Centre Universitaire de Mila

2 ème année licence LMD Informatique

Module : Systèmes d'exploitation 1

Bessouf Hakim

CHAPITRE 2:

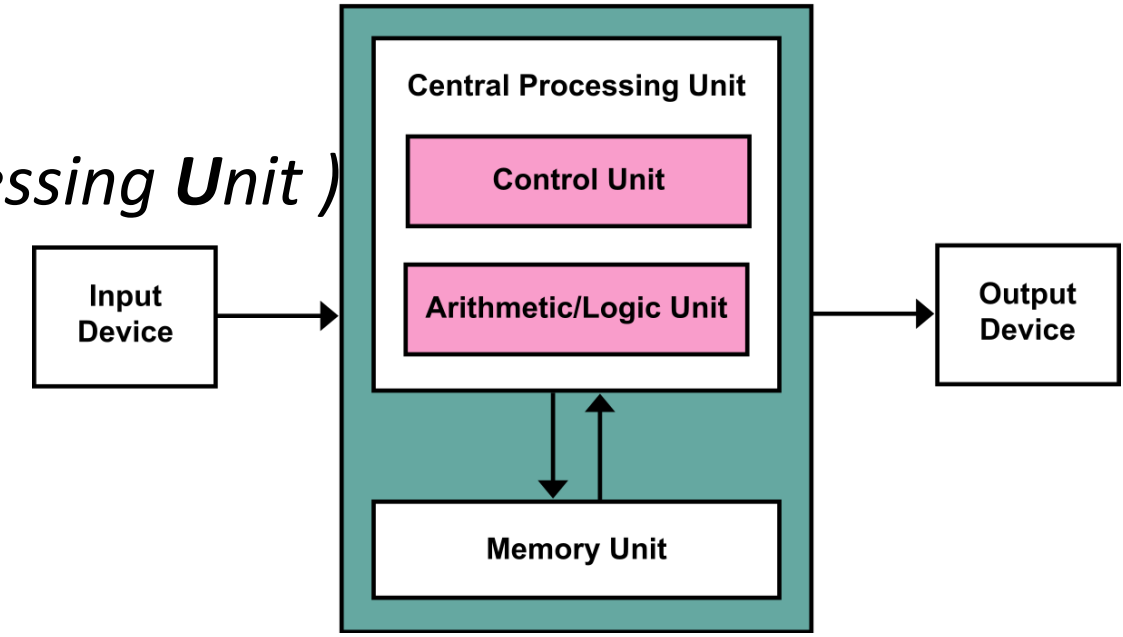
Basic Mechanisms of Program Execution

1. Hardware Structure of a Von Neumann Machine
2. Flow of a Program in a System
3. Concepts of Process and Multiprogramming (Process Context, State, Context Switching Mechanism)

Structure matérielle d'une machine de Von Neumann

- Composition:

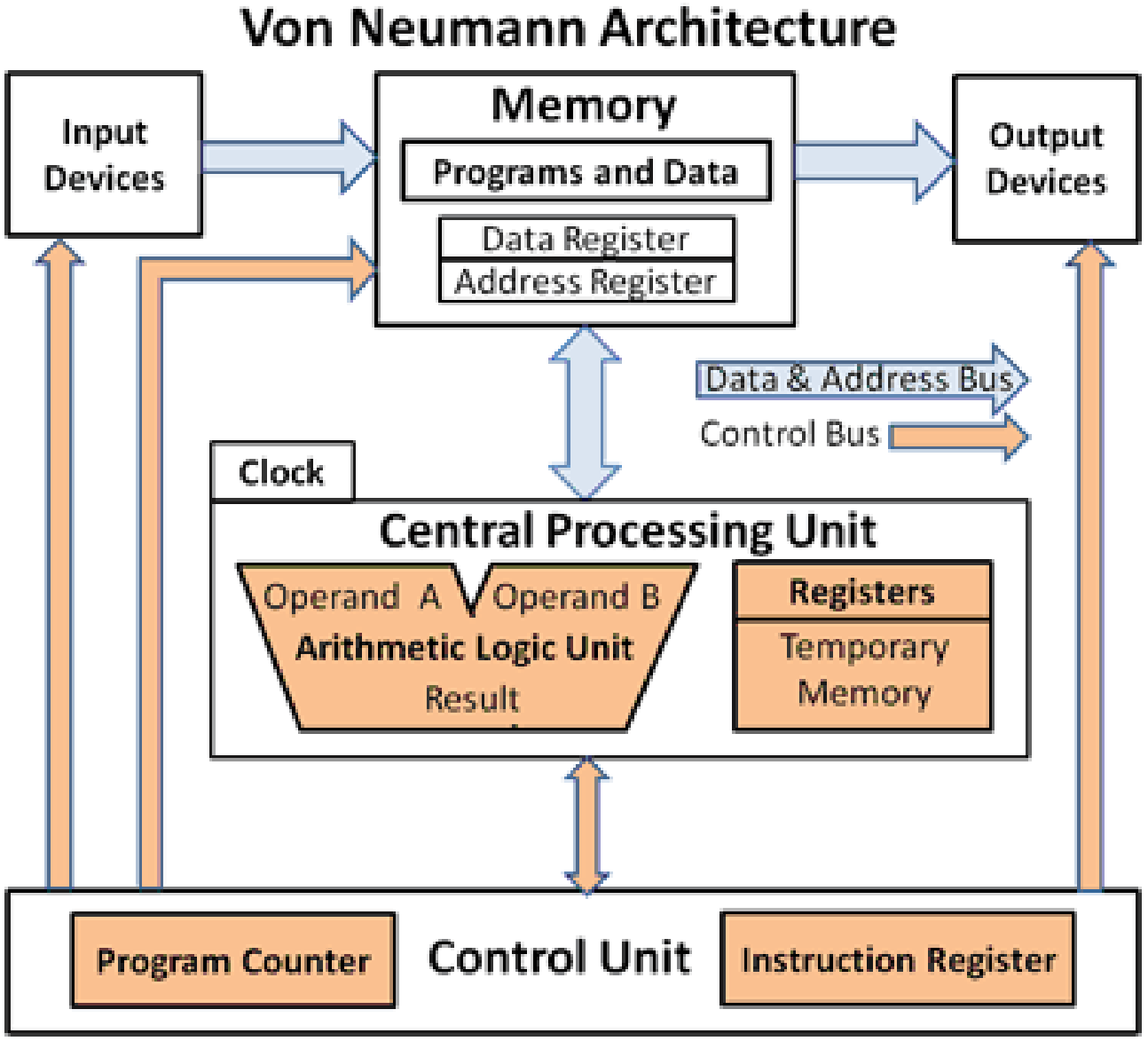
- L'unite centrale UC *(**CPU** : **C**entral **P**rocessing **U**nit)*
- La mémoire centrale (*MC*)
- les périphériques.



- **L'unite arithmétique et logique:** effectue les opérations de calcul en binaire telle que l'addition, la soustraction, le décalage ..etc.
- **L'unite de contrôle:** contrôle le fonctionnement de la machine et exécute les instructions.

Les registres du processeur

Le compteur ordinal (CO) : contient l'adresse mémoire de la prochaine instruction à exécuter.
Le registre instruction (RI) : contient l'instruction en cours d'exécution.
Les registres généraux (accumulateur, registres d'index) : utilisés pour effectuer les calculs et sauvegarder temporairement les résultats,
Le registre pile : pointe ver la tète de la pile du processeur. La pile est une partie de la mémoire centrale utilisé en particulier pour faire des appelles de procédures (stocker les paramètres, retourner les résultats ..etc.)
Le registre d'etat PSW (Processor Status Word): indique l'état du processeur à chaque instant.



Typical Structure of the PSW:

- **Condition Codes (Flags):**

Zero (Z): Set if the result of the last operation is zero.
Sign (S) / Negative (N): Set if the result of the last operation is negative.
Overflow (V): Set if there was an arithmetic overflow in the last operation.
Carry (C): Set if there was a carry or borrow in arithmetic operations.
Parity (P): Set if the number of set bits in the result is even (used in some architectures).

- **Interrupt Control Bits:**

Interrupt Enable/Disable (IE): Controls whether interrupts are allowed.
Priority Level (PL): Defines the priority level of the currently executing process.

- **Supervisor/User Mode Bit:**

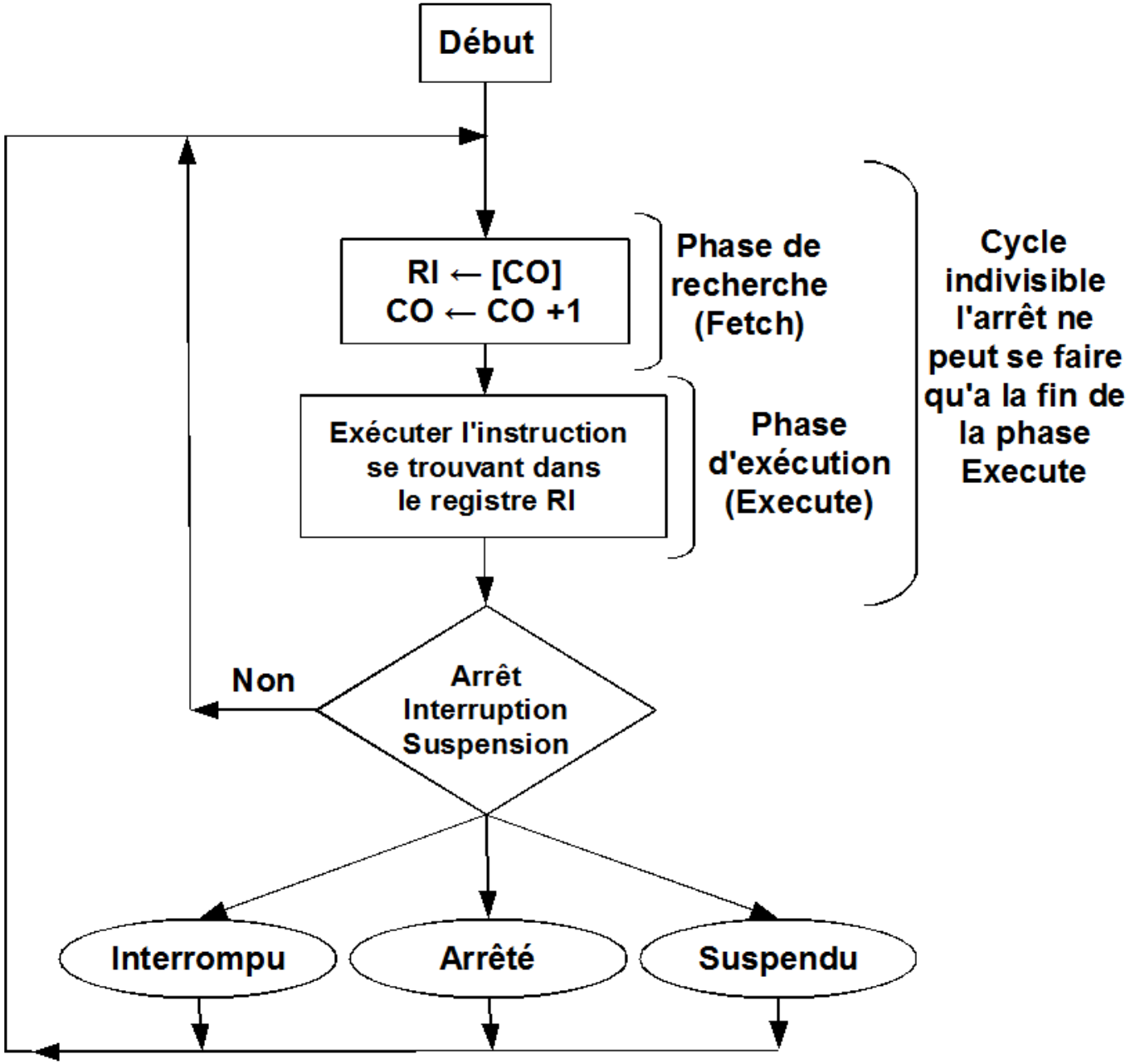
Determines whether the CPU is operating in privileged (kernel/supervisor) mode or user mode.

- **Program Counter (PC) in Some Architectures:**

Some architectures integrate the PC (Program Counter) or part of it within the PSW.

Cycle d’exécution du processeur

- La recherche de l'instruction en mémoire (fetch)
- Le décodage de l'instruction (decode),
- L'exécution de l'instruction (execute),
- Le passage a l'instruction suivante (next) .



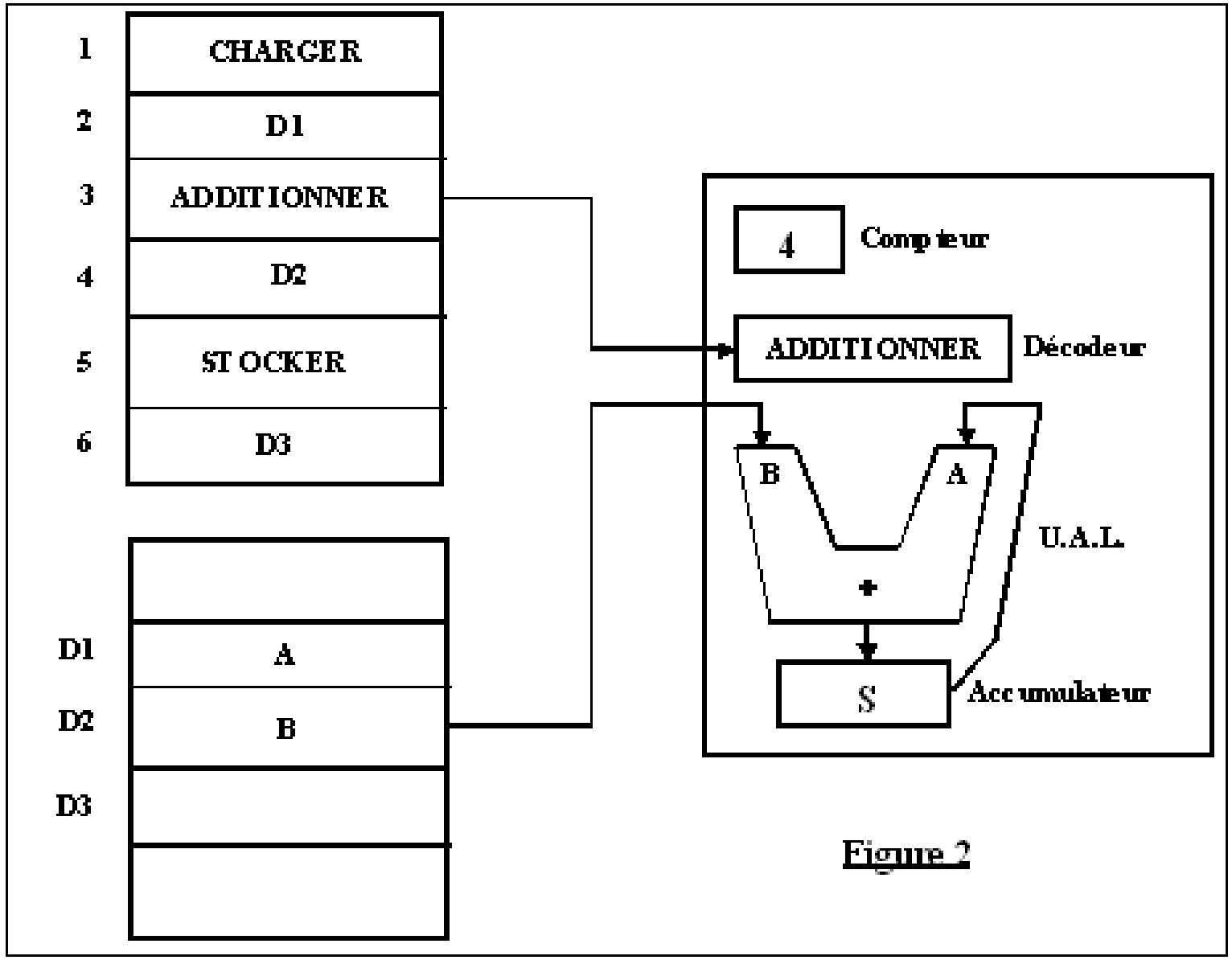
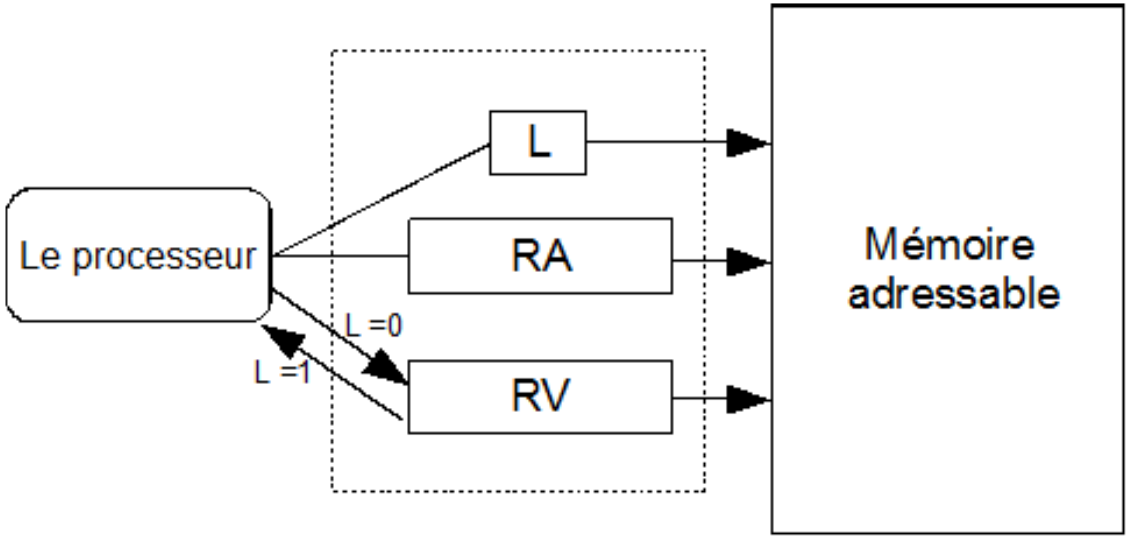

1
CHARGER
2
D1
3
ADDITIONNER
4
D2
5
STOCKER
6
D3
4
Compteur
ADDITIONNER
Décodeur
B
A
U.A.L.
+
S
Accumulateur
D1
A
D2
B
D3

Figure 2

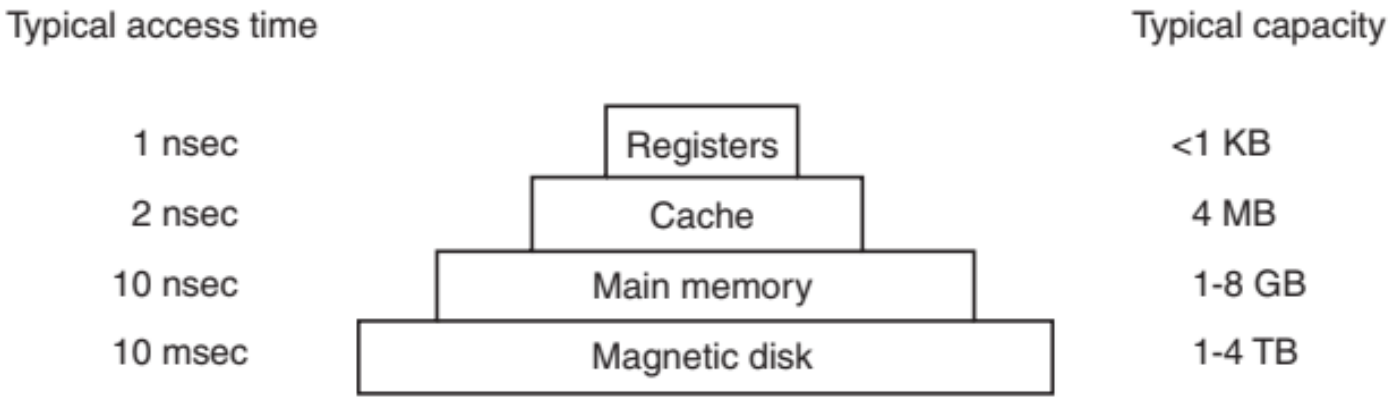
La mémoire centrale

- La mémoire centrale est utilisée pour stocker les instructions et les données des programmes à exécuter.
- Le processeur communique avec la mémoire centrale en utilisant trois registres :

-**RA** (Registre Adresse) : contient l'adresse de l'emplacement mémoire
-**RV** (Registre Valeur) : contient l'information lue ou écrite dans la mémoire
-**L** (Le registre mode de 1 bit) : désigne le mode lecture ou le mode écriture

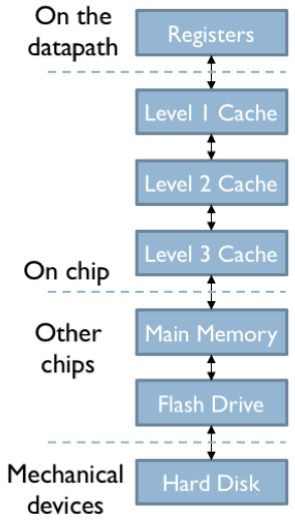


A typical memory hierarchy



A Typical Memory Hierarchy

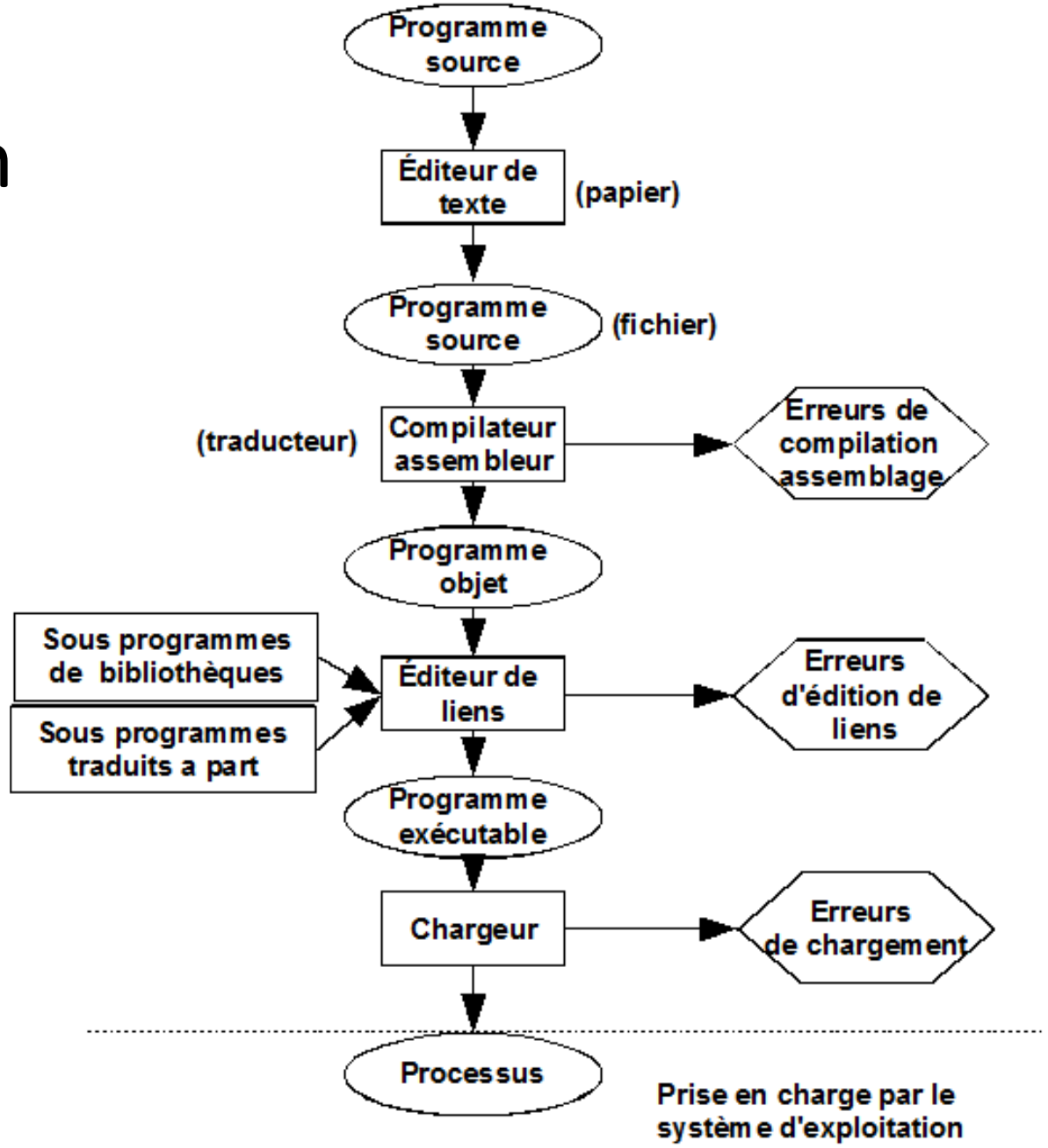
- Everything is a cache for something else…



Access time	Capacity	Managed By
1 cycle	1 KB	Software/Compiler
2-4 cycles	32 KB	Hardware
10 cycles	256 KB	Hardware
40 cycles	10 MB	Hardware
200 cycles	10 GB	Software/OS
10-100us	100 GB	Software/OS
10ms	1 TB	Software/OS

Flow of a Program in a System

- Programs are generally written in a high-level language. For a program to be executed by the machine, it must go through several stages:
  - The text editor
  - The Interpreter
  - The compiler/Assembler
  - The linker
  - The loader.
- These stages are illustrated in the following figure:

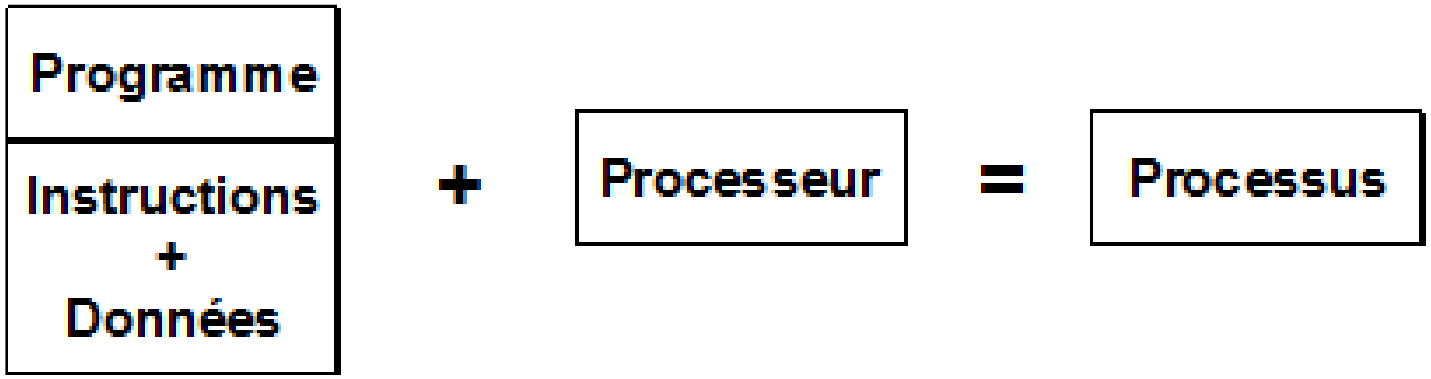


Process

- The main task of operating systems is to manage programs.
- A program running in the system is called a **process**.
- A process consists of executable code and context:
  - **Executable code:** This is the code produced by the linker.
  - **Context:** It describes the execution environment of the program (the program counter, the PSW status word, general registers, etc.).

Process

- A program is not a process; a program is a passive entity, just like any other file stored on a disk, whereas a process is an active entity.
- For example, if two users execute two copies of the same program, there will be two separate processes.



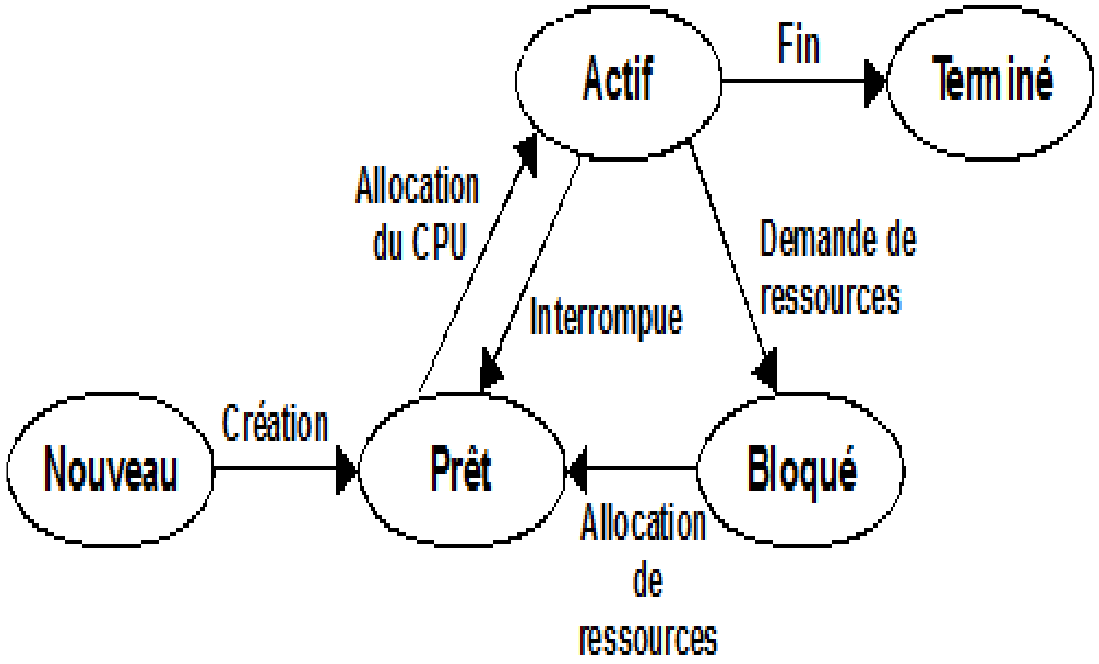
Process

Example:

- Consider a mother preparing a cake using a recipe. She has ingredients like flour, eggs, sugar, etc.
- In this example:
  - The recipe represents the program (a sequence of instructions).
  - The mother represents the processor (CPU).
  - The ingredients are the data provided to the program.
  - The process is the mother's activity of reading the recipe, gathering the necessary ingredients, and baking the cake.

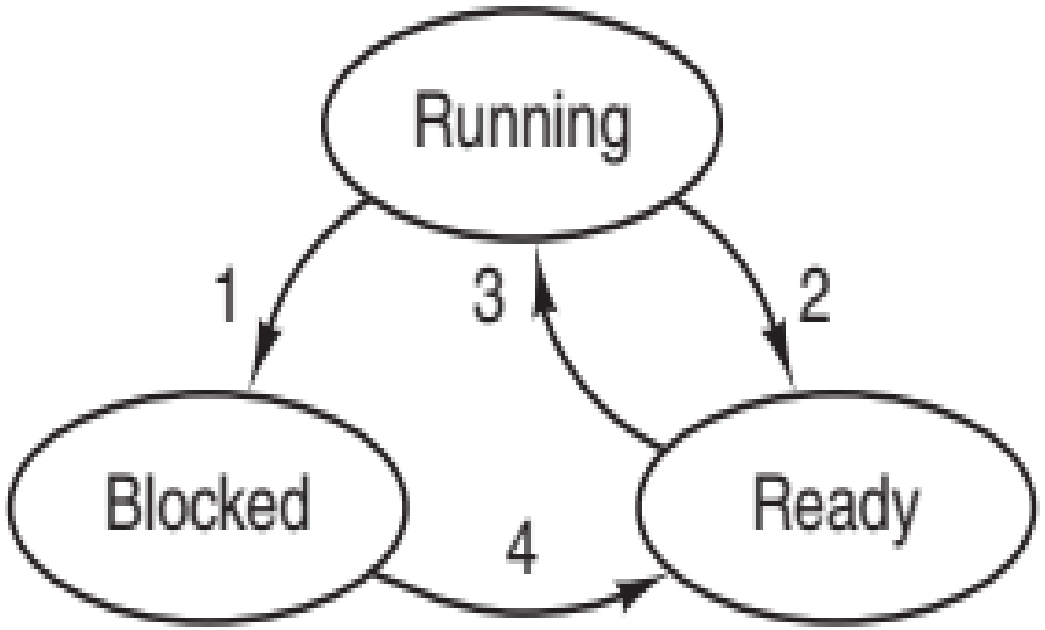
The Different States of a Process

- **Ready**: The process is waiting for the processor to be available in order to execute.
- **Running** (**Active**, **Elected**): The process is currently being executed.
- **Blocked** (**Waiting**): The process is waiting for an event to continue (e.g., completion of an I/O operation, memory allocation, etc.).
- **New**: The process is being created.
- **Terminated**: The process has completed its execution.

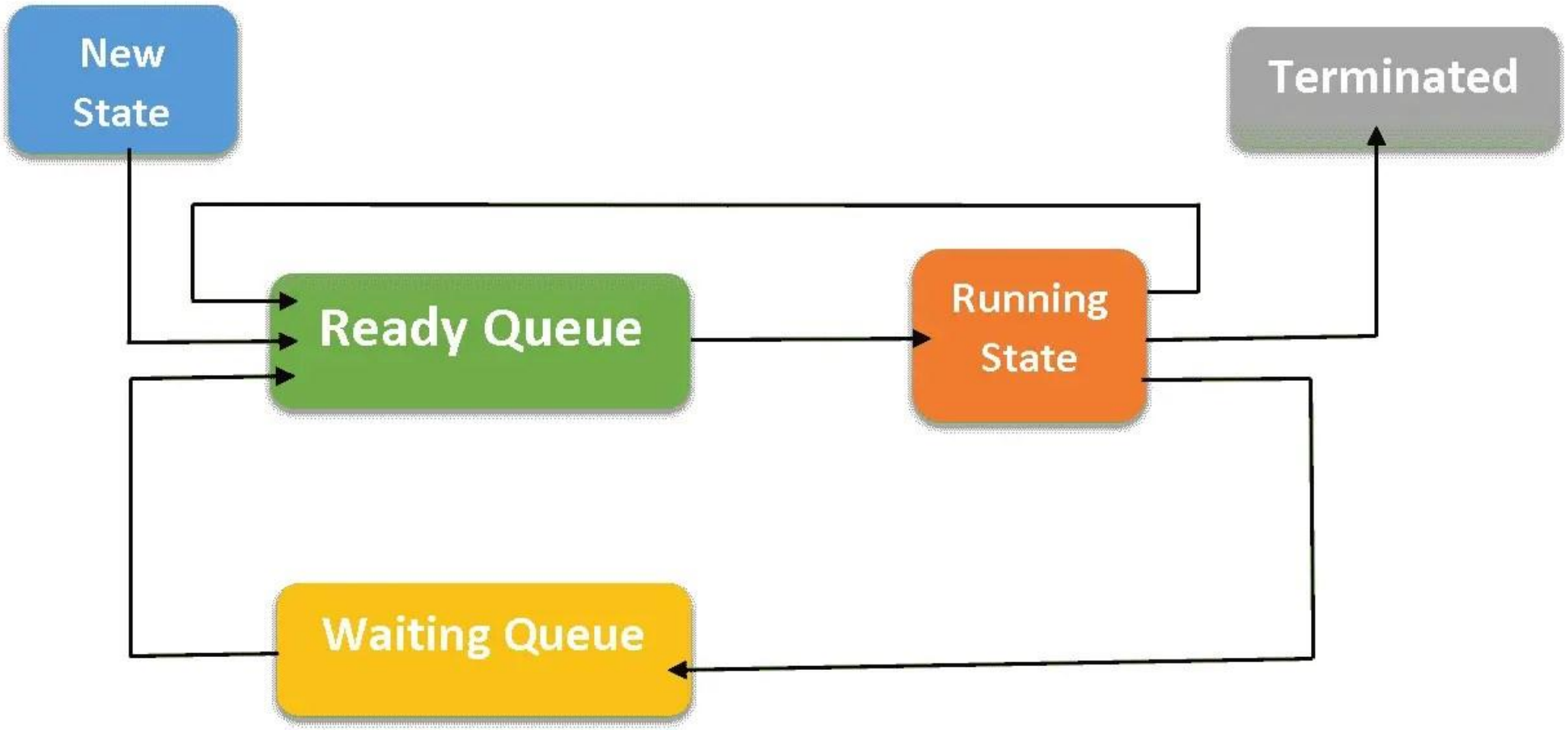


State Transitions of a Process

The Different States of a Process



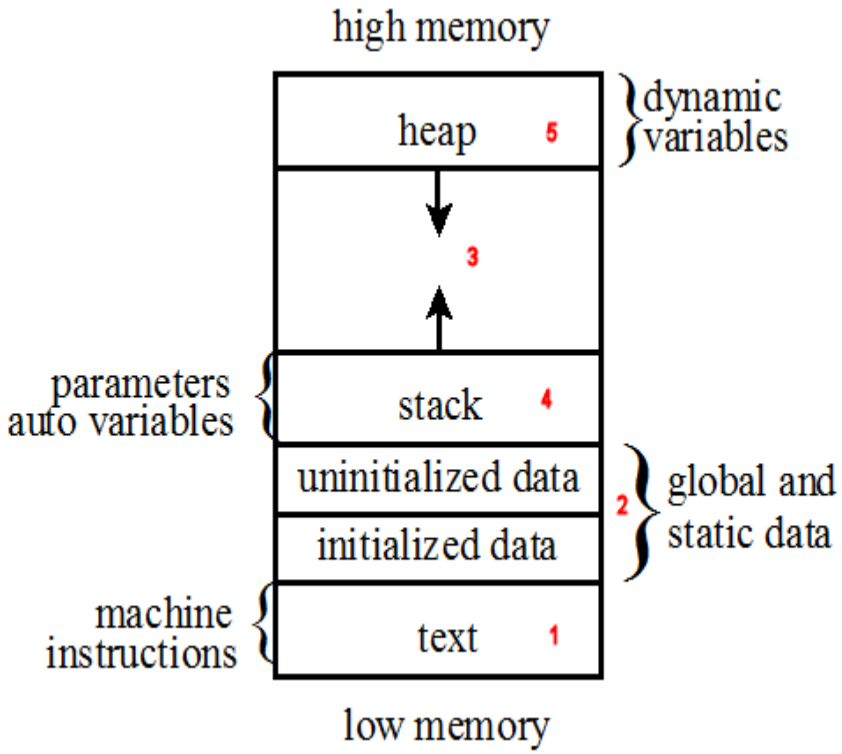
1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available



Memory Map of the process

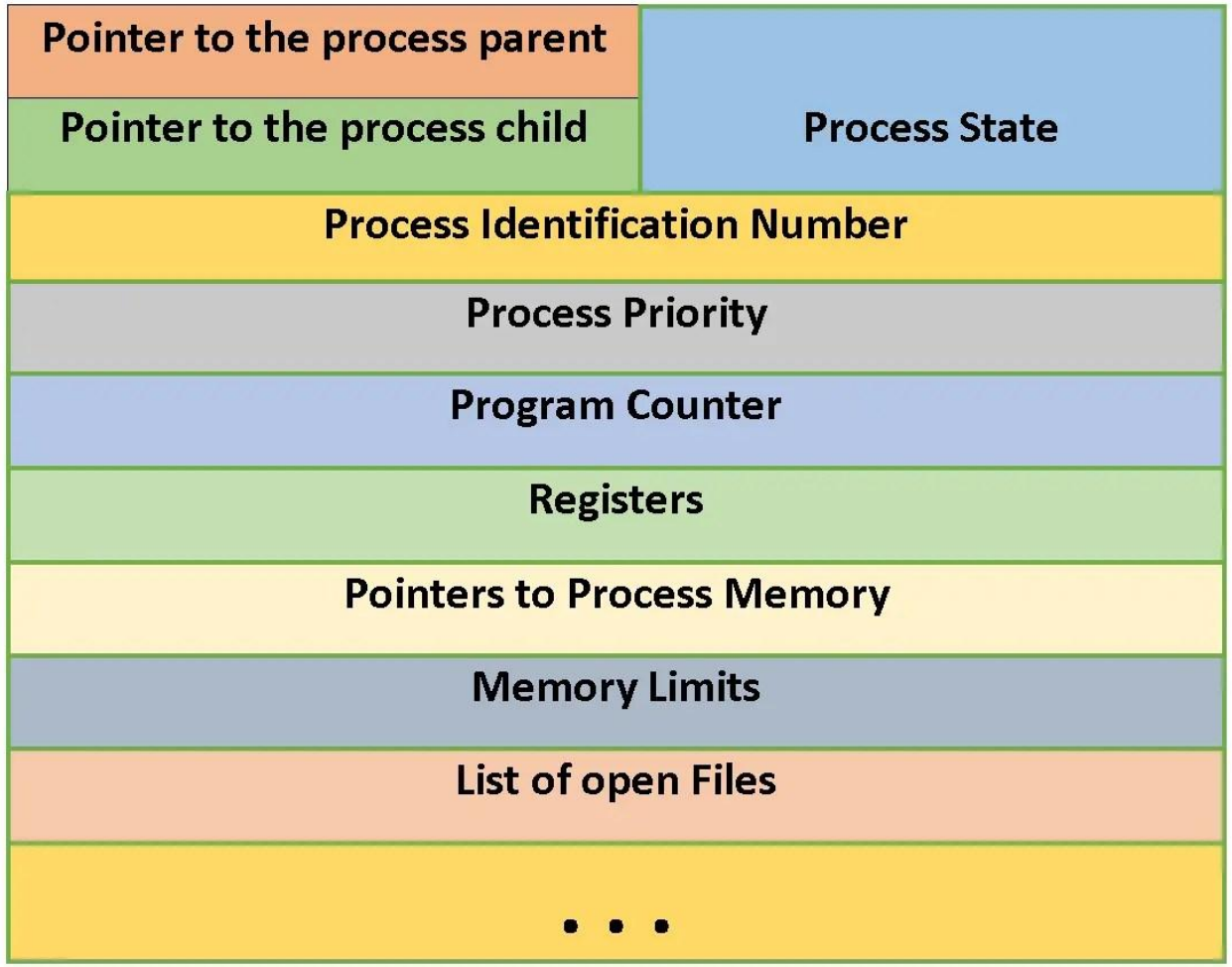
- The memory space allocated to a process is called the **Memory Map** of the process.
- It is divided into several sections:

1) **Code:** Contains the instructions of the program to be executed. Access to this memory area is **read-only**.
2) **Data:** Contains all declared **constants** and **variables**
3)
4) **Stack:** Used to store register values, **local variables**, **function parameters**, and **return addresses** of functions.
5) **Heap:** A dynamically allocated memory area during process execution, using functions such as **new** and **malloc**.



Process Control Blocks (PCB)

- To manage processes, the operating system uses a table in main memory called the **process table** or **Process Control Block (PCB)**.
- This table contains various information about each process, such as the **program counter**, **status words**, **stack pointer**, and **resources used**, ...



PCB

Identité :
- Nom du processus

Gestion du processus :
- Compteur ordinal
- mot d'état
- Registres
- Temps de traitement utilisé
- processus fils
- … etc

Gestion de la mémoire :
- Limites mémoire
- ..etc

Gestion des fichier :
- Répertoire racine
- Fichiers utilisés
- ..etc

Context Switching

- In multiprogramming systems, multiple processes are executed in parallel (pseudoparallelism).
- Switching from one process to another requires saving the context of the stopped process and loading the context of the new process. This operation is called **context switching**.
- The small context includes the Program Counter (PC) and the Processor Status Word (PSW).
- The large context includes the general registers and the stack pointer.

Context Switching