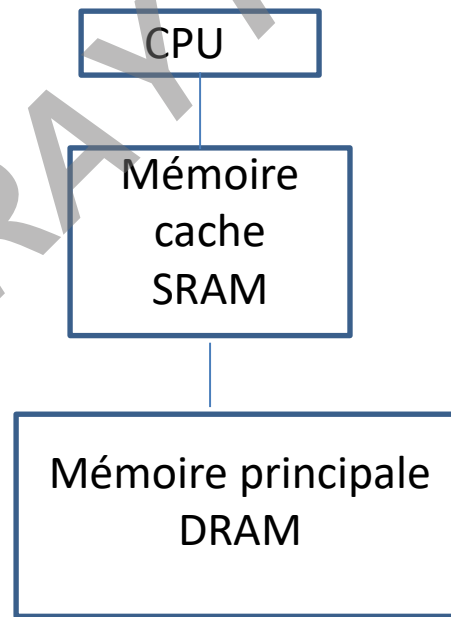
- Des supports magnétiques (disque dur, bande magnétique)
- Des mémoires flash (clé USB)

La mémoire cache/antémémoire

- La mémoire centrale étant lente par rapport à la vitesse du processeur , on utilise une mémoire cache de type SRAM entre le processeur et la mémoire
- La mémoire cache est un type de mémoire vive (RAM) à laquelle le microprocesseur peut accéder plus rapidement qu'à la mémoire RAM habituelle. Généralement, cette mémoire cache est directement intégrée dans la puce de l'unité centrale (UC) ou placée sur une puce distincte dotée d'une interconnexion par bus à l'UC.

La mémoire cache/antémémoire

- Elle est plus rapide et plus proche du processeur, mais plus petite que la mémoire pour laquelle elle sert d'intermédiaire
- Peuvent être situées à l'intérieur du microprocesseur ou en dehors de celui-ci



La mémoire cache/antémémoire

- Utilisations
 - ✓ Installation des logiciels : on installe généralement le système d'exploitation et les applications sur le disque dur de l'ordinateur sur lequel on veut les utiliser
 - ✓ Transfert d'informations: vente de logiciels sur CD-ROM, clé USB, etc.

Définition

Appelé aussi l'unité centrale des traitement (CPU)

- ✓ Il correspond au moteur de la machine (cerveau d'un ordinateur) Il interprète et exécute les consignes reçues en langage binaire. Le processeur est très utile pour le fonctionnement des programmes et l'affichage de votre écran
- ✓ Il exécute les instructions des programmes
- ✓ Un processeur construit en un seul circuit intégré est appelé microprocesseur

Composants essentiels d'un processeur

- ✓ L'unité arithmétique et logique (UAL, en anglais Arithmetic and Logical Unit- ALU) : prend en charge les calculs arithmétiques élémentaires et les tests
 - ✓ L'unité de contrôle : commande et contrôle le fonctionnement de processeur et ses interactions avec la mémoire centrale
 - ✓ Les registres :
 - le registre d'instruction: contient l'instruction en cours de traitement
 - le compteur ordinal : contient l'adresse mémoire de la prochaine instruction à exécuter
- L'accumulateur : ce registre est utilisé pour stocker les données en cours de traitement

Quelques caractéristiques du processeur

- ✓ La fréquence de son horloge exprimée en MHZ ou GHZ
- ✓ Le nombre de bits que le processeur peut traiter ensemble
- ✓ Le nombre de noyaux de calcul (cœurs) :
un processeur multi-cœur est tout simplement un processeur composé non pas de 1 mais de 2(ou 3 ou 4 ou 8) unités de calcul.

Définition

- Un bus est le regroupement de n lignes (fils) permettant l'envoi en parallèle d'un mot de n bits entre deux composants 1 et 2
- Un bus informatique désigne l'ensemble des lignes de communication connectant les différents composants d'un ordinateur

Bus d'adresse (BA)

- Le bus d'adresse permettant l'adressage de la mémoire dans des système à base de processeur, de microprocesseur, de microcontrôleur.
- Les tailles les plus courantes ont été/sont:
 - 16 bits, soit 64K mots,
 - 20 bits, soit 1M mots,
 - 24 bits, soit 16M mots,
 - 32 bits, soit 4G mots,

Bus de données (BD)

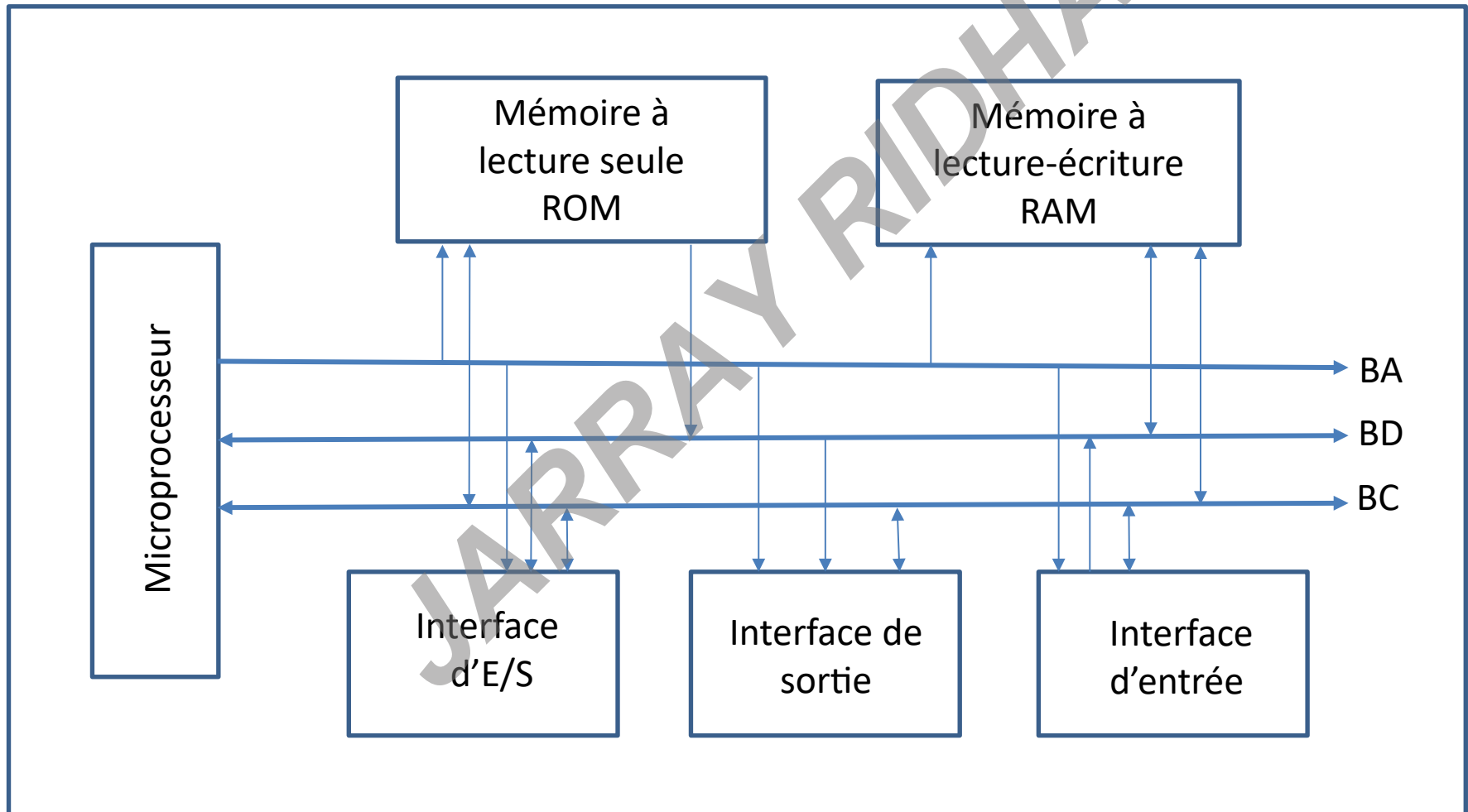
- Transfert les données au sein de l'ordinateur.
- Typiquement , il interconnecte le processeur , la mémoire centrale et les contrôleurs de périphériques. Il est bidirectionnel, contrairement au bus d'adresse.
- Un système 8 bits (16, 32, 64,...) possède un BD 8 bits (16, 32, 64,)= taille de l'info que le processeur peut traiter en une opération élémentaire

Bus de contrôle (BC)

- Sert à spécifier le sens de transfert de la donnée qui circule (ou qui va circuler) sur le bus.
- Ensemble de lignes transmettant des signaux permettant le fonctionnement du microprocesseur, de circuit mémoire, de circuit d'interface....

JARRAY RIDHA

Architecture d'un ordinateur : schéma 2



Exécution d'un programme

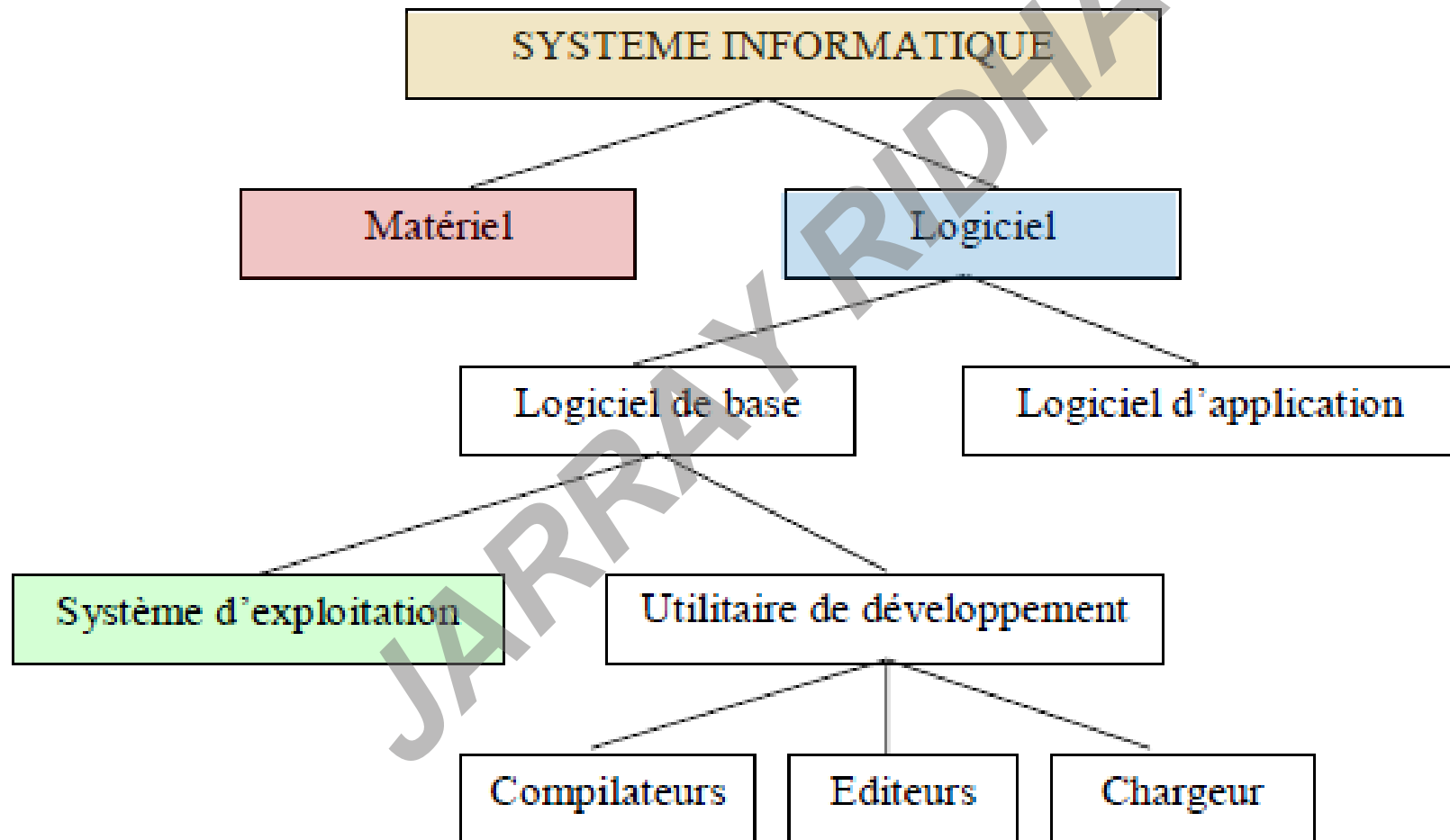
- Le programme et ses données sont d'abord chargés en mémoire centrale puis les instructions du programme sont amenées, une par une, séquentiellement, à l'unité de commande qui les analyse et déclenche les traitements appropriés au niveau de l'unité arithmétique et logique ou les traitements nécessités par les échanges avec les unités d'entrées/sorties ou la mémoire centrale



Aperçu général sur les systèmes d'exploitation

- 1 **Système d'exploitation: Rôles et Définitions**
- 2 **Composantes d'un système d'exploitation**
- 3 **Fonctions d'un système d'exploitation**
- 4 **Les interfaces d'un système d'exploitation**
- 5 **Structures de systèmes d'exploitation**
- 6 **Classes de systèmes d'exploitation**

Un système informatique



Un système informatique

Un système informatique est composé de deux grandes parties :

- **Des ressources matérielles**

- Un regroupement de **composants électroniques** (ordinateur, smartphone, tablette, ...) + un ensemble de **périphériques** (souris, clavier, scanner, clé USB, ...).
- Permettent de : **stocker de l'information, d'exécuter un traitement**, et de fournir un moyen de **communiquer entre eux et avec les humains**.

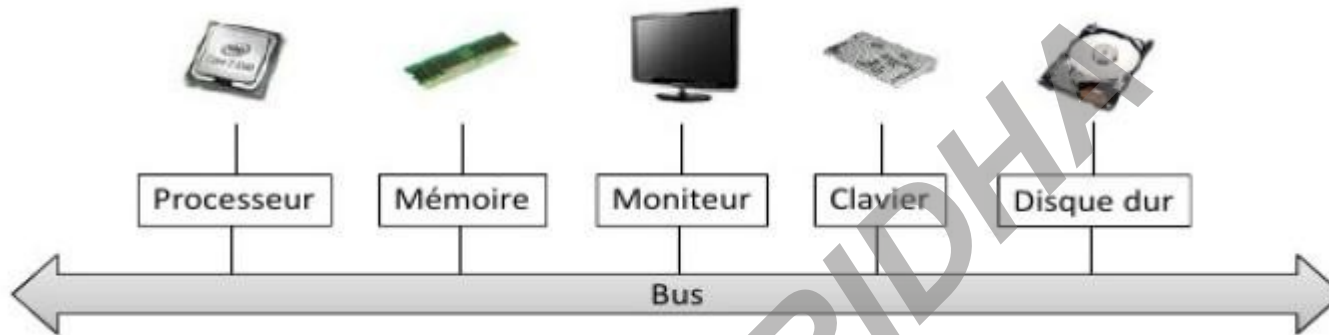
- **Logiciels ou Programmes**

- **Logiciels application.**
- **Logiciels système (ou de base)** : le système d'exploitation en fait partie.



Le système d'exploitation permet d'exploiter ces ressources matérielles.

Structure matérielle (simplifiée) d'un ordinateur



- **Processeur**

- Cerveau de l'ordinateur.
- Exécute les instructions machine.

- **Mémoire**

- Espace contenant les programmes à exécuter et leurs données



- **Périphériques d'Entrées/Sorties (E/S)**

- Acquisition de données (caméra, clavier, souris).
- Production de données (écran, imprimante).



Pourrions-nous faire fonctionner un ordinateur sans système d'exploitation ?

Un ordinateur peut parfaitement fonctionner à **condition** qu'il soit programmé correctement :

- la **programmation** était **étroitement liée** à la machine.
- la **conception** d'un programme nécessitait la **connaissance parfaite** du mode de fonctionnement de la machine.



Complexité que peu de personnes maîtrisaient.

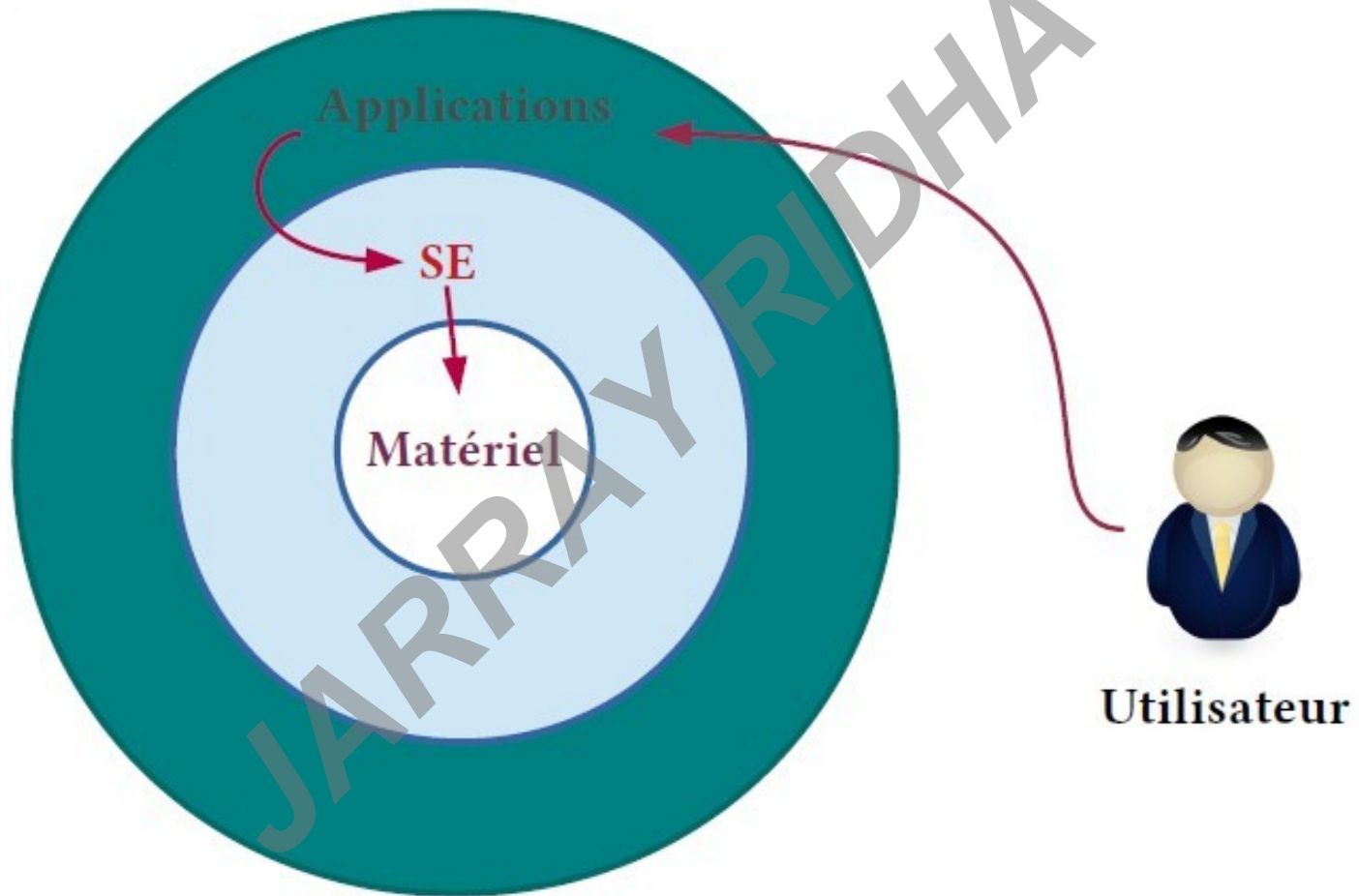


Peu de programmes pouvaient être développés.

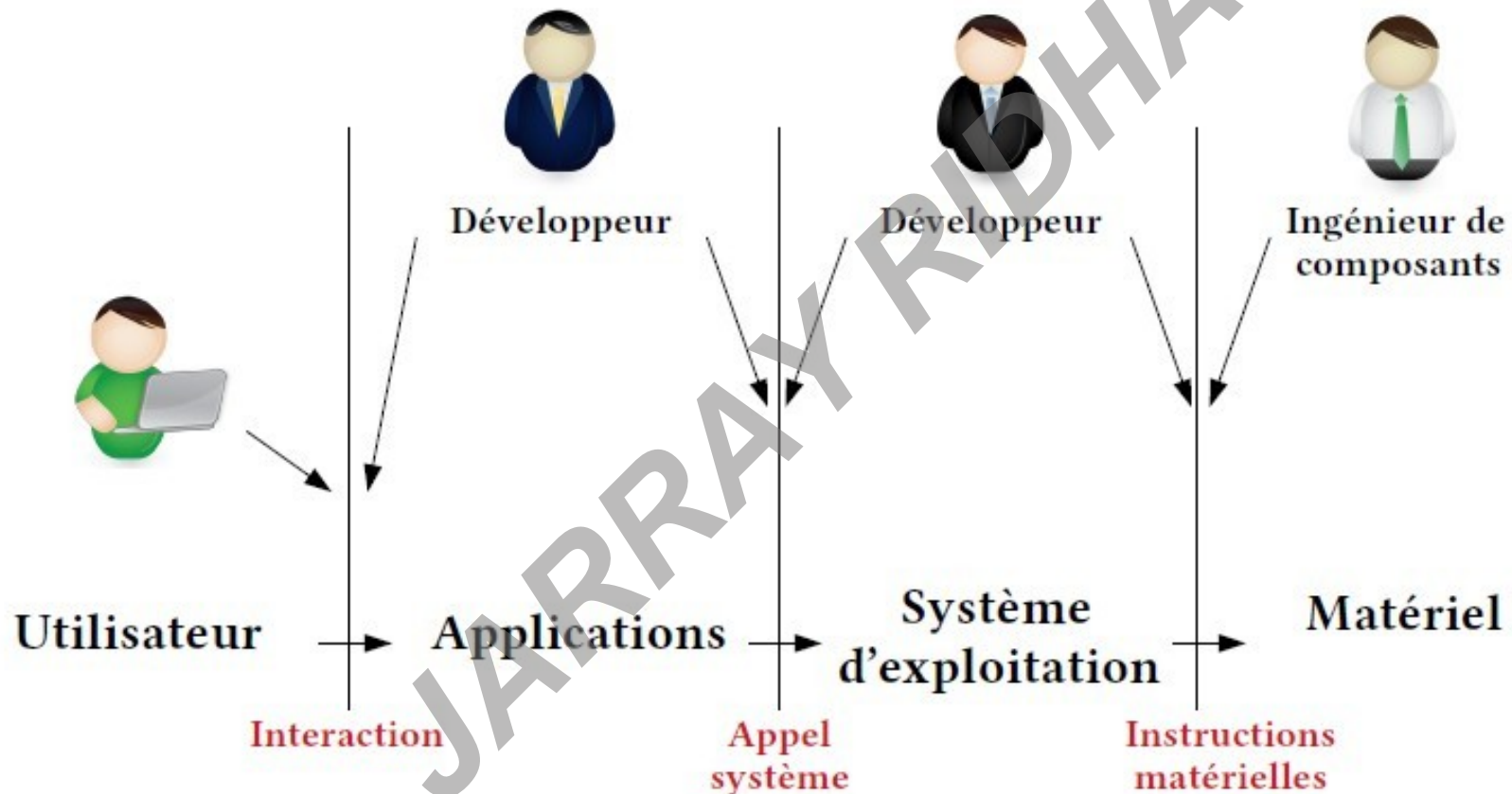


Besoin de développer une couche logicielle pour présenter le matériel aux programmeurs (et aux utilisateurs) comme une machine virtuelle plus facile à comprendre et à utiliser : **un système d'exploitation (SE)** (en anglais : **Operating System (OS)**).

Interactions Homme/SE



Interactions Homme/SE



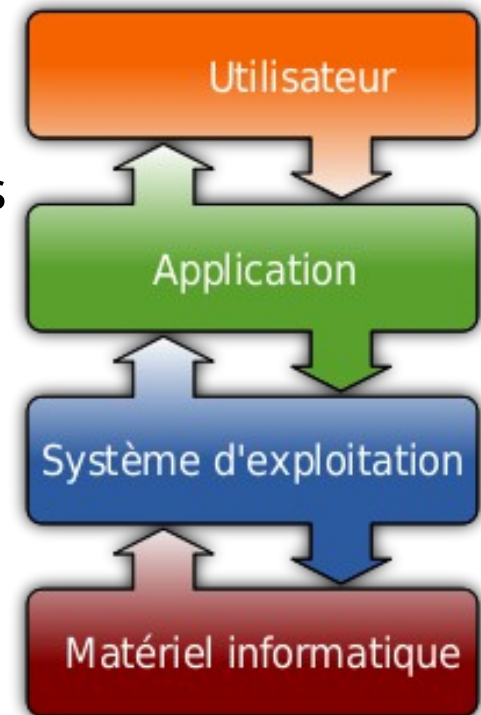
Rôles d'un SE

Définition 1

Le système d'exploitation est l'**intermédiaire** entre un **ordinateur** (ou en général un appareil muni d'un processeur) et **les applications qui utilisent** cet ordinateur ou cet appareil. C'est-à-dire ensemble de programme qui dirige l'utilisation des ressources matérielles d'un ordinateur par des logiciels applicatifs

Le système d'exploitation joue deux principaux rôles complémentaires :

- Une machine virtuelle (abstraite).
- Un administrateur ou gestionnaire de ressources matérielles et logicielles.



Le SE en tant que machine virtuelle

- Le SE présente au programmeur (et utilisateur) une **interface d'accès aux ressources** de l'ordinateur (sous forme d'appels système).
- Cette interface est fondée sur des objets abstraits dont les **éléments** sont les **fichiers** et les **processus**.

Exemple : Le programmeur (utilisateur) voit un disque comme une collection de fichiers qui peuvent être lus, écrits et fermés.

Définition 2

Le **système d'exploitation** est une **couche logicielle** qui permet de **masquer la complexité du matériel** et de proposer **des instructions plus simples** à l'utilisateur.

Le SE en tant que gestionnaire de ressources

- Le SE doit **gérer** les ressources d'une manière **efficace et sure**.
- Le SE doit **ordonnancer** et **contrôler** l'allocation des ressources (CPU, mémoire,...) entre les différents programmes qui y font appels.
- Pour chacune des ressources, le SE doit :
 - **connaître** à tout moment l'utilisateur de la ressources.
 - en **accorder l'usage** de manière **équitable**.
 - **éviter les conflits** d'accès entre les différents programmes ou utilisateurs.

Définition 3

Le **système d'exploitation** est un **gestionnaire de ressources** : il **contrôle l'accès à toutes les ressources** de la machine, **l'attribution** de ces ressources aux différents **utilisateurs** et la **libération** de ces ressources lorsqu'elles ne sont plus utilisées.

Exemples de systèmes d'exploitation

- **MS-DOS (Microsoft)**
- **MacOS (Apple)**
- **Windows (Microsoft)**
- **UNIX (premier par AT&T)**
- **Linux (OpenSource)**



Exemples de systèmes d'exploitation

- **MS-DOS (MicroSoft-Disk Operating System) :**
 - développé en 1980 pour l'IBM PC par B.Gates à partir du noyau DOS.
 - intègre petit à petit des concepts riches d'UNIX
- **Mac OS pour Macintosh d'Apple :**
 - intègre une IHM graphique, destiné à des utilisateurs qui ne connaissaient rien aux ordinateurs.
- **Windows de Microsoft :**
 - successeur de MS-DOS qui intègre une IHM graphique influencé par le succès de Macintosh (Windows 95/98/NT/2000/XP)
- **UNIX, Linux (Linus+Unix) :**
 - intègre de l'IHM graphique avec un système de fenêtres appelé XWindow.

Les composantes d'un système d'exploitation sont :

- **Le noyau (kernel en anglais)**

- la **partie fondamentale** du SE.
- un **ensemble de programmes** qui réside constamment en mémoire centrale.
- **assure les fonctions fondamentales** d'un SE : la **gestion** des périphériques (au moyen de pilotes), des processus, des fichiers et des protocoles réseau.

- **L'interpréteur de commande (Shell)**

- représente une **interface** entre l'**utilisateur** et le **système d'exploitation**.
- permet la communication avec le SE par l'intermédiaire d'un **langage de commandes**.
- **Exemple de commandes :**
ls (sous Unix) : lister les répertoires et les fichiers du répertoire courant



Quelles sont les fonctionnalités d'un SE ?

- Les systèmes d'exploitation ont tous les mêmes objectifs et les mêmes fonctionnalités.
- Les systèmes d'exploitation se différencient par l'interface qu'ils proposent et les algorithmes et stratégies qu'ils appliquent.
- Les systèmes d'exploitation prennent en charge les mêmes fonctionnalités de base :
 - **La gestion du processeur,**
 - **La gestion du dialogue entre l'utilisateur et la machine,**
 - **La gestion de la mémoire centrale,**
 - **La gestion des fichiers et répertoires,**
 - **La gestion des périphériques,**

La gestion du processeur

- Le système d'exploitation gère l'exécution des programmes au niveau du processeur : ceci se ramène à la **gestion des processus**.
- Un **processus** est un programme **en cours d'exécution** : il regroupe le code du programme ainsi que ses données et un ensemble de ressources allouées.
- Le SE est responsable de:
 - **Allocation du processeur aux processus**
 - **Création, terminaison des processus**
 - **Suspension, reprise des processus**
 - **Synchronisation, communication entre processus**

La gestion du processeur

Les **principales tâches** d'un système d'exploitation pour la **gestion du processeur** sont :

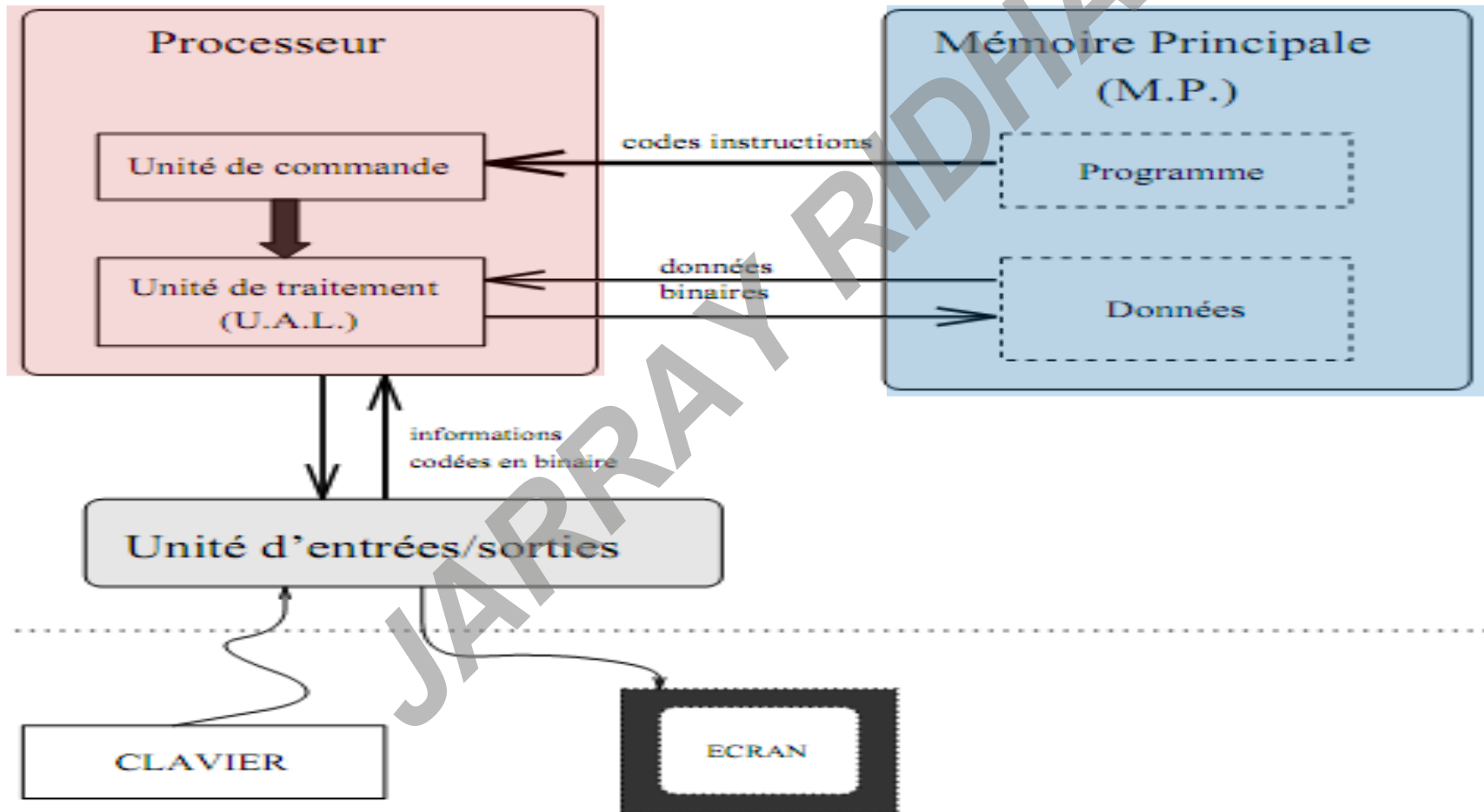
- **Allouer le processeur** aux processus: planifier l'exécution des programmes par le biais d' **un algorithme d'ordonnancement**.
- **Gérer les conflits** dus à la concurrence (les processus peuvent utiliser des variables en commun).
- Offrir **des primitives** pour assurer la **communication** entre les processus ainsi que leur **synchronisation**.
- Effectuer la **correction des situations d'interblocage** entre processus en fournissant les mécanismes adéquats.

La gestion du dialogue utilisateur/machine

La communication entre l'utilisateur et le SE s'effectue par:

- Un langage de commande.
 - Commande sous forme textuelle.
 - L'utilisateur doit connaître le nom de la commande pour effectuer une tâche.
- Interface graphique
 - Il suffit de choisir dans un menu, cliquer sur des objets graphiques.
 - Principe WYSIWYG.

La gestion de la mémoire centrale



La gestion de la mémoire centrale

- Les **principales tâches** d'un système d'exploitation pour la **gestion de la mémoire centrale** sont :
 - **Gérer l'allocation de la mémoire centrale** aux programmes (attribution, libération de mémoire, segmentation),
 - **Gérer les règles d'adressage**,
 - **Assurer** que les programmes en mémoire **ne puissent pas interférer entre eux**.



En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée «**mémoire virtuelle**».

La gestion des fichiers

Le système d'exploitation gère :

- la **lecture et l'écriture** dans le **système de fichiers**,
- les **droits d'accès aux fichiers** par les utilisateurs et les applications.

Exemple: Pour un SE **multi-utilisateurs**, un SE doit assurer la confidentialité des données en protégeant le contenu des fichiers du regard des autres utilisateurs.

Cette gestion permet le **stockage à long terme** des données et programmes sur la mémoire de masse.

La gestion des entrées/sorties

- Le système d'exploitation **contrôle les périphériques** d'entrée/sortie et **gère les flux** de données en entrée ou en sortie.
- Le système d'exploitation **doit** :
 - **émettre** des commandes vers ses périphériques,
 - **intercepter** les interruptions,
 - **gérer** les erreurs.
- Le système d'exploitation permet d'**unifier** et de **contrôler** l'accès des programmes aux ressources matérielles (périphériques d'E/S)
- par l'intermédiaire des **pilotes** (appelés également gestionnaires de périphériques ou gestionnaires d'E/S).

La gestion du réseau

Le système d'exploitation doit **assurer** :

- l'**interconnexion** des machines,
- le **partage** des fichiers utilisateurs,
- une **protection** pour ces fichiers,
- l'**identification** des machines et des utilisateurs connectés au réseau.

Les interfaces d'un système d'exploitation

- Une **interface** d'un système d'exploitation est un **moyen** pour permettre aux **utilisateurs** du système de faire **appel aux fonctions du système**.
- Un système d'exploitation **présente** en général **deux** interfaces :
 - Une interface de **commande**,
 - Une interface **programmative**.

Exemple : Types d'interfaces d'un système Unix

Interface de commande 1

Shell 1

Interface de commande 2

Shell 2

Interface programmatique

Système Unix

Interface de commande/ Interface programmatique

Interface de commande

- **Dédiée** aux **utilisateurs** humains d'un système d'exploitation.
- **Composée** d'un **ensemble de commandes** :
 - **Commandes textuelles** (shell).
 - **Commandes graphiques** (exemple : glisser-déplacer l'icône d'un fichier vers la corbeille).

Interface programmatique (Application Programming Interface (API))

- **Dédiée** aux **programmes** s'exécutant au-dessus d'un système d'exploitation.
- **Composée** d'un **ensemble d'appels systèmes** (appels de procédures, avec paramètres).

Les appels système

- Certains traitements **ne** peuvent **pas** être exécutés et contrôlés **directement** par un **programme utilisateur** car le système ne le permet pas.
- Le **programmeur** doit faire **explicitement appel** aux **services** du système d'exploitation pour **accéder** à certaines **ressources**.

Appel système (System Call, abrégé en Syscall)

Les appels système sont l'interface proposée par le système d'exploitation pour accéder aux différentes ressources de la machine.

Les appels système

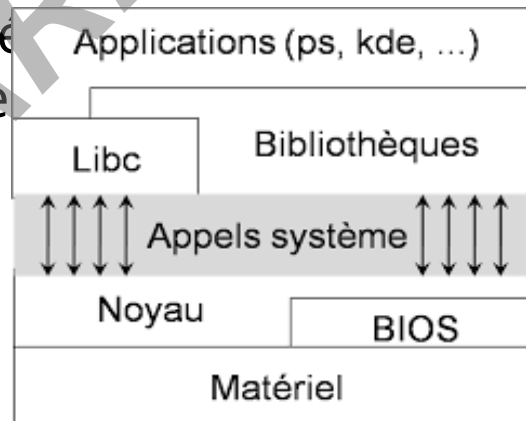
- Pour **chaque composant** de l'ordinateur le **système d'exploitation propose** des appels système.
- **Seuls** ces appels système **peuvent être utilisés**, en effet **le reste des primitives** du système est **protégé** et leur usage direct est **interdit** aux utilisateurs.
- Les **appels système** sont généralement classés en **catégories** :
 - Gestion des processus
 - Gestion des fichiers
 - Communication et stockage d'informations,
 - Gestion des périphériques.
- **Exemples** : fork, exec, exit , kill, wait, sleep, ...
open, read, write, close, ...

Les modes superviseur et utilisateur

- Afin de **protéger** l'exécution du système d'exploitation de celles des programmes utilisateurs, les **processeurs** proposent **deux modes de fonctionnement** :
 - **Le mode utilisateur** dans lequel les programmes utilisateurs sont exécutés.
 - **Le mode protégé ou superviseur** (également appelé **mode noyau**) est le mode d'exécution du système :
 - C'est un mode d'exécution privilégié qui autorise notamment l'appel à des **instructions interdites** en **mode utilisateur** (ex: manipulation des interruptions)
 - Ce mode assure la **protection** du système d'exploitation.
 - **Les appels système** s'exécutent dans ce mode.

Exécution d'un appel système

- Lorsqu'un programme effectue un **appel système**, son **exécution** en **mode utilisateur** est interrompue et le **système prend le contrôle** en **mode superviseur**.
- Un appel système **provoque une interruption** logicielle.
- Le passage du mode utilisateur/superviseur s'accompagne de **commutation** de contexte (sauvegarde du contexte utilisateur, changement de mode d'exécution, restauration du contexte utilisateur).
- Un **appel système** est exécuté en **mode programme noyau** même si le ayant demandé est exécuté en **mode utilisateur**.

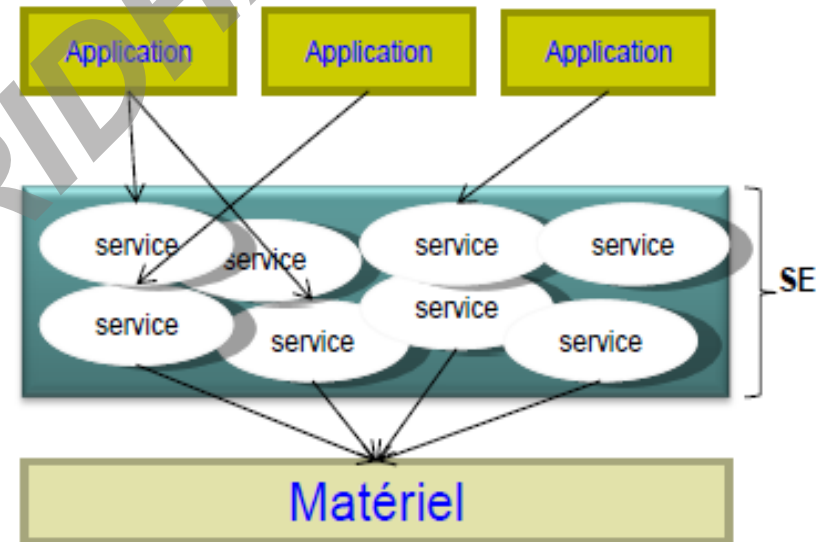


Structures de SE



Systèmes monolithiques

- Le terme **monolithique** désigne un **seul niveau**.
- Un système monolithique est constitué d'un **seul bloc** contenant l'ensemble des **services système** : **une collection de procédures pouvant chacune appeler l'autre à tout instant**.
- Un système monolithique peut **appeler plusieurs** **procédures** **principales** correspondant à un système appel.



Systèmes monolithiques

Avantages

- Ils intègrent une grande quantité de fonctionnalités dans le noyau,
- Ils sont facile dans la conception.

Inconvénients

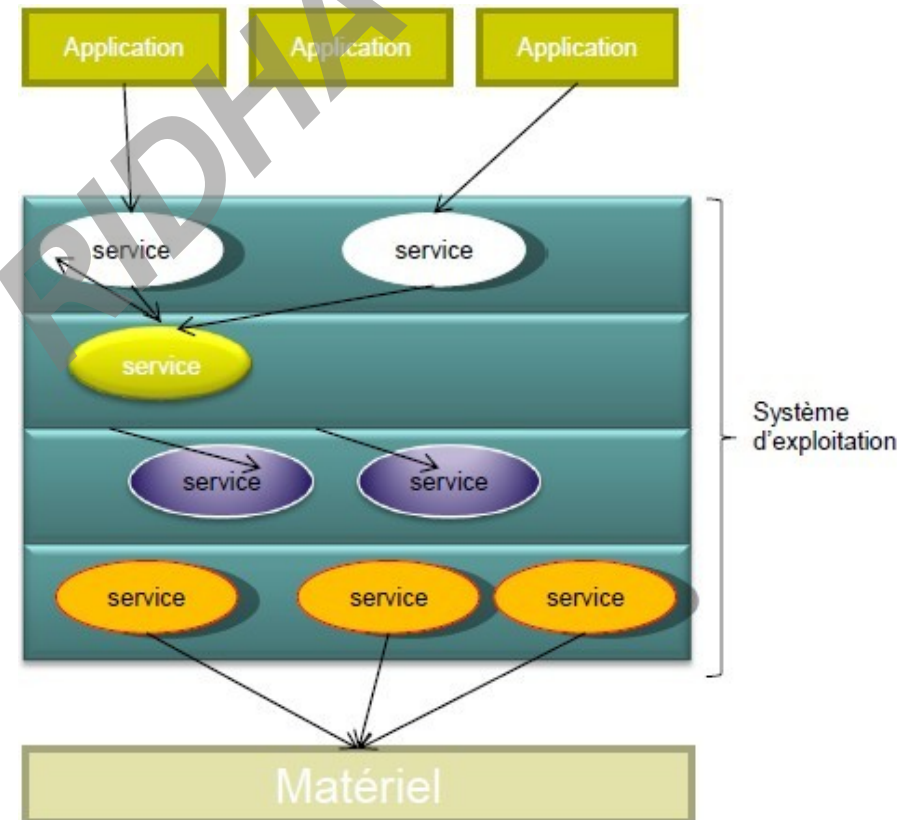
- L'implantation des nouveaux services est très délicate,
- Une erreur à un endroit du noyau peut entraîner un dysfonctionnement à un autre endroit qui, a priori, n'a rien à voir avec le premier,
- Ces systèmes sont relativement lourds à maintenir.

Exemple

MS-DOS, les premières versions d'Unix

Structure en couches

- Ces systèmes sont organisés par **couches** : une **communication** se crée entre **couches adjacentes**.
- Chaque couche **construite sur la base des services offerts par la couche inférieure**.
- L'**intérieur** de chaque niveau est **masqué** par rapport aux autres.



Structure en couches

Avantages

- Le masquage des informations protège les couches entre elles,
- Facilité de conception et de développement (système modulaire).

Inconvénients

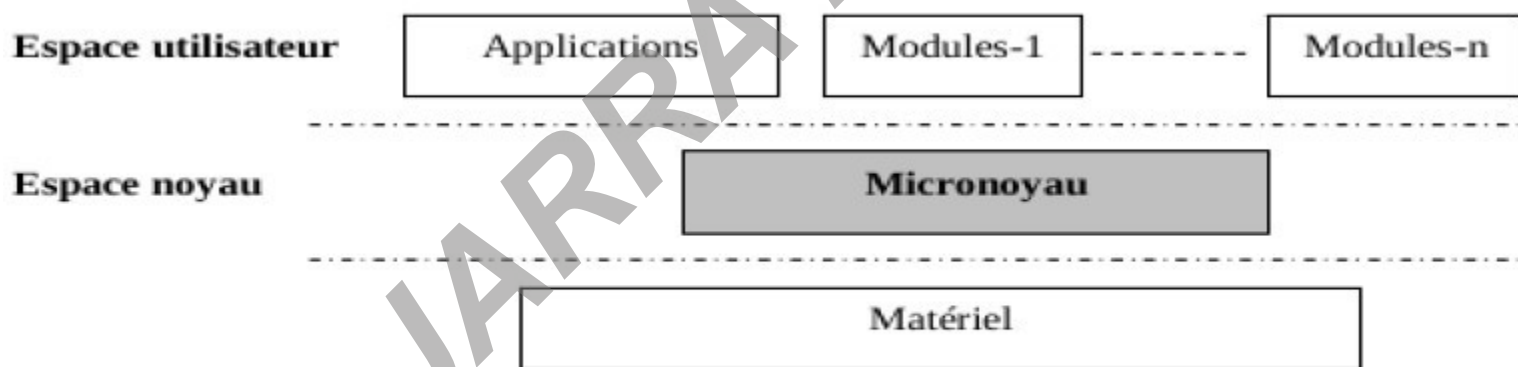
- Un appel système est transmis de couche en couche avant d'être traité peut ralentir son traitement,
- La difficulté de définir de manière adéquate les couches et les frontières entre elles.

Exemple

MULTICS (ancêtre d'UNIX), Linux, BSD, SOLARIS.

Systèmes à micronoyaux (modèle client/serveur)

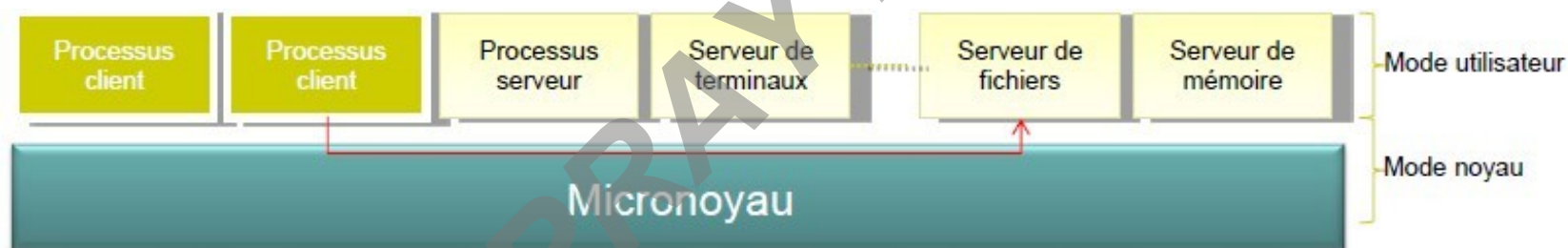
- Un système à **micronoyau** est basé sur :
 - **Un noyau minimal** (appelé **micronoyau**) de taille de code réduite,
 - **Des modules ou des librairies** indépendantes du noyau et qui complètent les fonctionnalités du micronoyau pour former le



Système à micronoyau

Systèmes à micronoyaux (modèle client/serveur)

- Les systèmes à micronoyaux déplacent plusieurs fonctions de SE vers des « **processus serveur** » s'exécutant en **mode utilisateur**.
- Tous les **appels système** ou presque sont désormais des **processus clients**. Le noyau gère la **communication** entre les processus clients et serveurs



- **Fiabilité augmentée**: si un processus serveur « crash », le système continue à fonctionner et il est possible de relancer ce service sans redémarrer.
- **Exemple** : *MAC OS X, GNU HURD, WindowsNT*

Machine Virtuelle

- La machine virtuelle permet de tourner **plusieurs SE** sur la **même machine physique**.
- Le **moniteur** de machine virtuelle (hyperviseur) **intercepte** les instructions privilégiées envoyées par le SE invité, les **vérifie** (politique de sécurité) et les **exécute**.
- **Exemples** : VMWare et VirtualBox.



Classes de SE

Nous pouvons distinguer plusieurs types (classes) de systèmes d'exploitation et les classer selon :

- les services rendus,
- leur architecture logique,
- leur architecture matérielle qui les supporte

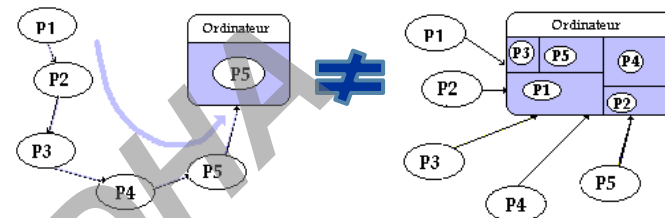
Systèmes mono/multitâches

Système Mono-tâche

- permet de faire fonctionner **un seul programme à la fois**.

Système Multi-tâche

- permet de faire fonctionner **plusieurs applications** simultanément,
- ces applications sont découpées en un ensemble de tâches : **partage du temps du processeur** entre **plusieurs tâches** qui semblent s'exécuter simultanément.



Systèmes mono/multi-utilisateurs

Système Mono-utilisateur

- le système **ne** peut gérer **qu'un seul utilisateur** (ex : le système MS-DOS)

Système Multi-utilisateurs

- autorise l'utilisation simultanée de ce système par **plusieurs utilisateurs** à travers une interface.
- conçu pour que **plusieurs utilisateurs** puissent profiter des **mêmes ressources simultanément**.

Système	Codage	Mono-utilisateur	Multi-utilisateur	Mono tâche	Multitâche
DOS	16 bits	X		X	
Windows3.1	16/32 bits	X			non préemptif
Windows95/98/Me	32 bits	X			coopératif
WindowsNT/2000	32 bits		X		préemptif
WindowsXP/7/8	32/64 bits		X		préemptif
Unix / Linux	32/64 bits		X		préemptif
MAC/OS X	32 bits		X		préemptif

Systèmes centralisés/répartis

Système centralisé

- L'ensemble du système est **entièrement présent** sur la machine considérée.
- Les machines éventuellement **reliées** sont vues comme des **entités étrangères** disposant elles aussi d'un système centralisé.
- Le système **ne gère que** les ressources de la machine sur laquelle il est présent.

Exemple : *UNIX*

Systèmes centralisés/répartis

Système réparti ou distribué

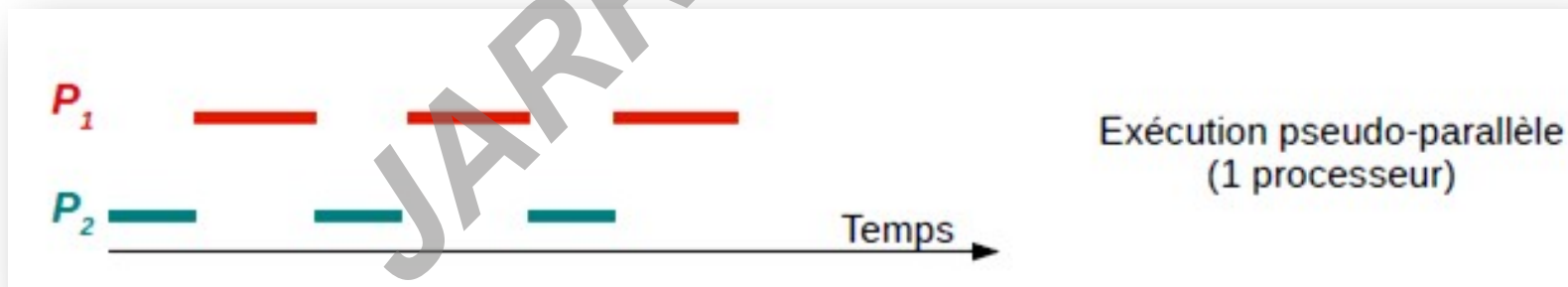
- Un SE distribué permet de gérer l'**ensemble de machines** (domaine) connectées et de gérer le partage des ressources entre eux.
- Le SE réparti **apparaît** aux yeux de ses utilisateurs **comme une machine unique**.
- L'utilisateur **ne se soucie pas** de la **localisation des ressources**. Quand il lance un programme, il n'a pas à connaître le nom de la machine qui l'exécutera.
- Ces systèmes exploitent au mieux les capacités de **parallélisme** d'un domaine et ils sont assez **résistants aux pannes**.

Exemples: Mach, Amoeba, Andrew, Athena, et Locus (Ils ont tous été développés par

Systèmes monoprocesseur/multi-processeurs

Système monoprocesseur

- **Processeur unique.**
- Un mécanisme de gestion des processus pour offrir un **(pseudo) parallélisme** à l'utilisateur : c'est la **multi-programmation** (une commutation rapide entre les différents processus pour donner l'illusion d'un parallélisme).



Systèmes monoprocesseur/multi-processeurs

Système multi-processeurs

- Le **multiprocessing** : technique consistant à faire fonctionner plusieurs processeurs en **parallèle** afin d'obtenir une puissance de calcul plus importante.
- Il y a une grande variété d'architectures **multi-processeurs** :
 - **SIMD** : Tous les processeurs exécutent les mêmes instructions mais sur des données différentes.
 - **MIMD** : Chaque processeur est complètement indépendant des autres et exécute des instructions sur des données différentes.
 - **Pipeline** : Les différentes unités d'exécution sont mises en chaîne et font chacune partie du traitement à effectuer.
- Ce système doit **gérer** le partage de la mémoire entre plusieurs processeurs et également **distribuer la charge** de travail.

Systèmes embarqués

Systèmes embarqués

- Ces systèmes d'exploitation sont **prévus** pour **fonctionner** sur :
 - des **machines** de **petite taille** (PDA, des smartphones, ...),
 - ou des **appareils électroniques autonomes** (sondes spatiales, robot, ordinateur de bord de véhicule, etc.)
- Les systèmes embarqués sont **caractérisés** par :
 - leur gestion avancée de **l'énergie**,
 - leur capacité à fonctionner avec des **ressources limitées**
(mémoire limitée, processeur lent ,...).
- **Exemples:** Windows Mobile, Symbian OS, iPhone OS, Android, BlackBerry OS.

Systèmes temps réel (RTOS)

Systèmes temps réel (RTOS)

- Ce sont des systèmes pour lesquels l'**exécution** des programmes est soumise à des **contraintes temporelles**.
- Les résultats de l'exécution d'un programme **n'est plus valide au delà** d'un **certain temps** connu et déterminé à l'avance.
- Un exemple typique est celui des systèmes d'ordinateurs qui contrôlent un **processus de fabrication** dans une usine (comme le montage de véhicule)
- **Exemples** : OS-9 ; RTLinux (RealTime Linux) ; QNX; VxWorks.



Merci pour votre attention

JARRA / RIDHA