

Centre Universitaire de Mila

2 ème année licence LMD Informatique

Module : Systèmes d'exploitation 1

Bessouf Hakim

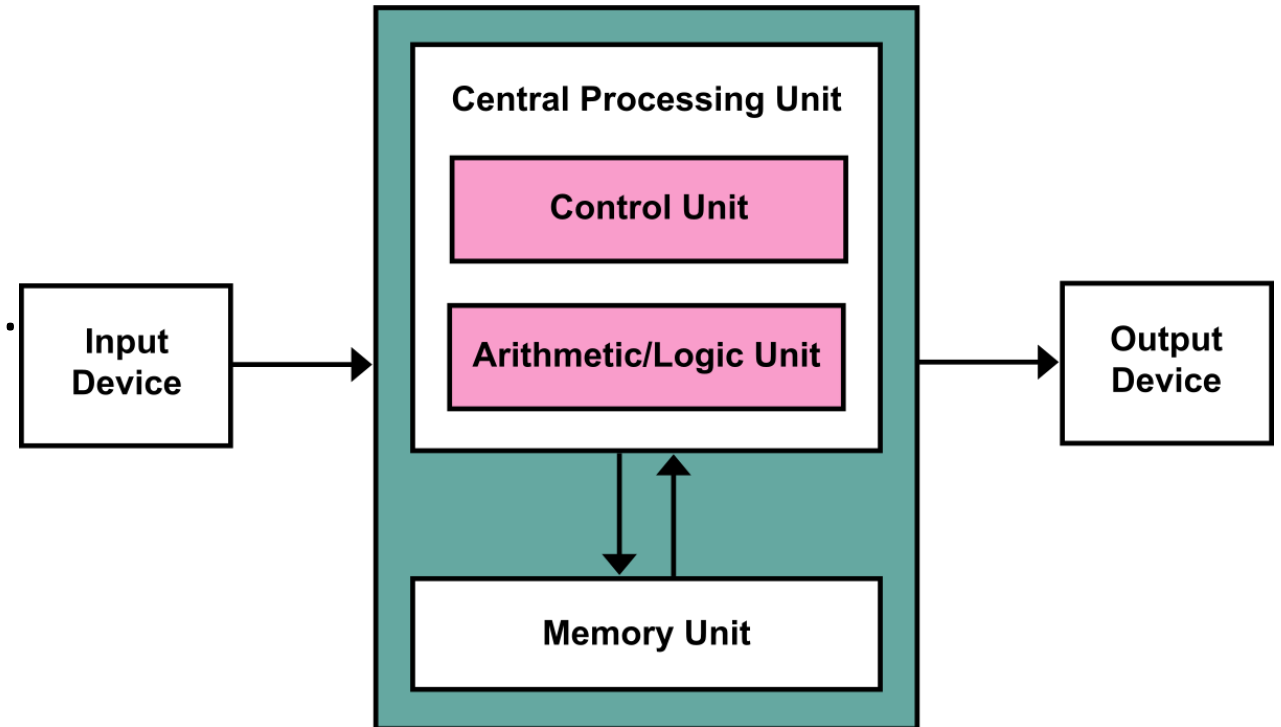
CHAPITRE 2:

Mécanismes de base de l'exécution d'un programme

1. Structure matérielle d'une machine de Von Neumann.
2. Cheminement d'un programme dans un système.
3. Concepts de processus et multiprogrammation (contexte d'un processus, état, mécanisme de commutation de contexte).
4. Les systèmes d'interruption :
 - Définition et organigramme générale d'une interruption.
 - Mécanismes de gestion des interruptions.
 - Systèmes d'interruption sur les PCS.

Structure matérielle d'une machine de Von Neumann

- Une machine de Von Neumann se compose de trois parties : L'unité centrale UC (*CPU : Central Processing Unit*), La mémoire centrale (*MC*) et les périphériques.
- L'unité arithmétique et logique effectue les opérations de calcul en binaire telle que l'addition, la soustraction, le décalage ..etc.
- L'unité de contrôle contrôle le fonctionnement de la machine et exécute les instructions.



Les registres du processeur

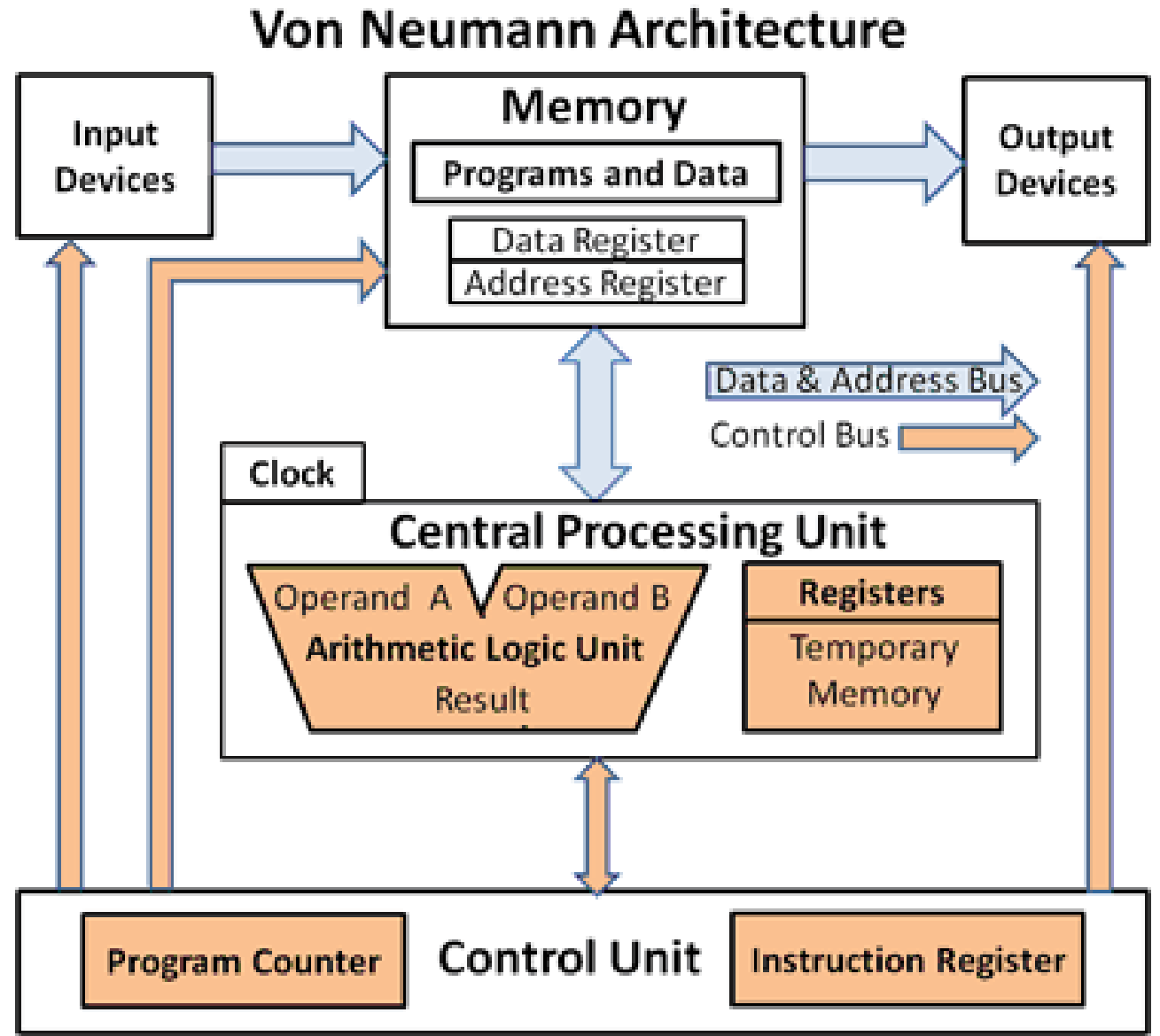
Le compteur ordinal (CO) : il contient l'adresse mémoire de la prochaine instruction à exécuter.

Le registre instruction (RI) : il contient l'instruction en cours d'exécution.

Les registres généraux comme l'accumulateur et les registres d'index : Ils sont utilisés pour faire les calculs et sauvegarder temporairement les résultats,

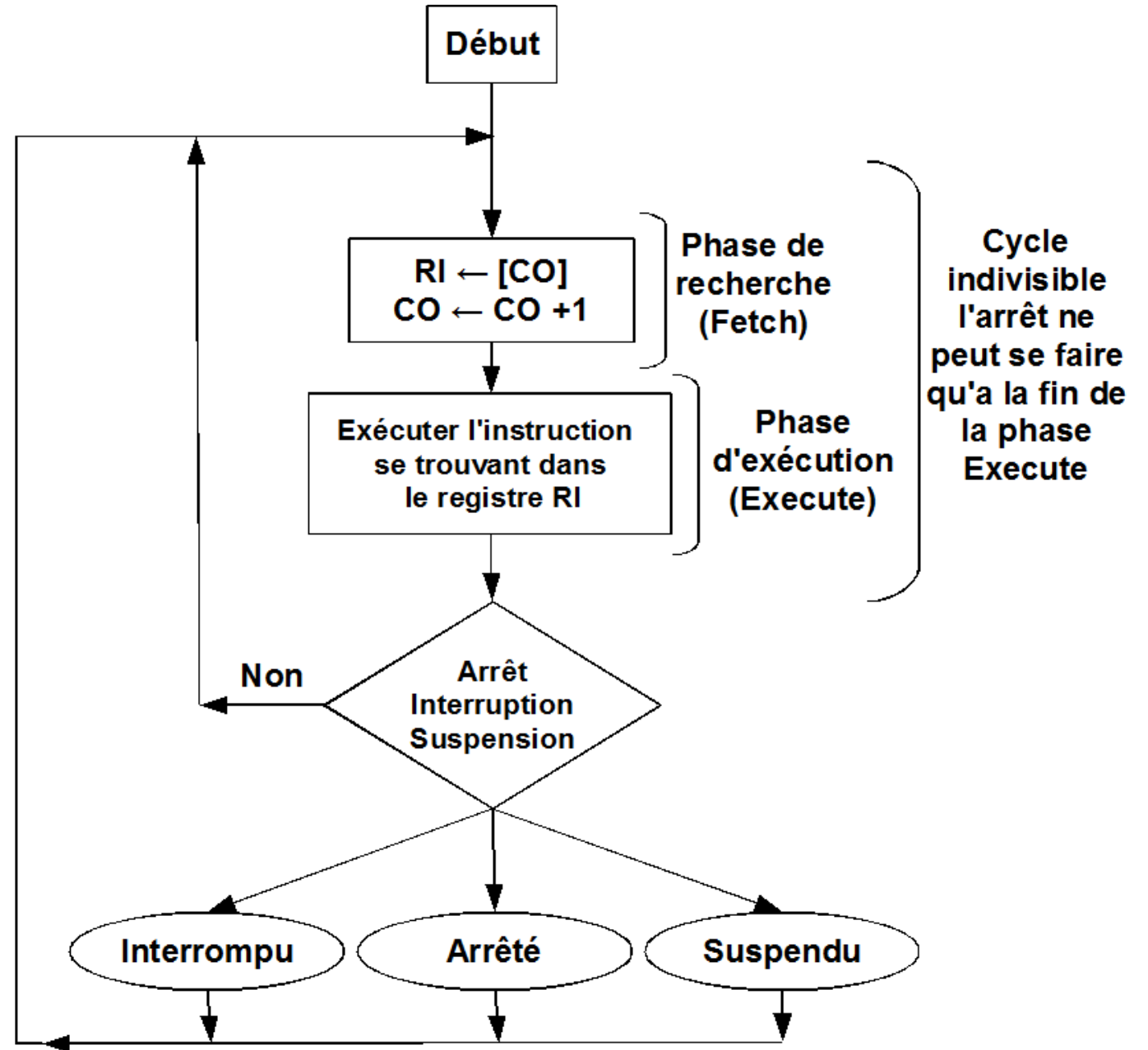
Le registre pile : il pointe vers la tête de la pile du processeur. La pile est une partie de la mémoire centrale utilisée en particulier pour faire des appels de procédures (stocker les paramètres, retourner les résultats ..etc.)

Le registre d'état PSW (Processor Status Word) : Il indique l'état du processeur à chaque instant.



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 à l'instruction suivante (next) .



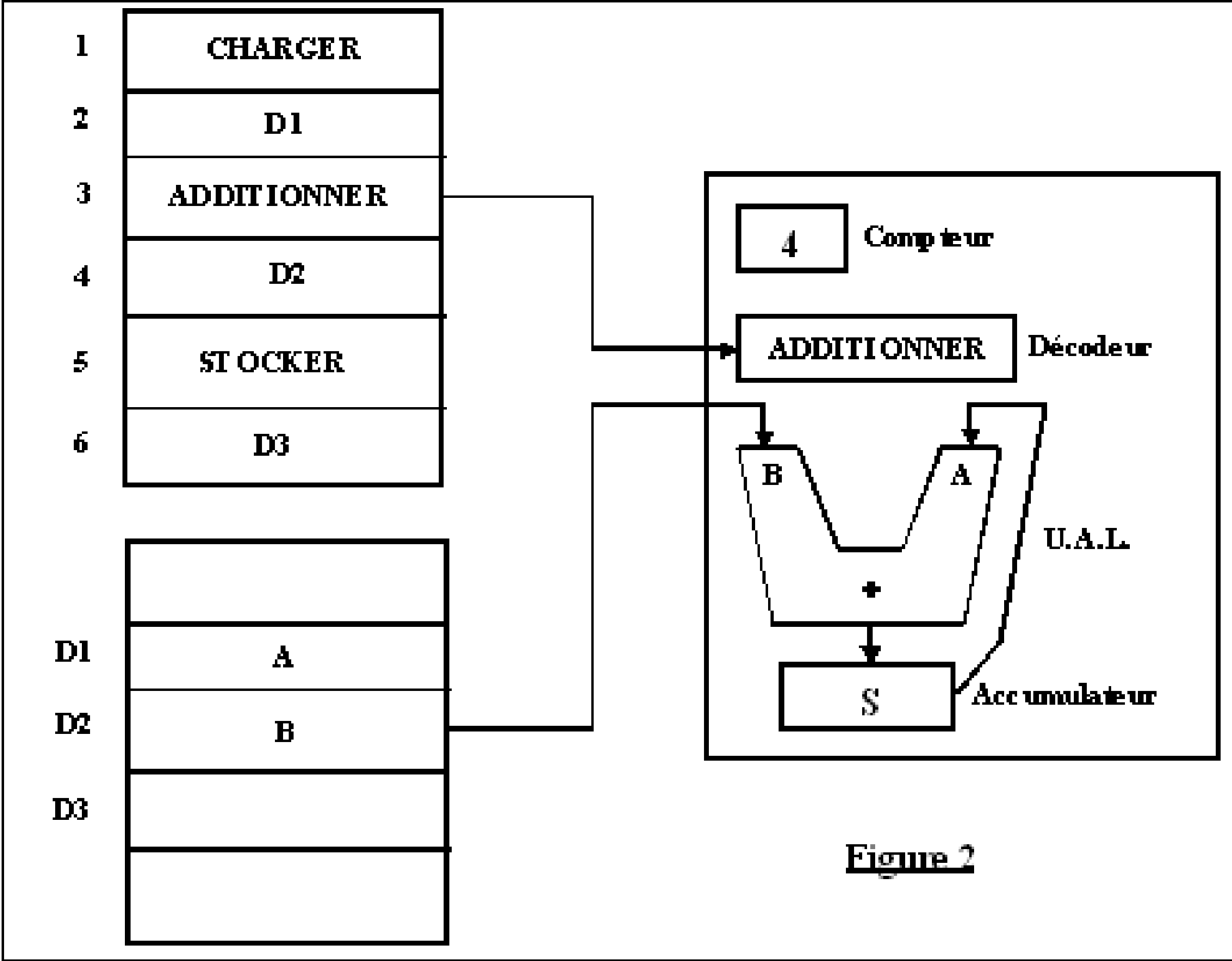
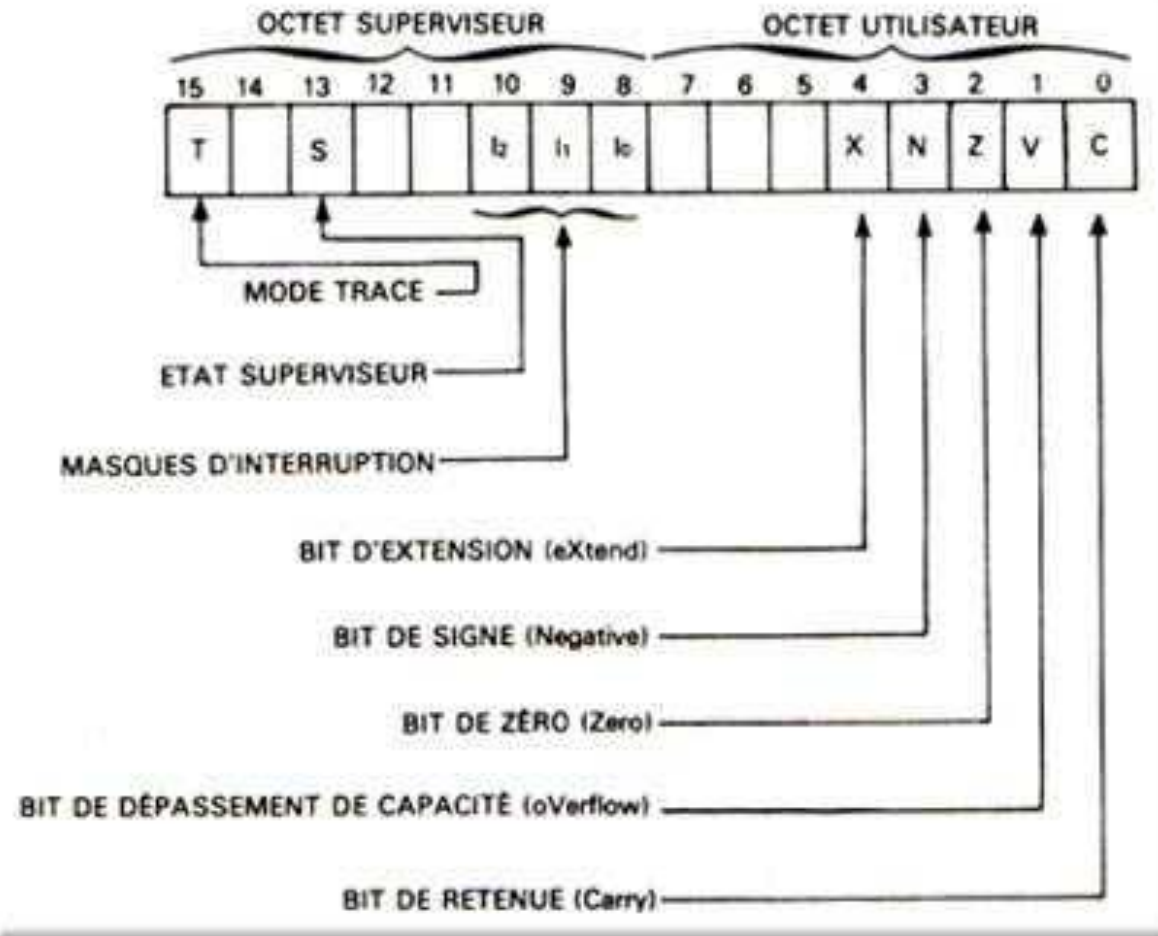


Figure 2

L'état du processeur

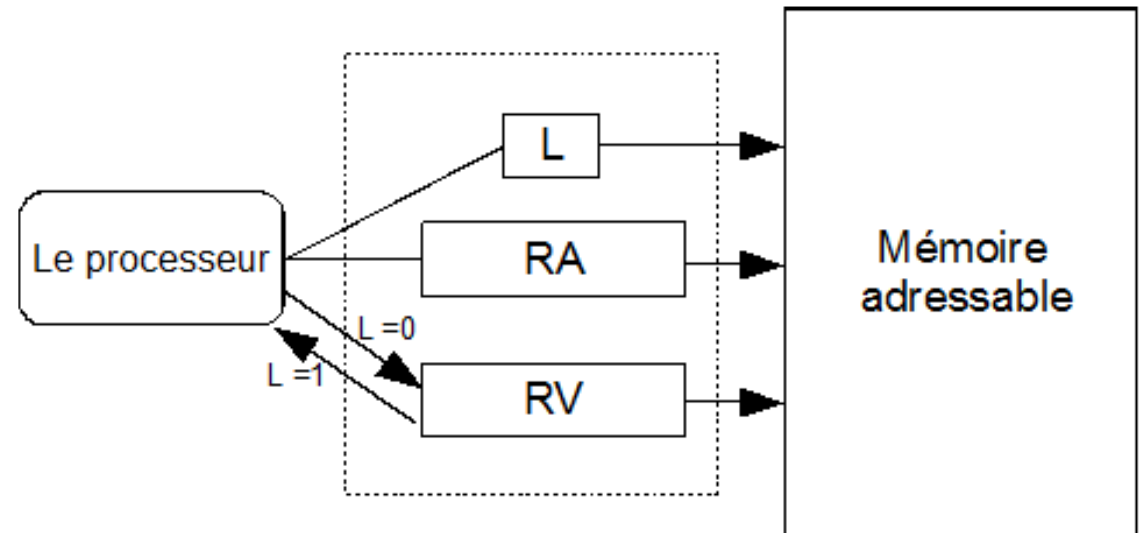
- Le mode d'exécution (noyau ou utilisateur),
- Des informations sur les interruptions qui définissent si d'autres programmes sont plus prioritaires et peuvent interrompre le programme en cours d'exécution.
- Les codes conditions (des bites utilisés dans les opérations arithmétiques et logiques par exemple le bit de retenu, et le bit débordement),
- des indicateurs de protection de la mémoire,
- ... etc.

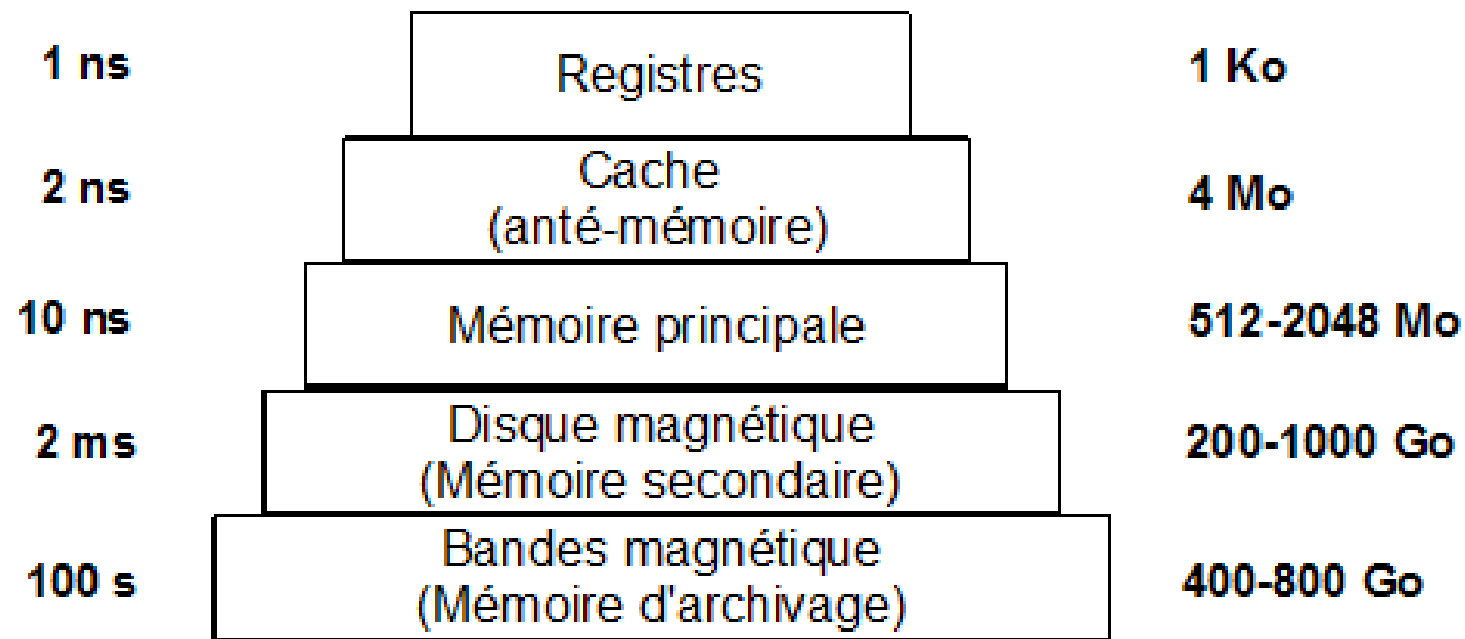


La mémoire centrale

- 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



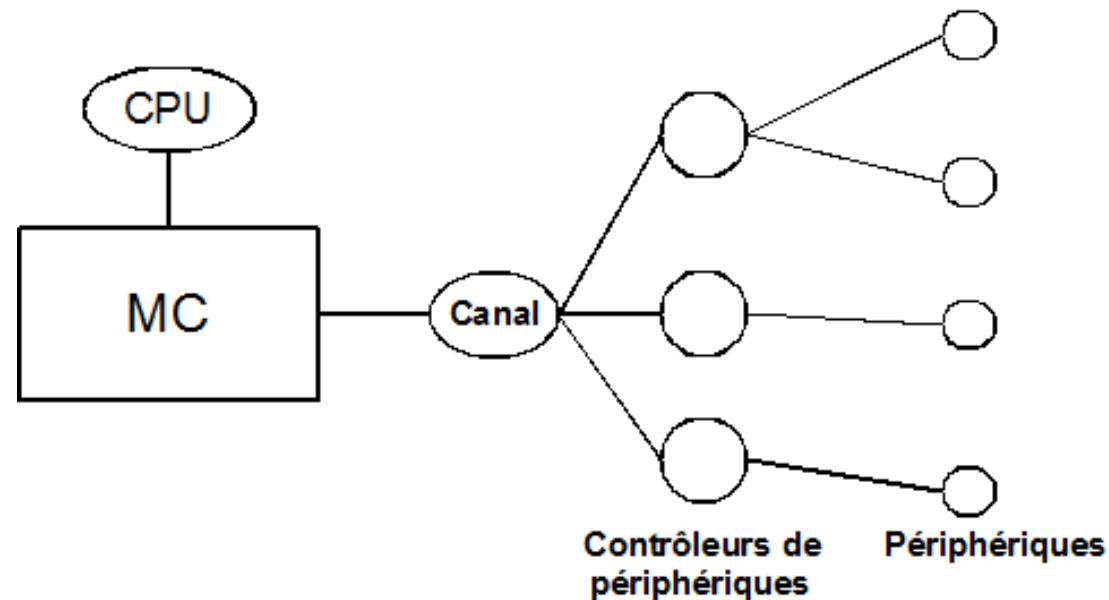


Decoupage hierarchique de la memoire.

Les périphériques d'E/S

Le processeur (CPU) échange les informations (données en entrée et résultats en sortie) avec l'extérieur par l'intermédiaire des dispositifs appelés Contrôleurs de périphériques et canaux d'E/S.

L'organisation la plus courante est illustre dans la figure suivante :

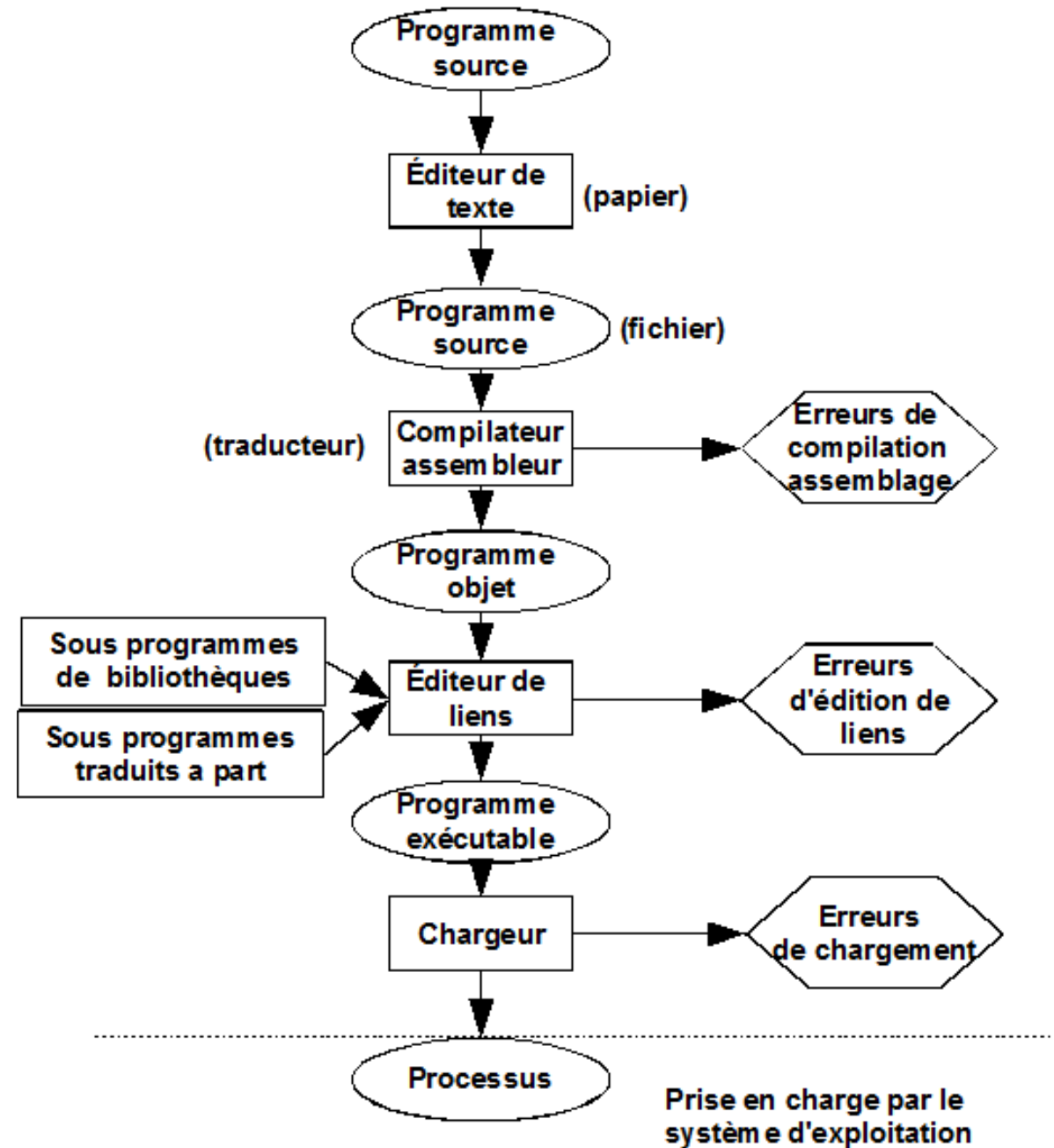


Les peripherique d'E/S

- **Le Canal (processeur d'E/S)** C'est un processeur spécialisé dans les opérations d'E/S. Il ne peut être lancé ou arrêté que par le processeur central. Après qu'il soit lancé Il peut gérer les E/S de un ou plusieurs périphériques sans l'intervention du processeur centrale.
- **Le Contrôleur de périphériques (coupleur)** C'est un dispositif de commande adapté à un type de périphériques qui permet de connecter plusieurs périphériques (de même type). Comme chaque contrôleur est spécifique, un programme spécifique est nécessaire pour contrôler chacun d'eux. Ce programme qui communique avec le contrôleur est appelé **pilote de périphériques** et ils est placé dans le système d'exploitation pour être exécuté en mode noyau.
- **Le périphérique** C'est un organe capable de transférer de l'information.

Cheminement d'un programme dans un système

- Les programmes sont en générale écrit en langage évolué. Pour qu'un programme soit exécuté par la machine il doit passer par plusieurs étapes: l'éditeur de texte, le traducteur, le compilateur/assembleur, l'éditeur de liens et le chargeur. Ces étapes sont illustrées dans la figure suivante :



Le processus

- La tâche principale des systèmes d'exploitation est de gérer les programmes. Un **programme en exécution** dans le système est appelé **processus**.
- Un processus est composé du code exécutable et du contexte :
 - **Le code exécutable** : C'est le code produit par l'éditeur de liens.
 - **Le contexte** : Il décrit l'environnement du programme en exécution (le compteur ordinal , le mots d'état PSW, les registres généraux, ..etc.)

Le processus

- Un programme n'est pas un processus, le programme est une entité **passive** comme n'importe quel autre fichier stocké sur disque tandis que le processus est une entité **active**.
- Si par exemple deux utilisateurs exécutent deux copies du même programme on aura deux processus séparés.

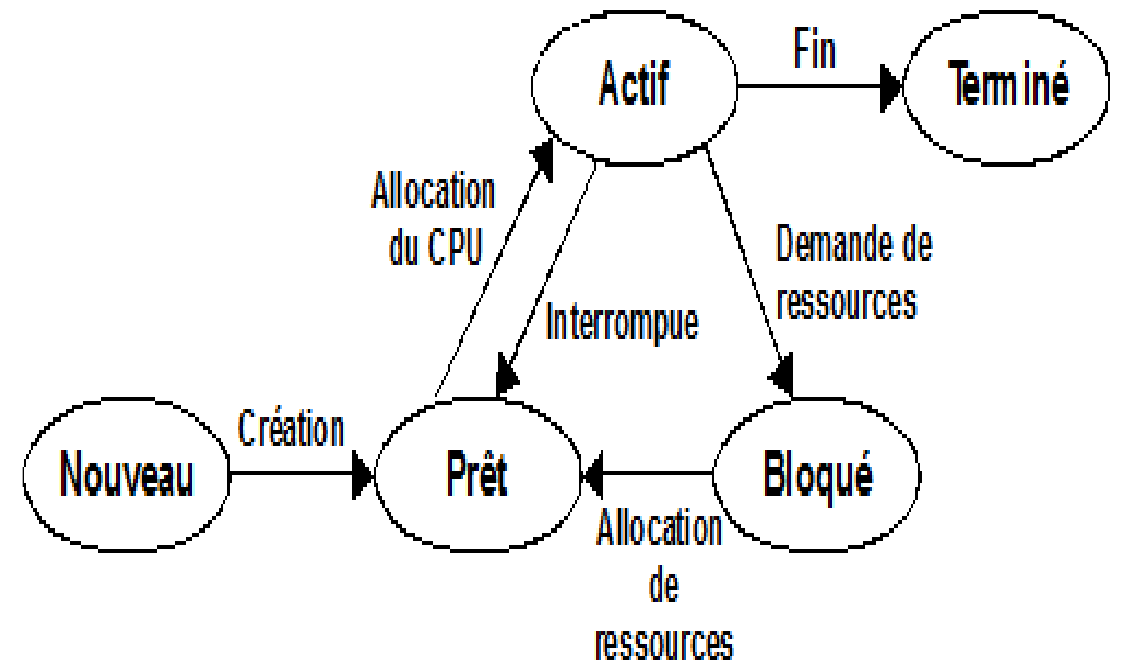


Le processus

- **Exemple :**
- Considérons la maman qui prépare un gâteau dont elle a une recette et, dispose dans sa cuisine de farine, œufs, sucre etc.
- Dans cette exemple la **recette** représente le **programme** (une suite d'instructions), la **maman** représente le **processeur** (CPU) et les ingrédients sont les données à fournir au programme. Le **processus** est l'**activité de la maman** qui lie la recette, trouve les ingrédients nécessaires et fait cuire le gâteau.

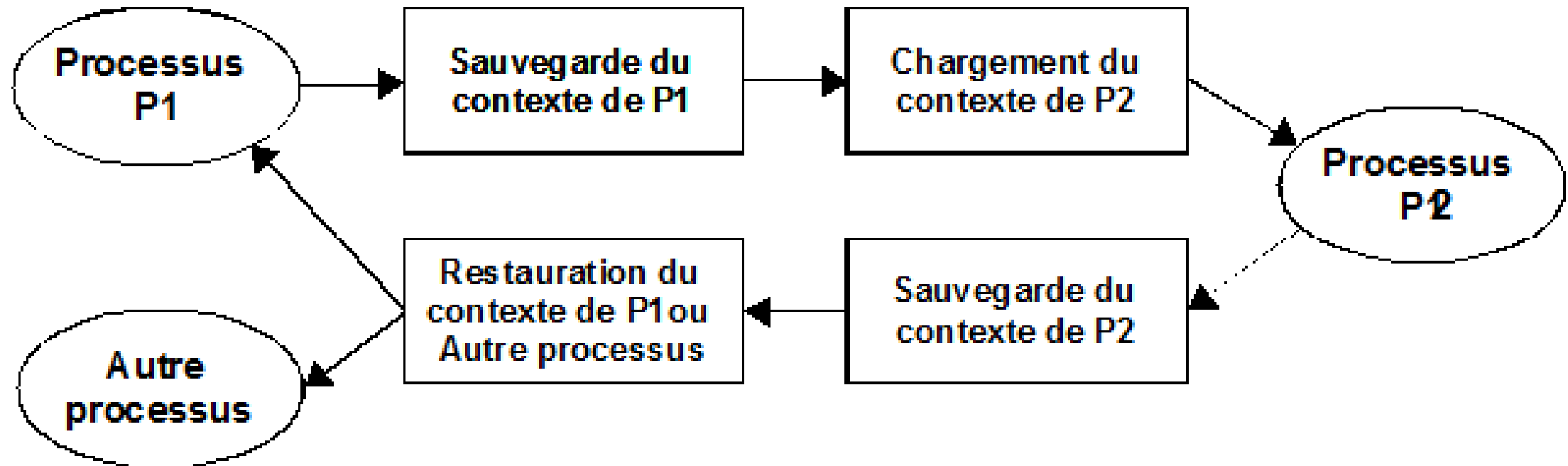
Les différents états du processus

- **Prêt** : le processus attend la libération du processeur pour s'exécuter.
- **Actif** (en exécution, élu) : Le processus est en cours d'exécution.
- **Bloqué** (en attente) : le processus attend un évènement pour pouvoir continuer (exemple : fin d'une E/S, allocation de mémoire .. etc.)
- **Nouveau** : le processus est en cours de création
- **Terminé** : le processus a terminé son exécution.



Les transitions d'états d'un processus.

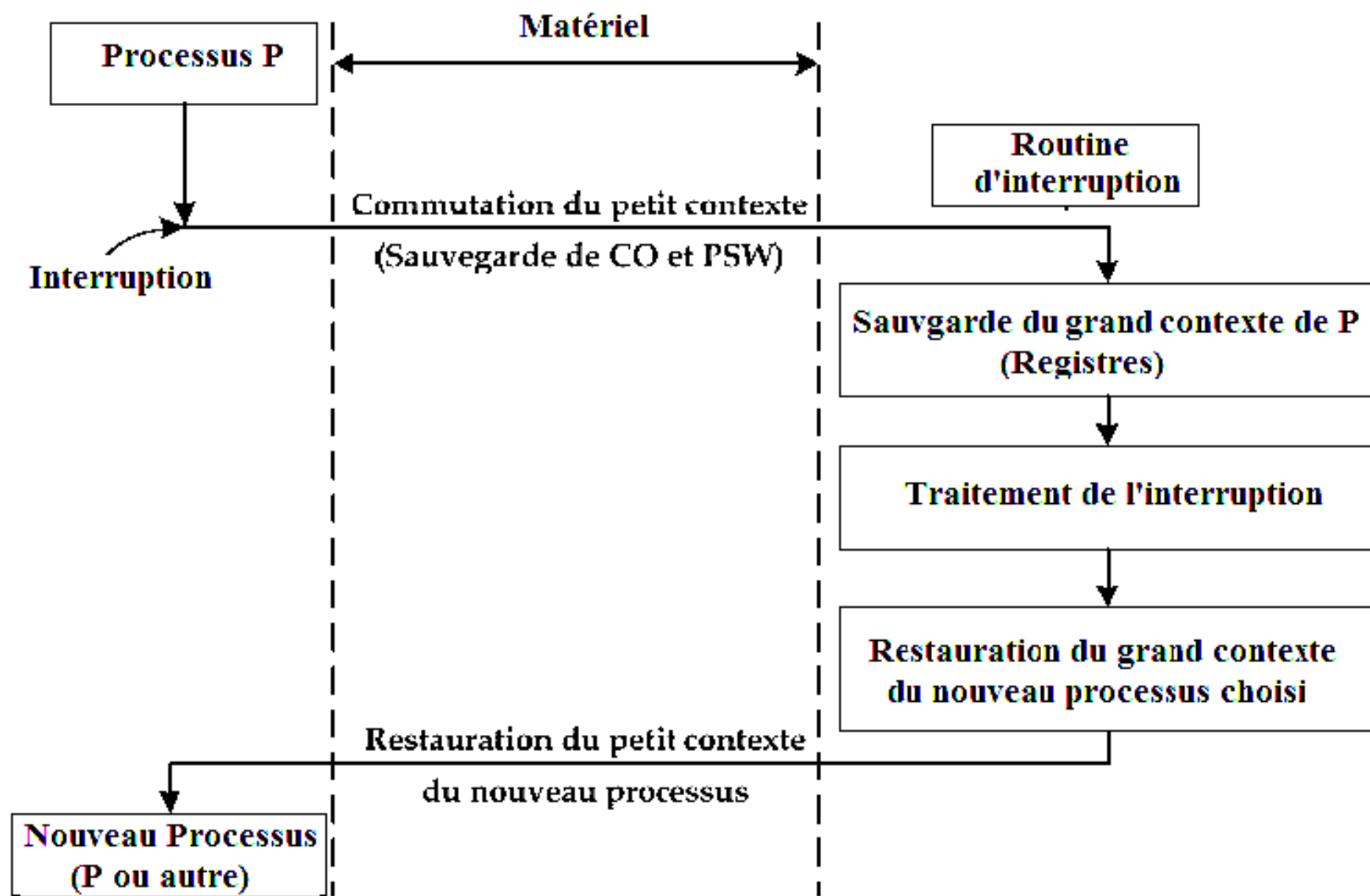
La commutation de contexte



La commutation de contexte

La commutation de contexte

- Dans les systèmes multiprogrammes plusieurs processus sont exécutés en parallèle (**pseudo-parallélisme**).
- Le passage d'un processus à un autre nécessite une opération de **sauvegarde du contexte** du processus arrêté et de **chargement du contexte** du nouveau processus. Cette opération est appelée **commutation de contexte**.
- Le **petit contexte** comprend le compteur ordinal (CO) et le mot d'état (PSW),
- Le **grand contexte** comprend les registres généraux et la pile.



Les systèmes d'interruption

- Une **interruption** est un signal envoyé au processeur qui provoque l'interruption de l'exécution du programme en cours et l'exécution d'un autre programme spécifique à l'interruption appelé **Routine d'interruption**.
- Une interruption ne peut être traitée que si le processeur est dans un **état observable** (il n'est pas entrain d'exécuter une instruction).
- Avant d'exécuter une routine d'interruption le processeur doit **sauvegarder le contexte** du programme en exécution et charger le contexte de la routine d'interruption.

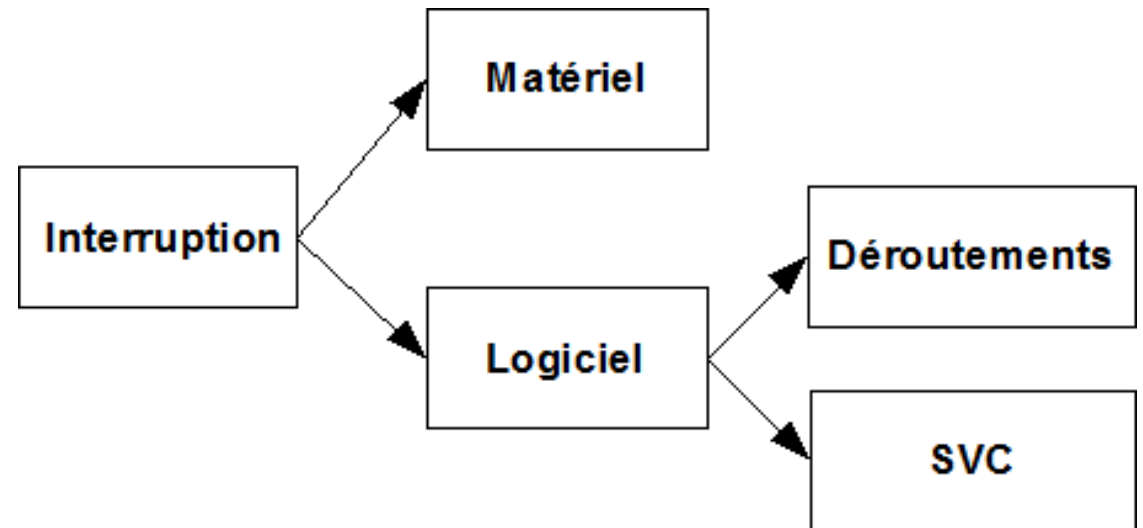
Types d'interruptions

- Les **interruptions externes (matériels)**

sont causes par des composants externes au processeur (horloge, périphériques .. etc).

- Les **interruptions internes (logiciels)**

sont causes par les programmes en exécution, ils se divisent en deux parties : les **déroutements** (trap), et les **appelles au superviseur**.



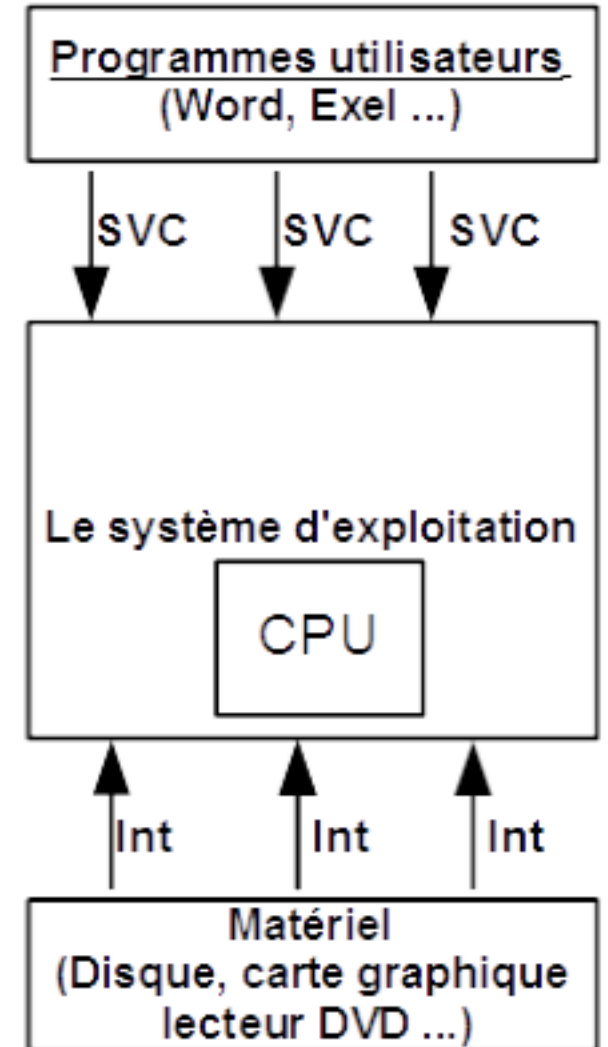
Types d'interruptions.

Types d'interruptions

- Les **déroutements** sont dus à des erreurs lors de l'exécution des programmes comme :
 - Division par 0,
 - Débordement arithmétique,
 - Exécution d'une instruction interdite,
 - Accès à une zone mémoire protégée ou inexistante ..etc.
- **L'appel au superviseur** est une instruction qui permet d'utiliser une fonction du système d'exploitation par un programme utilisateur.
Cette instruction change le mode d'exécution du processeur en **mode noyau**.

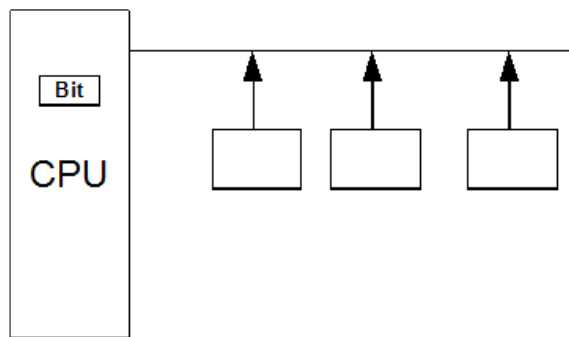
Types d'interruptions

- Les programmes communiquent avec le système en utilisant les SVC.
- Le matérielle communique avec les système en utilisant les interruptions matériels.

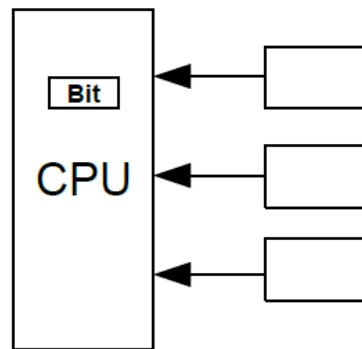


Mécanisme d'interruption

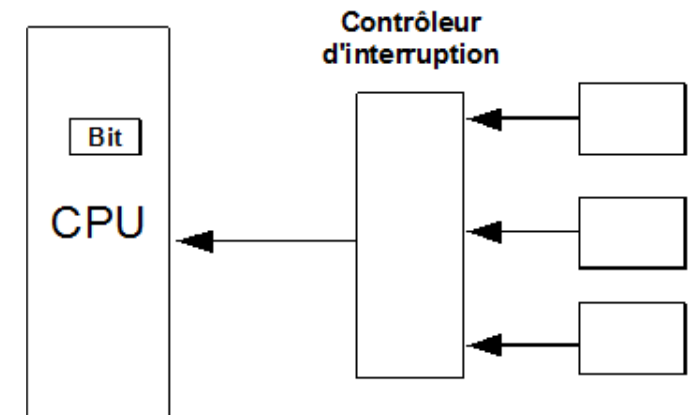
- Une interruption correspond physiquement a un signale électrique qui positionne un bit au niveau du processeur appelé **indicateur d'interruption**. Le processeur central consulte ce bit a chaque état observable pour vérifier si il y a une interruption.



Une seule ligne d'interruption



Plusieurs lignes d'interruption



Avec contrôleur d'interruption

Mécanisme d'interruption

- Deux schémas sont possibles pour la distinction des interruptions:
 - Un indicateur unique est utilisé pour toutes les interruptions; (on utilise donc une seule ligne d'information). Dans ce cas une routine d'interruption unique est exécutée, on a besoin donc d'un seul emplacement pour sauvegarder le contexte de la routine.
 - Un indicateur distinct est associé à chaque cause d'interruption (plusieurs lignes d'interruption) on parle de **Niveaux d'interruption**. Dans ce cas une routine d'interruption est associée à chaque niveau et on a besoin de plusieurs emplacements pour sauvegarder les contextes des routines d'interruption .

Identification de la cause d'interruption

- **Identification par scrutation**

Une seule routine d'interruption est exécutée pour toute les interruption . Elle vérifie les lignes d'interruption (si il y a plusieurs lignes), ou les registres d'état des différent composant (si il y a une seul ligne d'interruption) et exécute la procédure correspondante. Cette méthode **est lente** parce qu'elle vérifie chaque composant.

- **Identification par vecteur**

Dans ce cas c'est le composant qui a provoqué l'interruption qui s'identifie en envoyant un code (**code d'interruption**) au processeur. Le processeur utilise ce code pour choisir la procédure à exécuter. Dans cette méthode il faut utiliser un mécanisme pour empêcher deux composants d'envoyer le code d'interruption en même temps.

Priorités des interruptions

- Des fois plusieurs interruptions peuvent arrivés en même temps
- il faut donc régler le conflit en établissant un ordre de priorité.
- Cet ordre peut être fixé une fois pour toute ou modifiable (programmable).

Masquage des interruptions

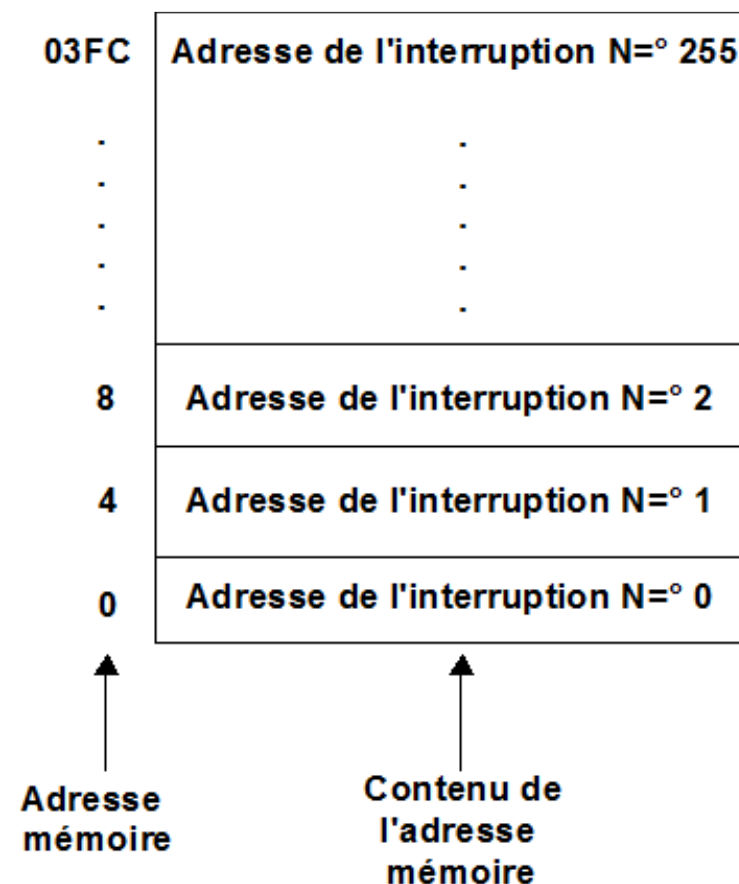
- Parfois il est utile de protéger des programmes en exécution contre certaines causes d'interruption
- Exemple: protéger une interruption contre une autre interruption
- Une interruption **masqué** ne peut pas interrompre le processeur, elle est retardé (mémorisé) jusqu'a ce quelle soit **démasqué**.

Activation des interruptions

- Parfois le processeur a besoin d'interdire toutes les interruptions possibles, toutes les interruption sont donc **Activé** ou **Désactivé**.
- Si les système d'interruption est désactivé, toutes les interruption sont retardé.

Systeme d'interruption dans les PC

- Un PC comprend 256 niveaux d'interruption numérotés de 0 a 255. un tableau appelé **vecteur d'interruption** permet de faire la correspondance entre le numéro de l'interruption et l'adresse de la **routine d'interruption** en mémoire central.
- Chaque adresse de routine d'interruption occupe 4 mots mémoire. Pour trouver l'adresse d'une routine d'interruption on multiplie le numéro d'interruption par quatre. Si par exemple le numéro de l'interruption est 5, l'adresse de la routine d'interruption se trouve a partir de l'adresse mémoire $5 \times 4 = 20 = 14H$. (H veut dire que le nombre est en hexadécimale).



- Chaque niveau d'interruption correspond a une routine d'interruption qui a la forme suivante :

Début

Sauvegarder les registres généraux

Traiter la cause de l'interruption

Restaurer les registre généraux

Reprendre l'exécution du programme interrompu

Fin

Les déroutements

- Les déroutements sont provoqués par les programmes en exécution, par exemple :
 - La division par 0 (interruption N=° 0h)
 - Problème d'accès mémoire (interruption N=° 2h)
 - Débordement arithmétique (interruption N=° 4h)
 - Instruction inconnu (interruption N=° 6h)
- En cas de déroutement le numéro de l'interruption est généré automatiquement par le processeurs.

Les appels système (SVC)

- Pour faire des appels système les programmeurs utilisent l'instruction **INT**. Cette instruction permet d'indiquer au processeur l'interruption à exécuter. Par exemple l'instruction **INT 9h** permet d'exécuter l'interruption N= 9h.

- **Exemple** : Un programme C qui lit une variable au clavier doit utiliser l'interruption 21h.

Programme C

```
int main()  
{  
.  
.  
scanf("%d", &x);  
.  
}
```

Programme compile

```
.  
.  
.  
Mov ah, 01h  
INT 21h  
.  
.
```

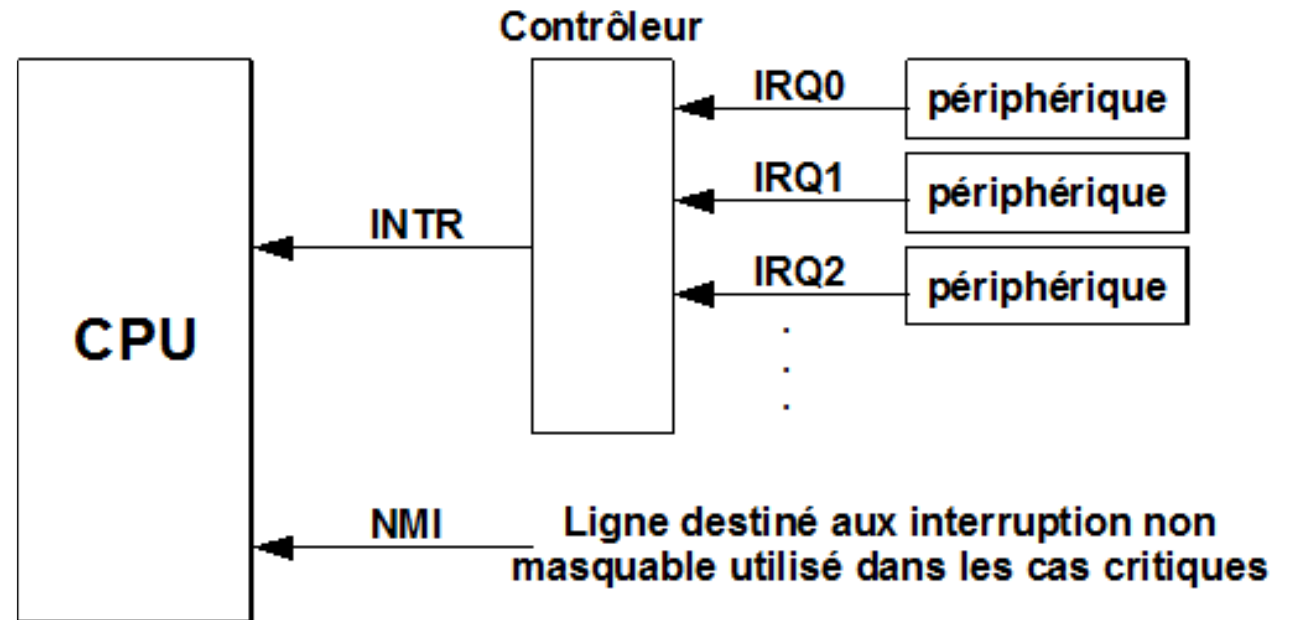
- L'appel système INT 21h est automatiquement inséré par le compilateur. L'interruption 21h permet d'accéder à une fonction système qui permet de lire des caractères du clavier.

Exemple d'appels Système

- Lire la date et l'heur : INT 1Ah
- Fonctions du dos : INT 21h
- Fonctions vidéo : INT 10h
- Lecture disquette : INT 25h
- Accès imprimante parallèle : INT 17h
- ...etc.

Les interruptions matériels

- Les PC équipés d'un microprocesseur Intel utilisent un contrôleur d'interruptions pour gérer les interruptions matériels.
- Les périphériques sont connectés au contrôleur d'interruption qui se trouve sur la carte mère par des lignes appelés IRQ. Les PC utilisent 16 lignes IRQ numéroté de 0 a 15.

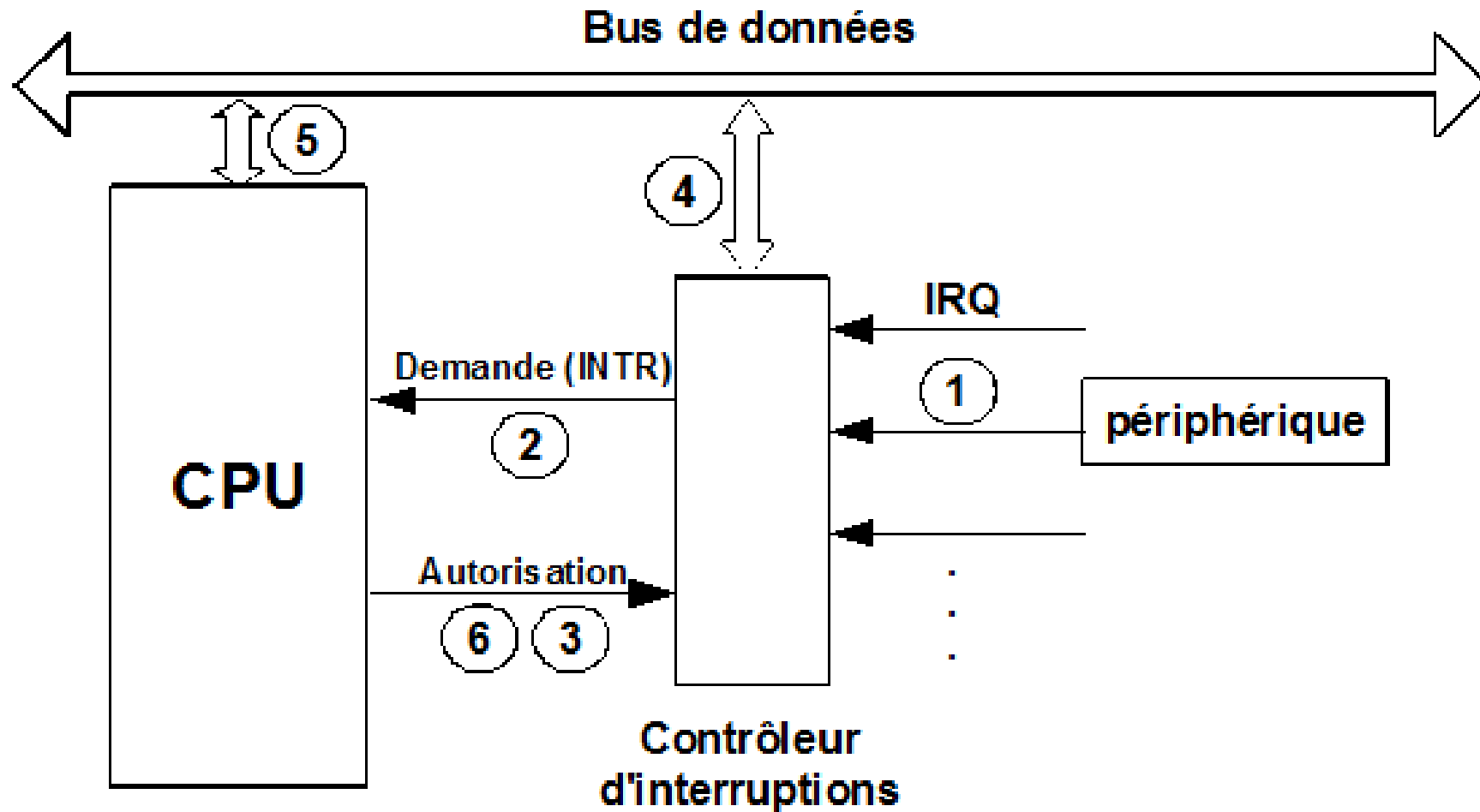


Systeme d'interruption dans les PC

Traitement d'une interruption

- Le périphérique nécessitant une interruption envoie un signal au contrôleur d'interruptions sur une ligne IRQ, **(1)**
- Le contrôleur d'interruption détecte le signal. Si aucune interruption n'est en attente le contrôleur la traite immédiatement, si non il traite l'interruption la plus prioritaire (si plusieurs interruptions sont déclenchés en même temps),
- Pour traiter l'interruption le contrôleur envoie un signal au processeur, **(2)**
- Le processeur reçoit le signal d'interruption et envoie un signal d'autorisation au contrôleur, **(3)**
- Le contrôleur reçoit l'autorisation du processeur et écrit le numéro de l'interruption sur le bus de données. **(4)**
- Le processeur lit le numéro de l'interruption sur le bus de données et utilise ce numéro pour trouver l'adresse de la routine d'interruption et l'exécute, **(5)**
- Quand le processeur termine l'exécution de la routine d'interruption il envoie un signal au contrôleur d'interruption, celui ci va traiter une autre interruption. **(6)**

Traitement d'une interruption



- **Exemple de numéro d'interruptions matériels :**

- IRQ0 (interruption N=° 08h) : Timer appelé 18,2 fois/s,
- IRQ1 (interruption N=° 09h) : Clavier,
- IRQ5 (interruption N=° 0Dh) : Disque dur,
- IRQ6 (interruption N=° 0Eh) : Disquette,
- ..etc