



le cnam

## Polynésie française



le cnam  
Polynésie française





le cnam  
Polynésie française



**Licence 3  
informatique  
Module UTC502**



# Licence 3 Informatique

***UNITÉ D'ENSEIGNEMENT UTC502***  
***Principes fondamentaux des***  
***Systèmes d'Exploitation***

***FORMATEUR***  
***Jean-Denis Girard***  
***jd.girard@sysnux.pf***  
***87.797.527***

# Sommaire

Constitution d'un ordinateur

Systemes d'exploitation

- Les mémoires

- Les processus

- Les systèmes de fichiers

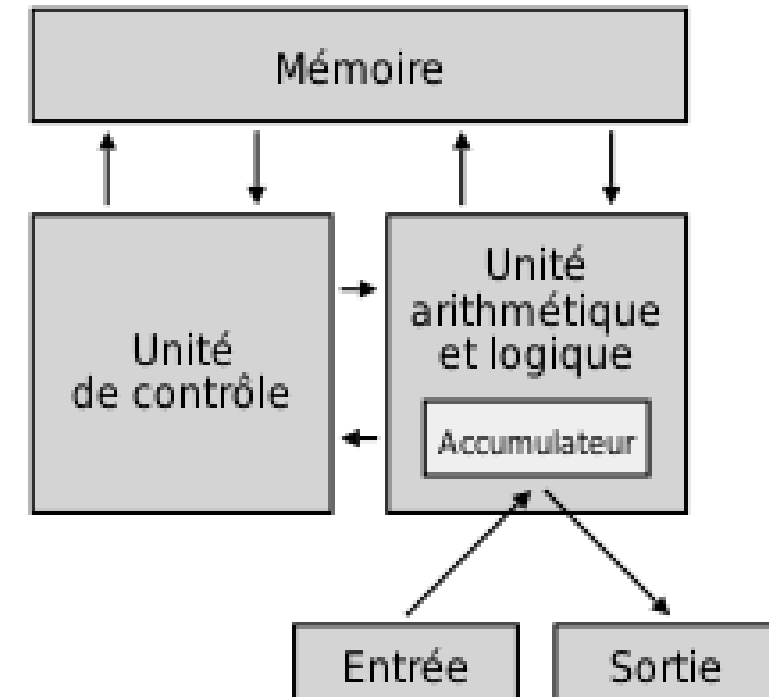
- Les entrées / sorties

Introduction aux systèmes distribués

# Constitution d'un ordinateur

Machine de **Turing**, architecture de **Von Neumann** (1945)

- **CU** : gère le séquençement des opérations
- **ALU** : traite les données élémentaires (registres)
- **Mémoire** : stocke les données et les programmes
- **E / S** : communiquent avec le monde extérieur



# Ordinateur personnel (PC)

Carte mère :

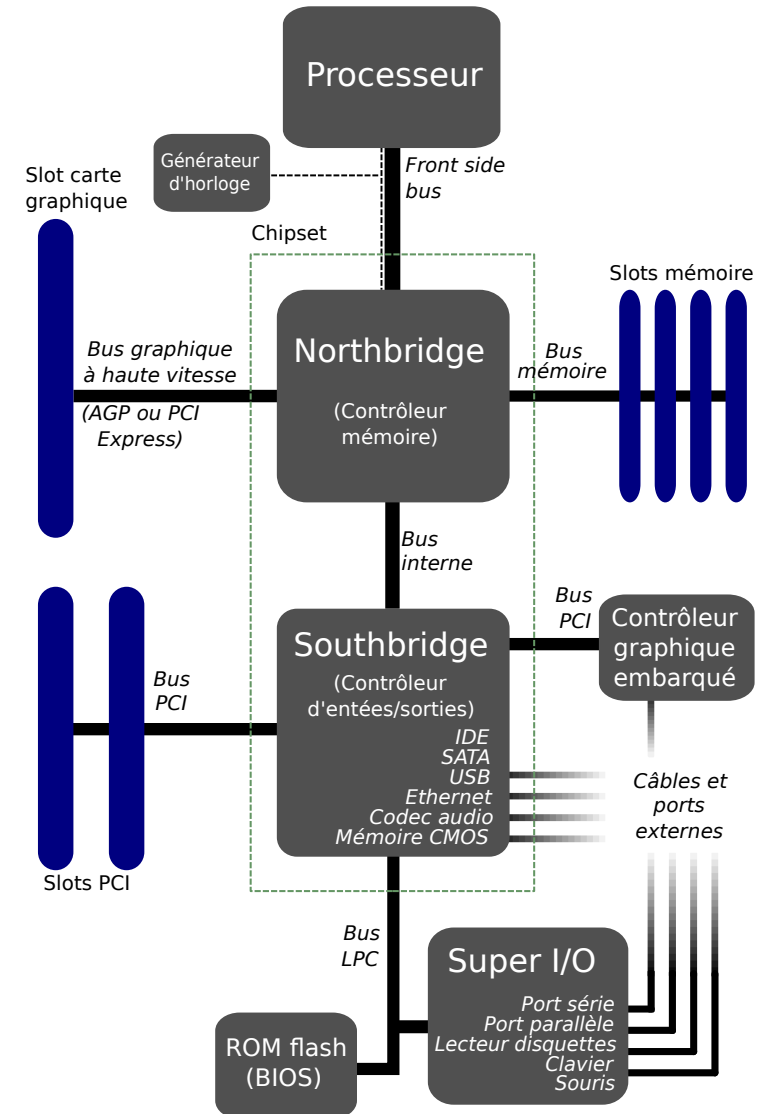
- micro-processeur (**CPU**),
- mémoire vive (**RAM**) ou morte (**ROM**),
- carte graphique (**GPU**),
- carte son,
- carte réseau

Stockage (disque dur, SSD, ...)

Clavier, souris

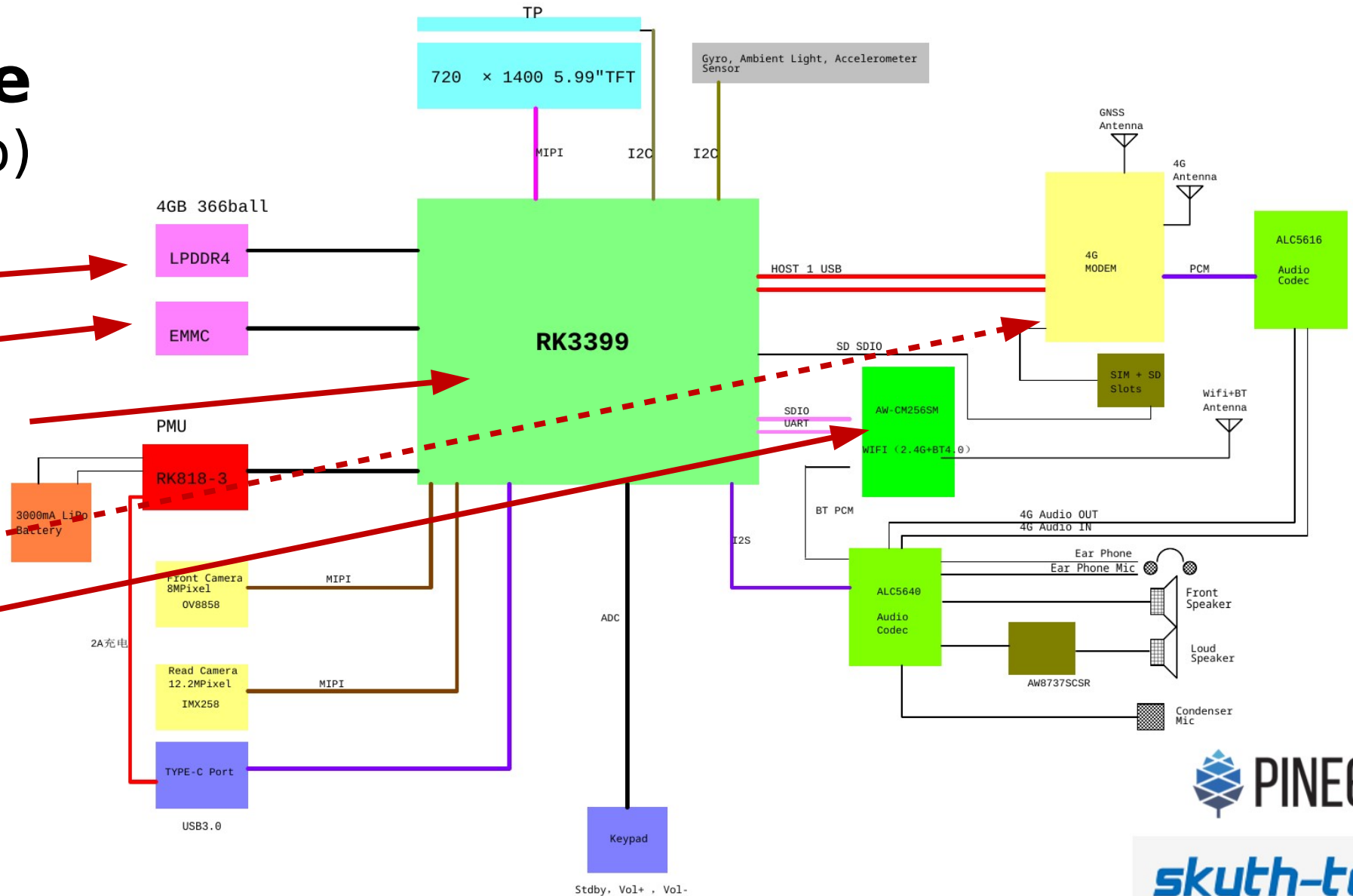
Écran

Autres périphériques...



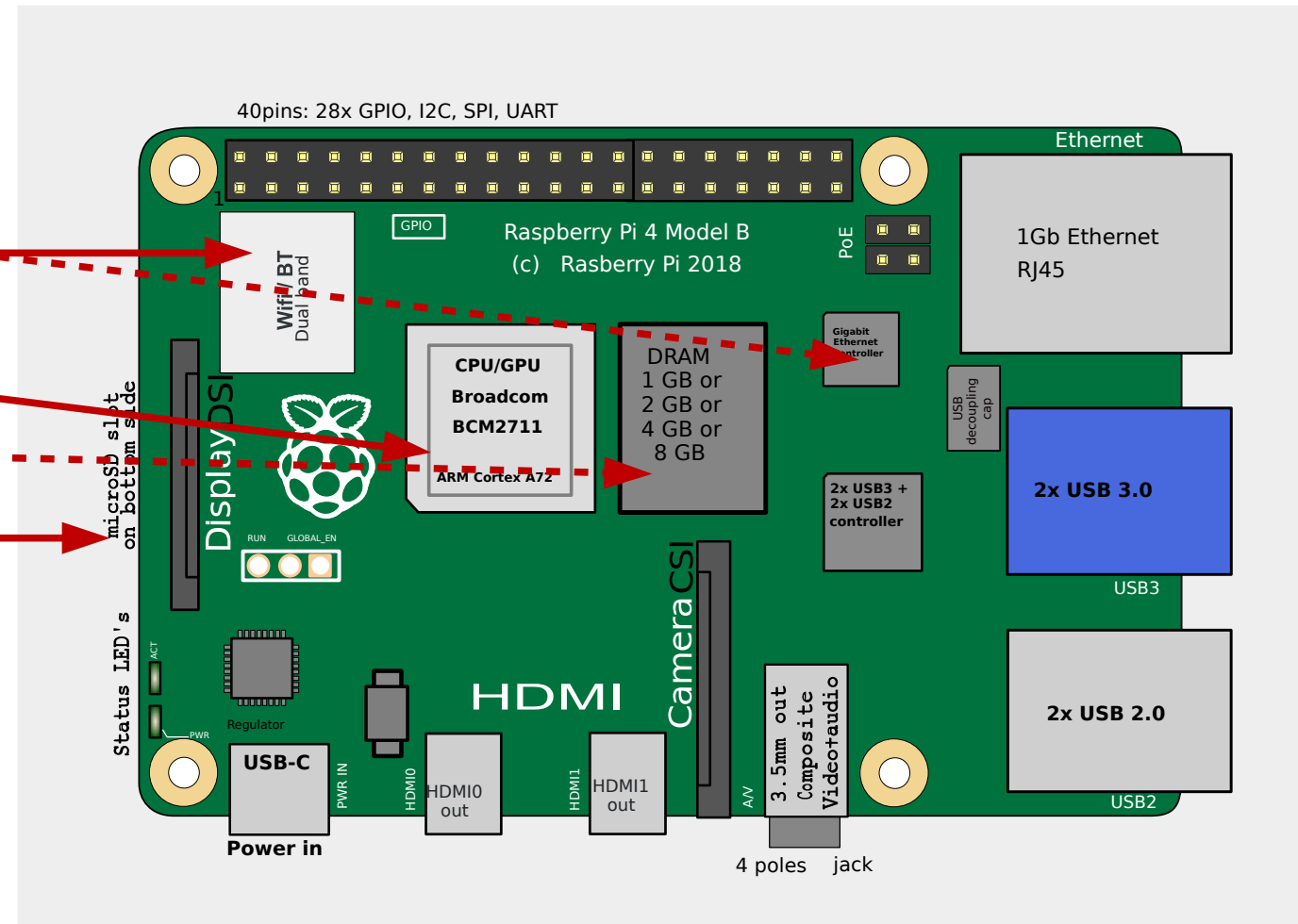
# Smartphone (Pinephone Pro)

**RAM**  
**Stockage**  
**SOC (CPU +  
E/S, + ...)**  
**4G**  
**WiFi / BT**



# Raspberry Pi

Ordinateur au format carte  
de crédit :  
réseau (Ethernet et WiFi),  
CPU, GPU, son,  
RAM,  
disque (MicroSD),  
entrées / sorties (USB +  
GPIO)





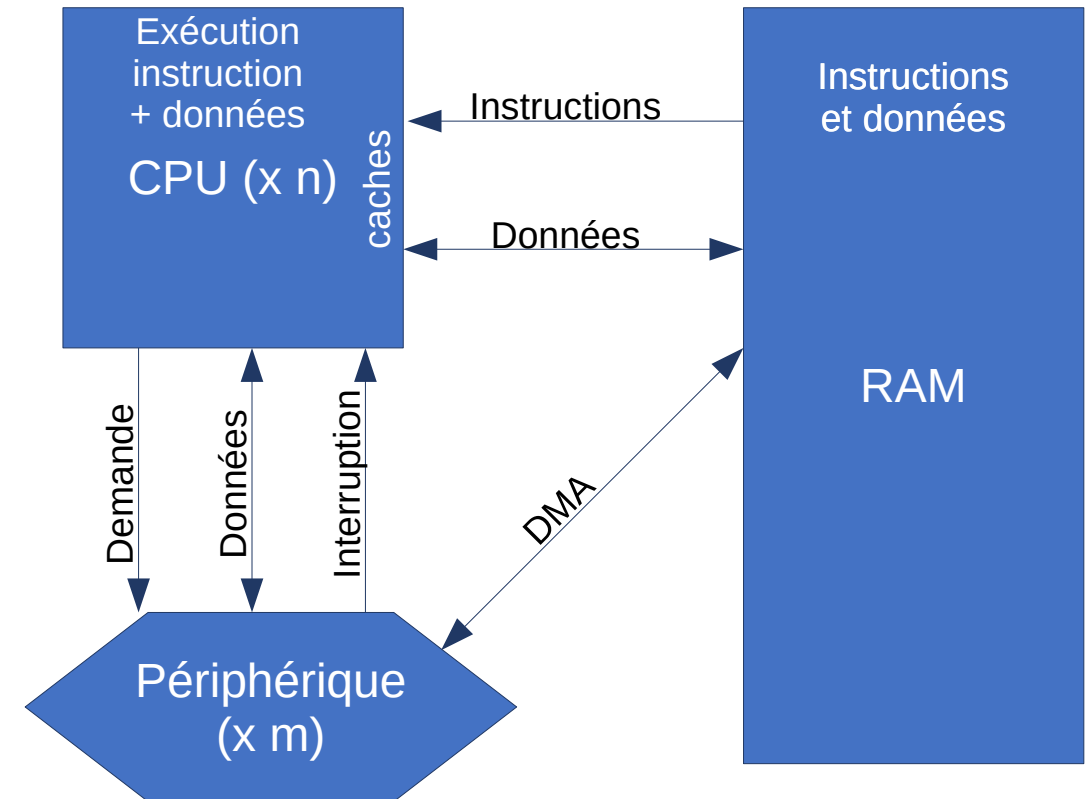
# Fonctionnement

## CPU :

- compteur d'instructions
- charge instruction en RAM
- charge (éventuellement) donnée(s) en RAM
- exécute instruction sur les données

## Si nécessaire, accès périphérique :

- initié par CPU,
- éventuellement DMA : accès direct mémoire pour libérer CPU.

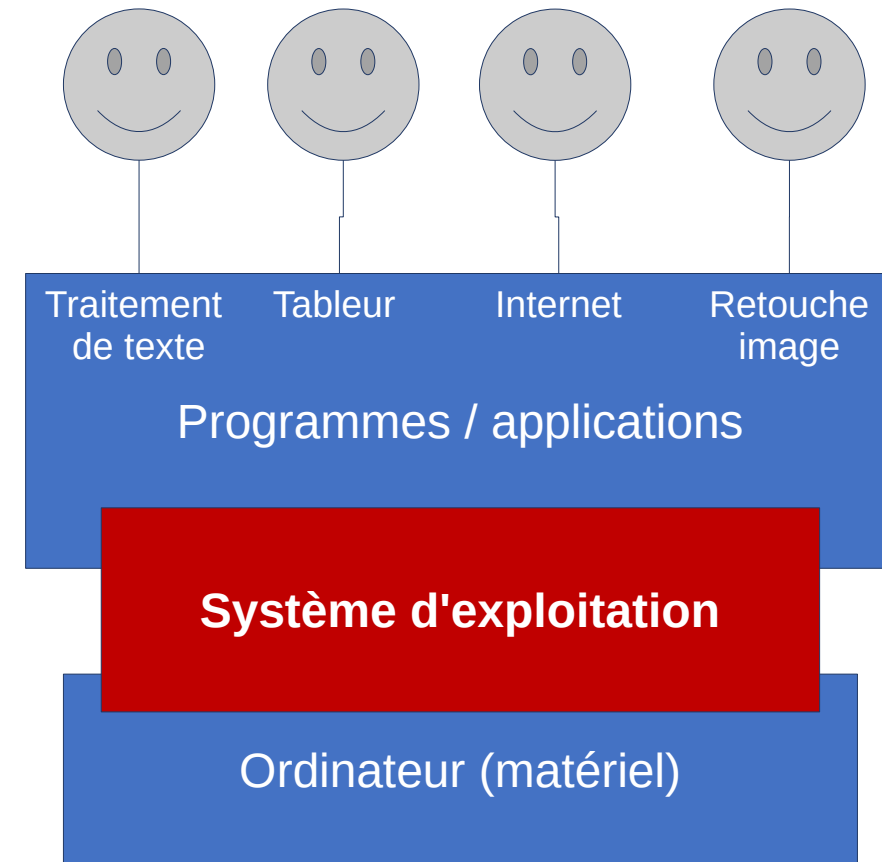


# Systeme d'exploitation (OS)

Logiciels qui gèrent le matériel constituant l'ordinateur, en donnant accès à ses ressources.

Exécuté après l'amorçage (*boot*), jusqu'à arrêt du système.

Intermédiaire entre la machine et les applications exécutées par les utilisateurs : environnement.





Définition ? noyau (*kernel*) +  
ensemble de programmes résidents (*daemons*)

## Caractéristiques

- multiutilisateur,
- multitâche,
- multiprocesseur, multicœur,
- standardisation,
- sécurité,
- code libre / propriétaire,
- licence,
- coût !

## Point de vue utilisateur

- très variable selon utilisation (travail jusqu'à smartphone)
- facilité d'utilisation,
- fonctionnalités,
- fluidité,
- fiabilité.

Éventuellement pas d'utilisateur !

# Histoire

Pas d'OS, instructions directes !

FORTRAN, COBOL compilation, lien, exécution (cartes, bandes, imprimantes !)

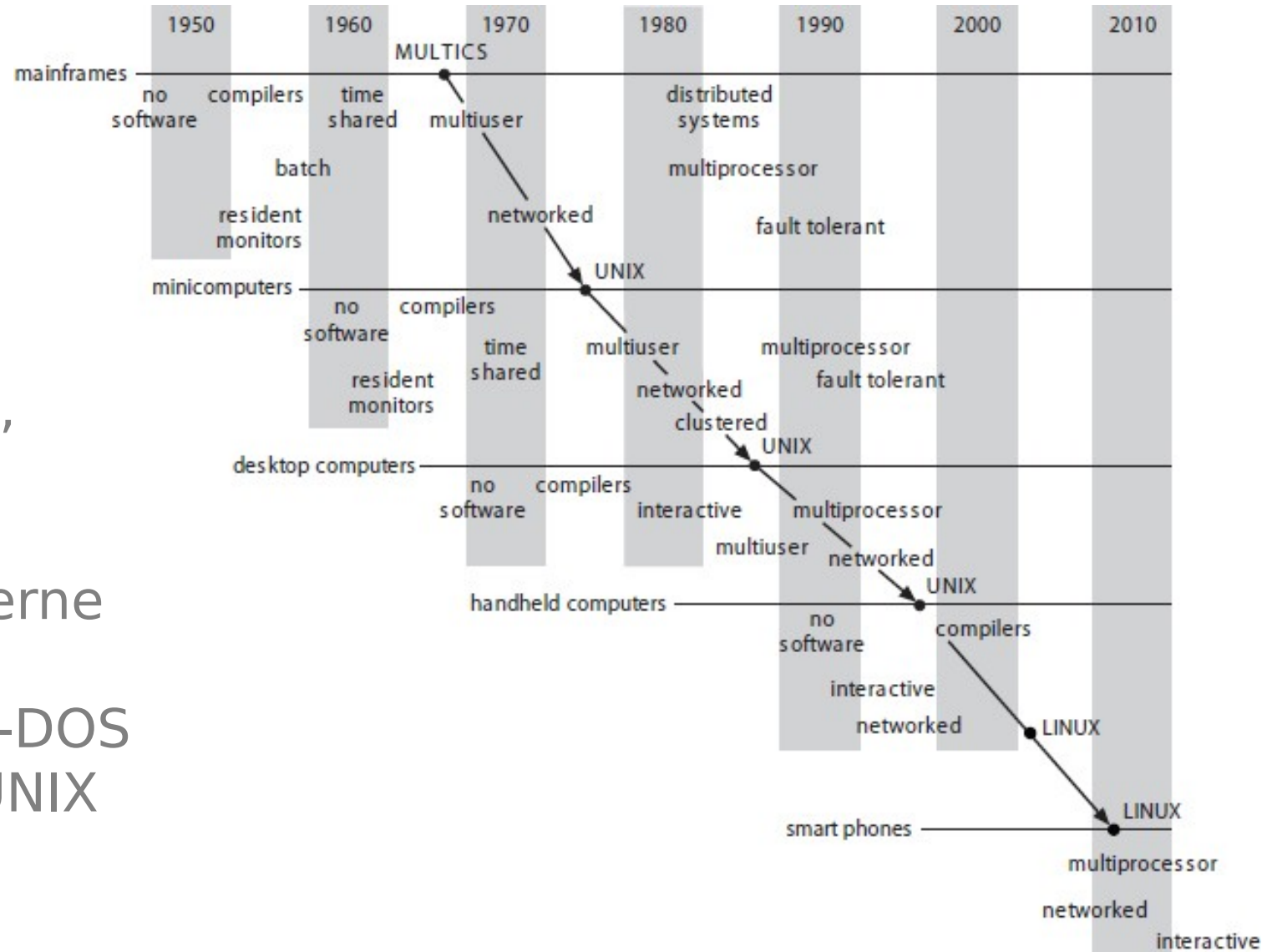
Systèmes à **temps partagé**

OS/360 (IBM) premier OS moderne  
UNIX

Microprocesseurs : CP/M et MS-DOS

Mac OS, Windows, et dérivés UNIX

Android, iOS



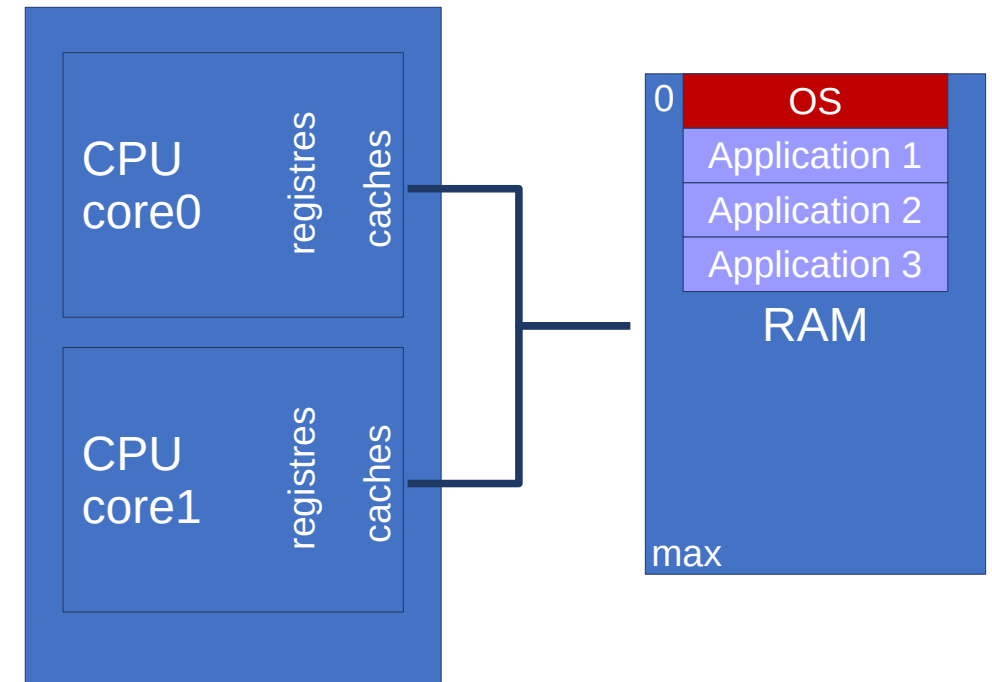


# Fonctionnalités

L'ordinateur exécute de nombreuses applications chargées en RAM en parallèle. L'OS attribue la mémoire et veille au **cloisonnement**.

Les processeurs travaillent sur quelques **registres**, avec un accès instantané.  
Accès RAM très lent pour le CPU, amélioré par cache(s).

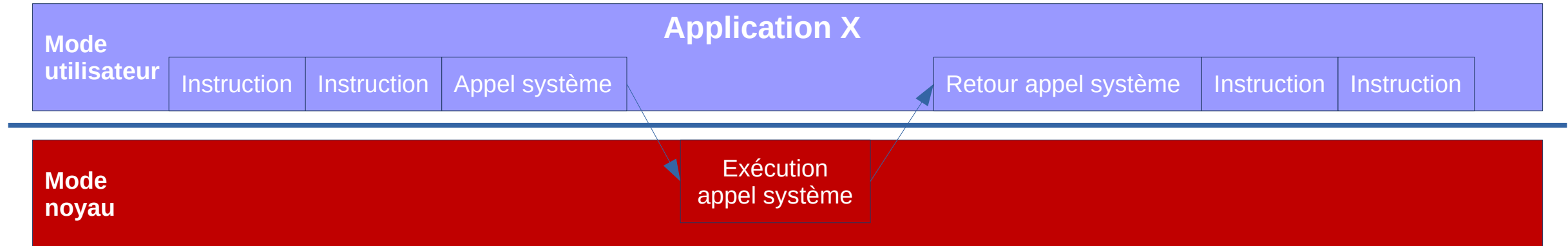
La vitesse d'accès à la mémoire peut être indifférencié ou au contraire privilégié selon le processeur : **NUMA**.



Les applications fonctionnent dans le **mode utilisateur**, avec des droits restreints.

L'OS fonctionne dans le **mode noyau ou mode privilégié**, sans restriction de droits.

Lorsque nécessaire, l'application exécute un **appel système** pour accéder à des ressources protégées.



Les modes sont gérés et définis par les CPU. En cas de violation de privilège, l'exécution est transférée à l'OS, qui peut **terminer** le programme responsable, afficher une erreur...



# Processus

Les programmes ou applications en cours de fonctionnement sont appelés **processus** (*process*). Plusieurs dizaines ou centaines de processus semblent s'exécuter sur l'ordinateur à un instant donné, ce qui n'est pas possible en raison des ressources limitées.

Le système d'exploitation doit gérer au mieux les processus (utilisateur ou système) :

- exécuter le programme -> création du processus,
- lui allouer des ressources (temps CPU, RAM, fichiers, ...) au démarrage et pendant toute sa durée de vie,
- suspendre et relancer le processus,
- vérifier les accès,
- fournir des mécanismes de synchronisation et de communication.

Exprimé en octets (byte), soit 8 unités binaires (bit).

Multiples  $k = 10^3$ ,  $M = 10^6$ ,  $G = 10^9$ ,  $T = 10^{12}$ .

Multiples  $ki = 2^{10}$ ,  $Mi = 2^{20}$ ,  $Gi = 2^{30}$ ,  $Ti = 2^{40}$ .

L'accès au stockage est un goulot d'étranglement pour le CPU.

Niveau	1	2	3	4	
Nom	Registres CPU	Cache CPU	RAM	SSD	DD
Taille	ko	Mo	Go	To	10 To
Technologie		SRAM	DRAM	Flash	Magnétique
Temps accès (ns)	0,1	1	100	10 000	1 000 000
Transfert (Mo / sec)	100 000	10 000	1 000	500	100
Gestion	Programme (compilateur)	Matériel	OS	OS	OS

## Entrées / sorties

L'OS cache le fonctionnement bas niveau (matériel) des périphériques. Par exemple, un principe d'UNIX est que tout périphérique est considéré comme un fichier.

Anciennement les périphériques devaient être déclarés ; de nos jours le fonctionnement est complètement dynamique.

L'OS s'occupe :

- de détecter le matériel,
- de charger le gestionnaire de périphérique adapté (*device driver*),
- d'assurer les échanges avec le matériel.



# Sécurité

Le système d'exploitation met en place des mécanismes de **protection**, afin d'éviter qu'un utilisateur accède aux fichiers d'un autre utilisateur, qu'un processus accède à la RAM du noyau ou d'un autre, qu'un processus monopolise le CPU, etc.

Chaque utilisateur a un identifiant unique (uid *user id* ou sid *security id*) ; lorsqu'il lance un programme, le processus est associé à cet identifiant.

Le processus lui-même dispose d'un identifiant unique (pid, *process id*). Le partage de ressources est possible grâce aux groupes d'utilisateurs, à nouveau identifiés de manière unique (gid *group id*).

Des mécanismes permettent d'étendre ce modèle simple (SELinux par ex.), ou donner des privilèges spécifiques (sudo par ex.).

# Système d'exploitation **temps réel**

De nos jours, la majorité des ordinateurs en fonctionnement sont des machines dites **embarquées**. Elles sont conçues pour une tâche précise, par exemple relever des données météo, et les transmettre via réseau 4G.

Un système d'exploitation généraliste partage les ressources entre ses différents processus, mais n'offre pas de **garantie** sur le temps alloué à chaque processus.

Un système d'exploitation temps réel (**RTOS**) garantit qu'une entrée verra sa réponse traitée dans un **temps déterminé**.

Des OS spécifiques existent, ainsi que des extensions à des OS généralistes.

# Gestion des processus

Un processus est un programme en cours de fonctionnement. C'est l'unité de travail dans un système à temps partagé moderne. Il a besoin d'utiliser le CPU, la RAM, probablement des fichiers, des périphériques.

Les processus peuvent se reproduire (*fork*) ; le résultat est un nouveau processus, avec son propre contexte (en particulier nouvel identifiant, mémoire).

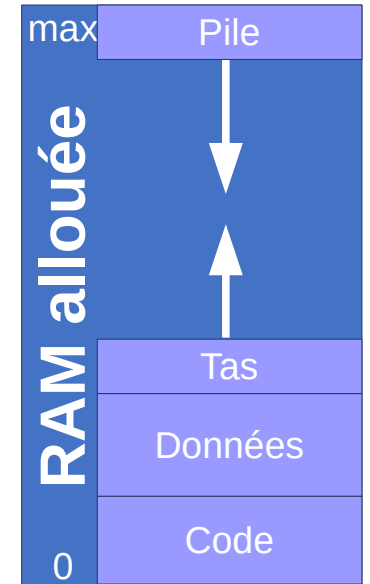
De nos jours les processus peuvent eux-mêmes lancer plusieurs unités de traitement, ou processus légers (*thread*) ; ici les ressources sont partagées.



L'OS charge d'abord le programme et ses données en RAM. Il alloue deux espaces supplémentaires pour les données dynamiques :

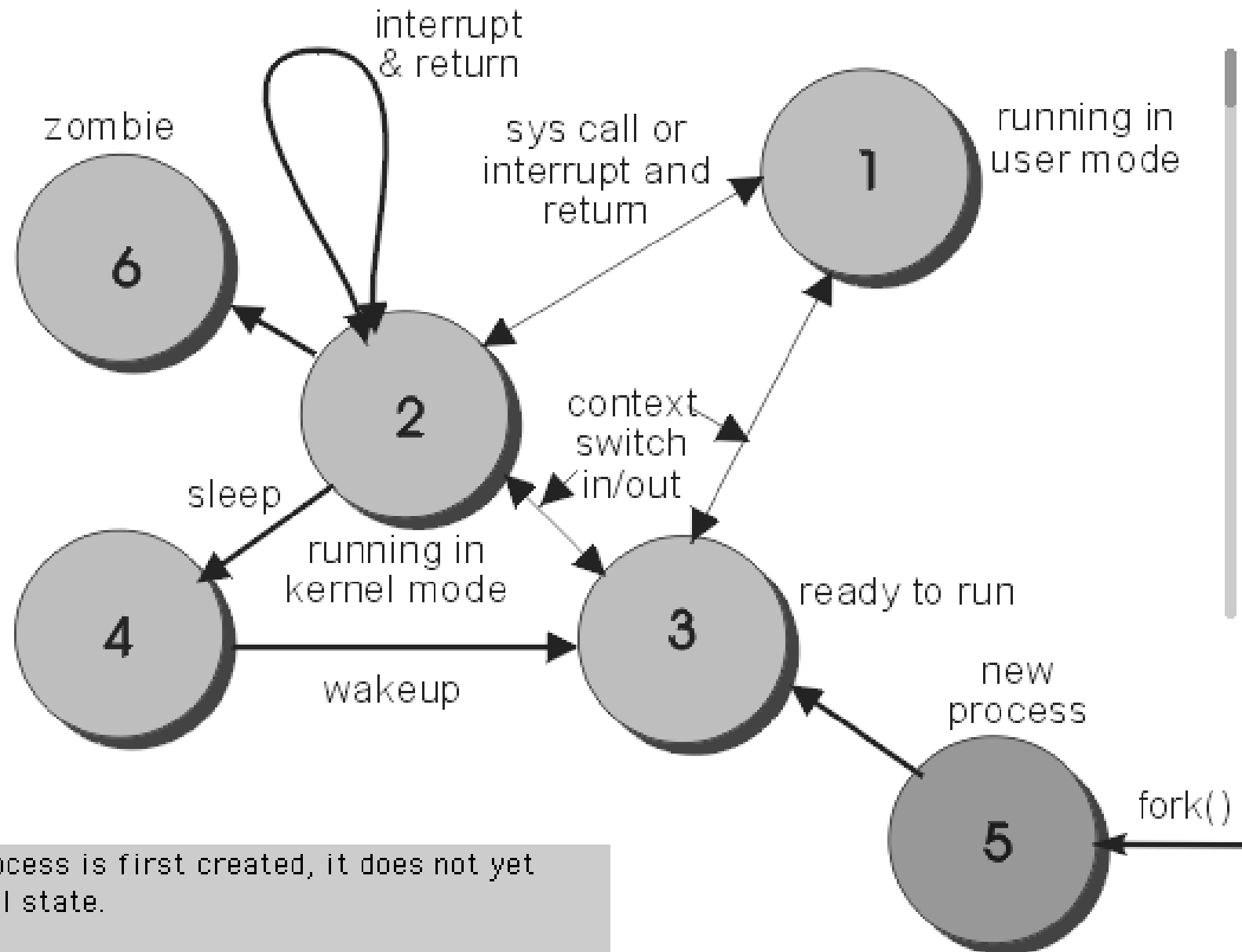
- la pile (*stack*), utilisée dans les branchements,
- le tas (*heap*), pour la création de variables dynamiques.

La taille de ces deux zones varie (en sens opposé) pendant l'exécution.



Le processus évolue entre différents états. L'OS maintient en mémoire des infos pour chaque process (**Process Control Block**) :

- état,
- compteur instruction,
- registres CPU,
- infos séquençement,
- infos mémoire,
- statistiques,
- entrées / sorties.



L'OS contrôle le fonctionnement de chaque processus, en planifiant l'affectation des ressources à tour de rôle : *scheduler*. Il doit s'assurer que chaque processus reçoit la part dont il a besoin, et qu'aucun processus ne bloque les autres.

On parle de changement de contexte (*context switch*) lorsque le contrôle est transféré à un autre processus ; les processeurs peuvent prévoir et améliorer ce fonctionnement.

Le multitâche est aussi supporté sur les OS pour mobiles (iOS 4 et Android).



« **Merci de votre  
attention** »

**Contact**

mei.howan@lecnam.net

40 43 25 44

**Contact Cnam**

**secretariat**

secretariat@cnam-polynesie.com

40 43 25 44