Centre Universitaire de Mila

2 ème année licence LMD Informatique

Module : Systèmes d'exploitation 1

Bessouf Hakim

CHAPITRE 5:

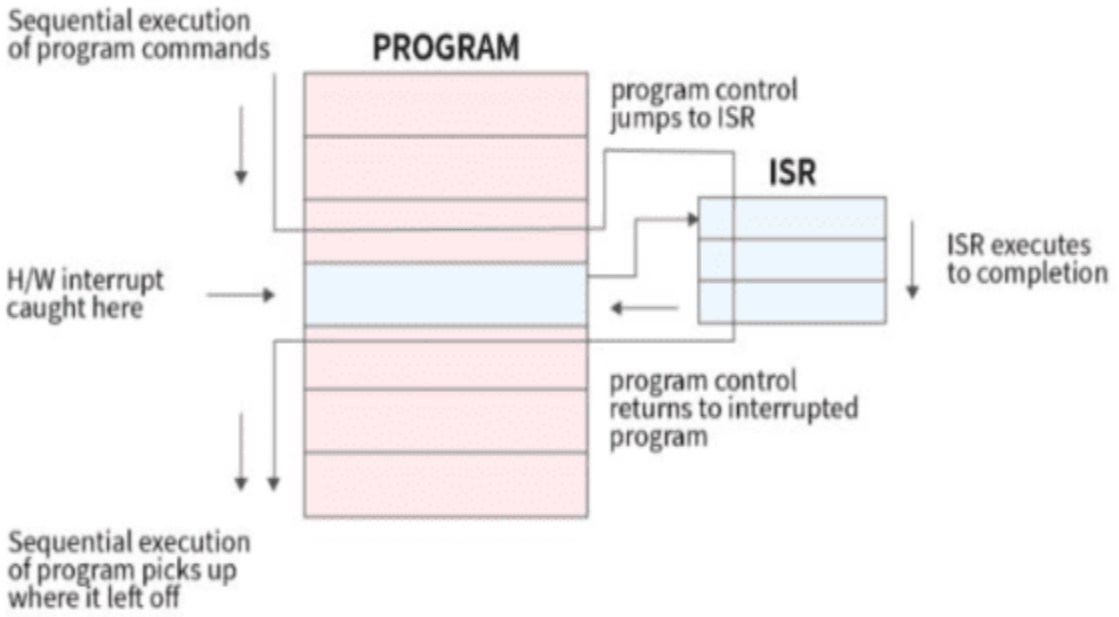
Interrupt Systems

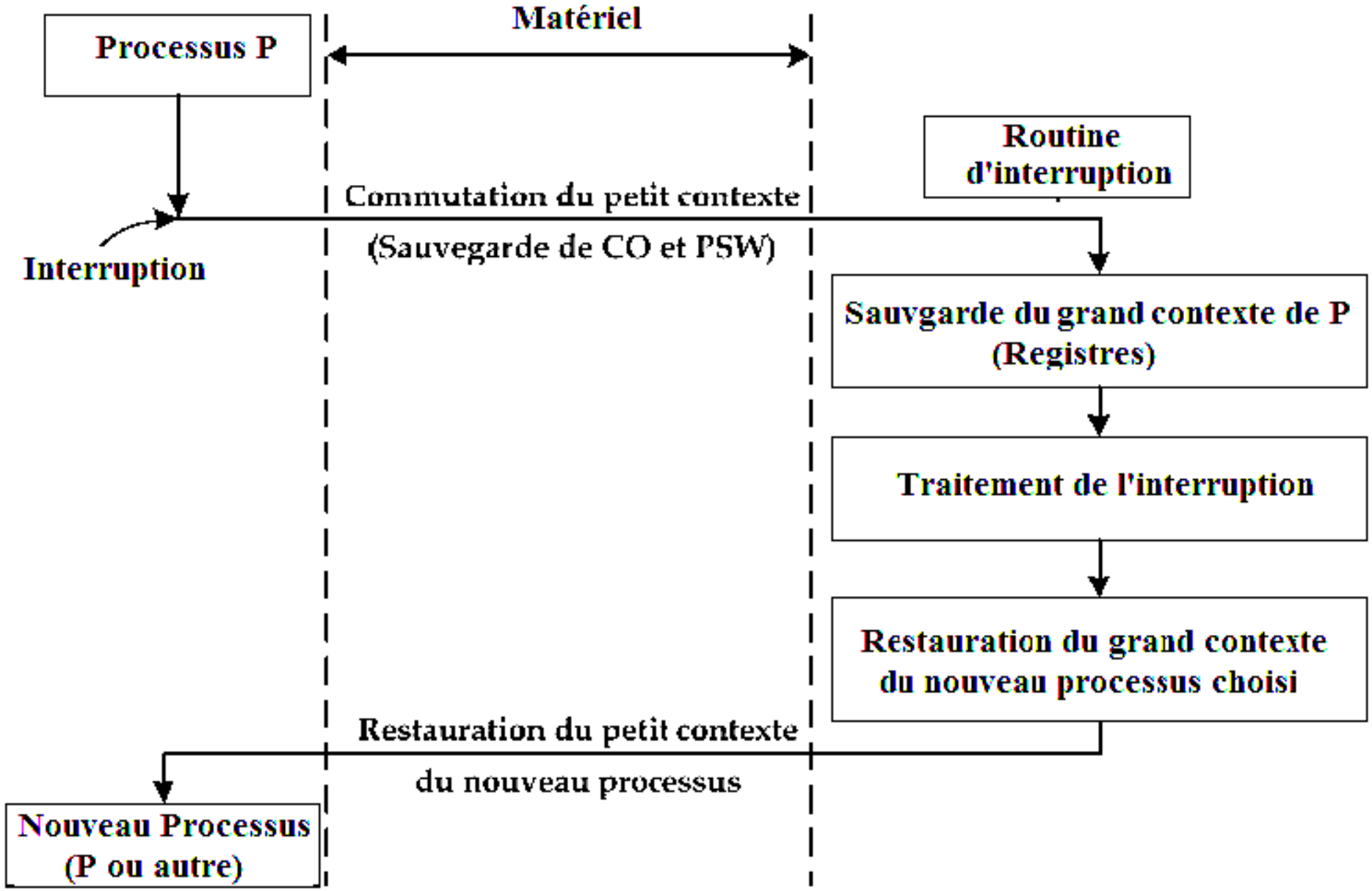
Interrupt Systems:

- Definition and General Flowchart of an Interrupt
- Interrupt Management Mechanisms
- Interrupt Systems on PCs

Interrupt Systems

- An interrupt is a signal sent to the processor to halt (stop) the currently executing program and trigger an **Interrupt Service Routine (ISR)**.
- The processor can only handle interrupts when in an **observable state** (i.e., not mid-instruction execution).
- Before executing the ISR, the processor must **save the context** (e.g., registers, program counter) of the interrupted program.




Matériel
Processus P
Routine d'interruption
Commutation du petit contexte
(Sauvegarde de CO et PSW)
Interruption
Sauvgarde du grand contexte de P (Registres)
Traitement de l'interruption
Restauration du grand contexte du nouveau processus choisi
Restauration du petit contexte
du nouveau processus
Nouveau Processus (P ou autre)

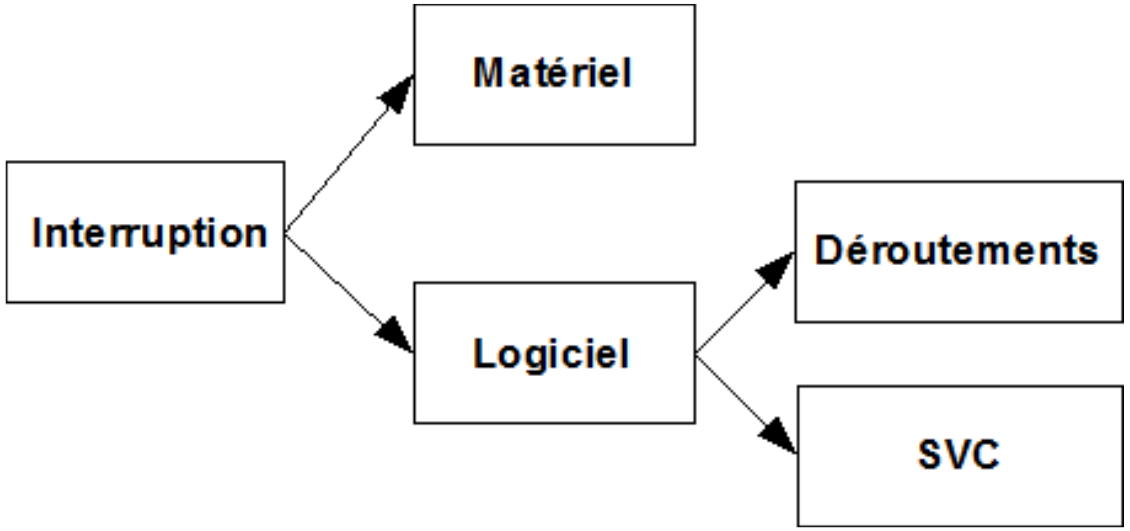
Types of Interrupts

External Interrupts (Hardware):

- Generated by external devices (e.g., clock, peripherals).

Internal Interrupts (Software):

- Caused by programs, subdivided into:
  - **Traps** (e.g., errors like division by zero).
  - **Supervisor Calls (SVCs)** (requests for OS services).



Types d'interruptions.

Types of Interrupts

Traps: Triggered by runtime errors:

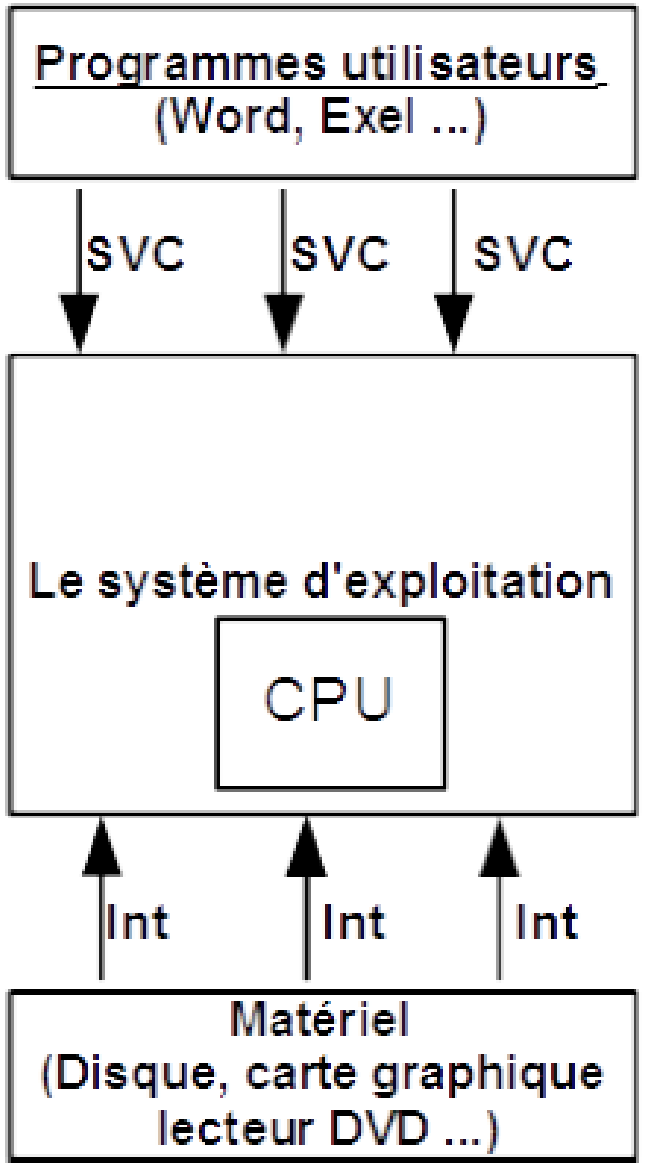
- Division by zero.
- Arithmetic overflow.
- Illegal instruction execution.
- Access to protected memory.

Supervisor Call (SVC):
A privileged instruction that allows a **user program** to request a service from the operating system.

- **Key Note:**This instruction switches the processor's execution mode from **user mode** to **kernel mode** (supervisor mode), granting access to protected OS functions.

Communication via Interrupts

- **Programs** communicate with the OS using **SVCs**.
- **Hardware** communicates with the OS using **hardware interrupts**.



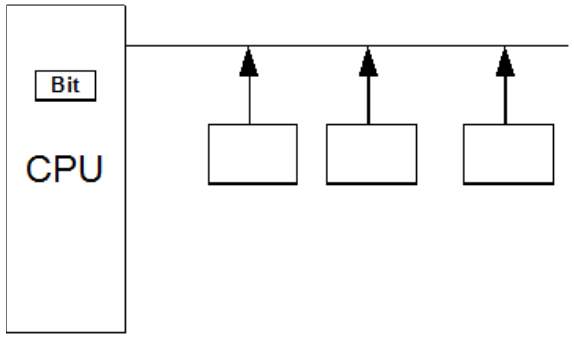
Interrupt Mechanism

An interrupt is an **electrical signal** that sets an **interrupt flag** in the processor. The processor checks this flag at every observable state.

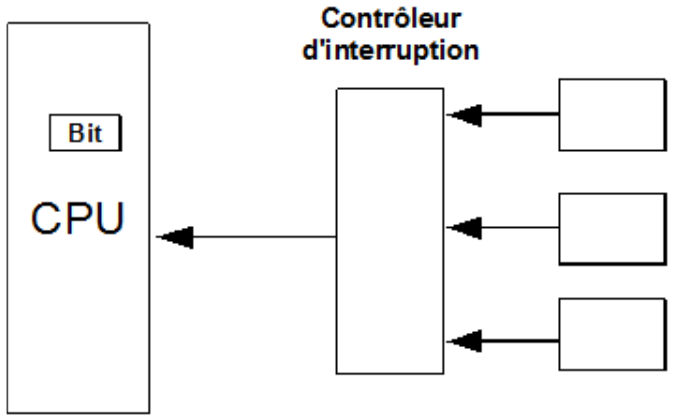
Two Designs:

Single Interrupt Line: One routine handles all interrupts

Multiple Interrupt Lines: Uses an **interrupt controller** for prioritized handling.



Single Interrupt Line



interrupt controller

Interrupt Identification Methods

1) **Polling (Software-Based)**
   - Single ISR checks all devices **sequentially**
   - Checks interrupt lines or device status registers
   - *Simple but slow* – wastes CPU cycles on checks
1) **Vectored (Hardware-Based)**
   - Device sends unique interrupt code to CPU
   - CPU uses code to jump directly to correct ISR
   - Fast but complex – needs priority arbitration

Key Difference

- Polling: "Who interrupted me?" (CPU asks)
- Vectored: "It's me!" (Device tells CPU)

Interrupt Priorities

- **Why Priorities?**

When multiple interrupts occur simultaneously, The system must decide which to handle first

- **Priority Schemes**
  - **- Fixed Priorities**
    - Predefined order (e.g., timer > disk > keyboard)
    - Simple and predictable
  - **- Programmable Priorities**
    - Can be changed dynamically by software
    - Flexible for different scenarios

Higher-priority interrupts can preempt lower-priority ones

Interrupt Masking

- **Why Mask Interrupts?**

To **protect critical code sections** from being interrupted, Example: Handling a high-priority interrupt, Executing time-sensitive operations.

- **How It Works**

Masked interrupts are temporarily ignored, and remain **pending** until unmasked,

CPU can mask:

Specific interrupts (e.g., mask keyboard only), mask all interrupts (global disable)

Key Use Case

•Prevents nested interrupts from causing instability

(Main → ISR A → ISR B → ISR A → Main)

Interrupt Systems in PCs

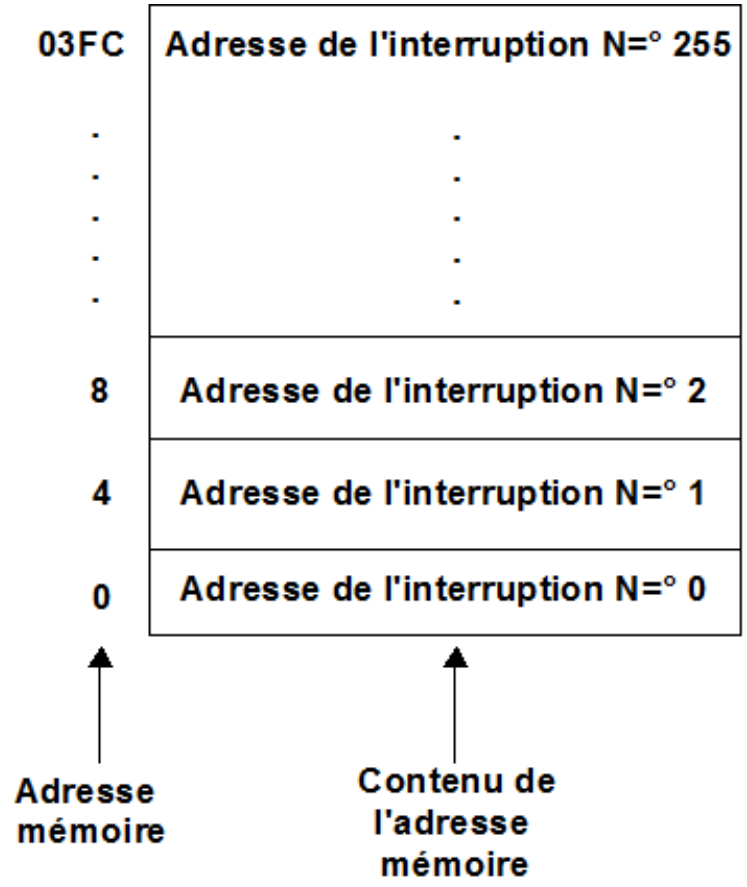
256 Interrupt Levels (0–255):

- 0–31: CPU-reserved (e.g., traps, faults)
- 32–255: User-definable (hardware/OS interrupts)

Interrupt Vector Table (IVT)

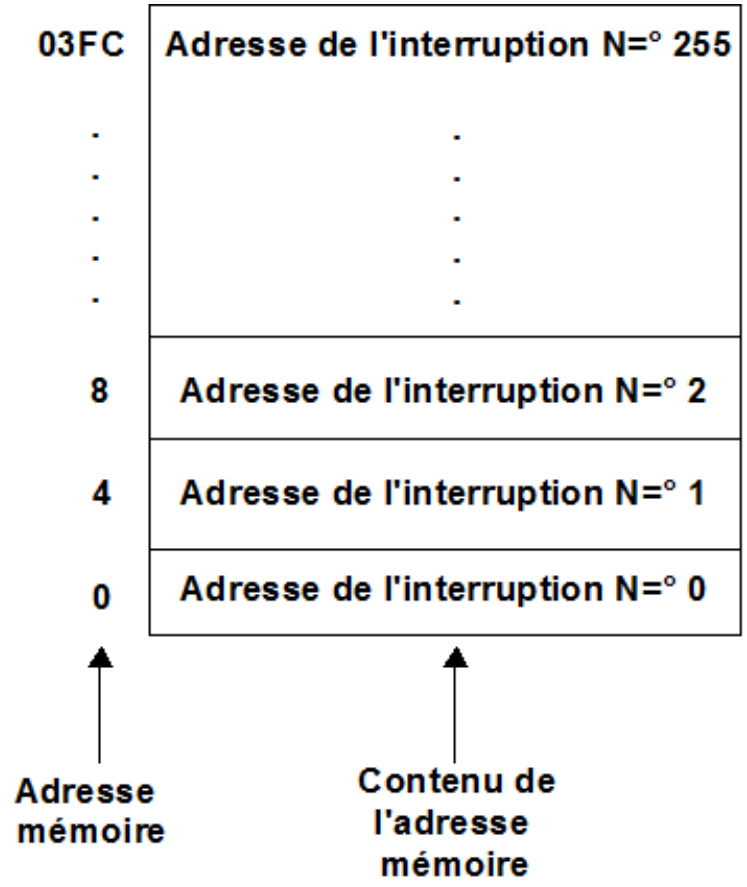
- Maps interrupt numbers → ISR addresses
- Entry Size: 4 bytes (32-bit: 2 bytes segment + 2 bytes offset)
- Location: Stored in RAM (address 0x00000000 in real mode)

ISR Address=Interrupt Number×4 Example: INT 5 → 5 × 4 = 20 (0x14 in hex)



Interrupt Systems in PCs

- 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 x 4 = 20 = 14H**. (H veut dire que le nombre est en hexadécimale).



- Chaque niveau d'interruption correspond à 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 programme utilise 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 utilise l'interruption 21h.

Programme C	**Programme compilé**
int main()	**.**
{	**.**
.	**.**
.	***Mov ah, 01h***
scanf("%d", &x);	***INT 21h***
.	**.**
}	**.**

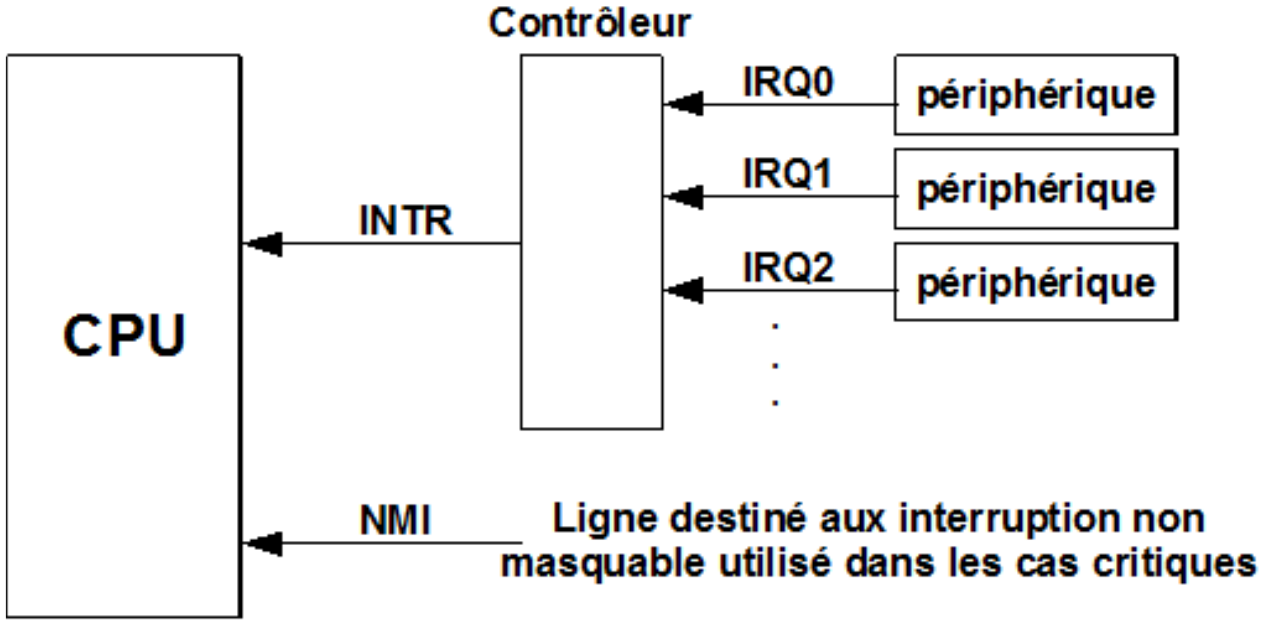
- L'appel système INT 21h est automatiquement inséré par le compilateur. L'interruption 21h permet d’accéder a une fonction système qui permet de lire des caractères du clavier.

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 connectes 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.



Système d'interruption dans les PC

Traitement d'une interruption

(1) Le périphérique nécessitant une interruption envoi un signal au contrôleur d'interruptions sur une ligne IRQ

- 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),

(2) Pour traiter l'interruption le contrôleur envoi un signal au processeur,

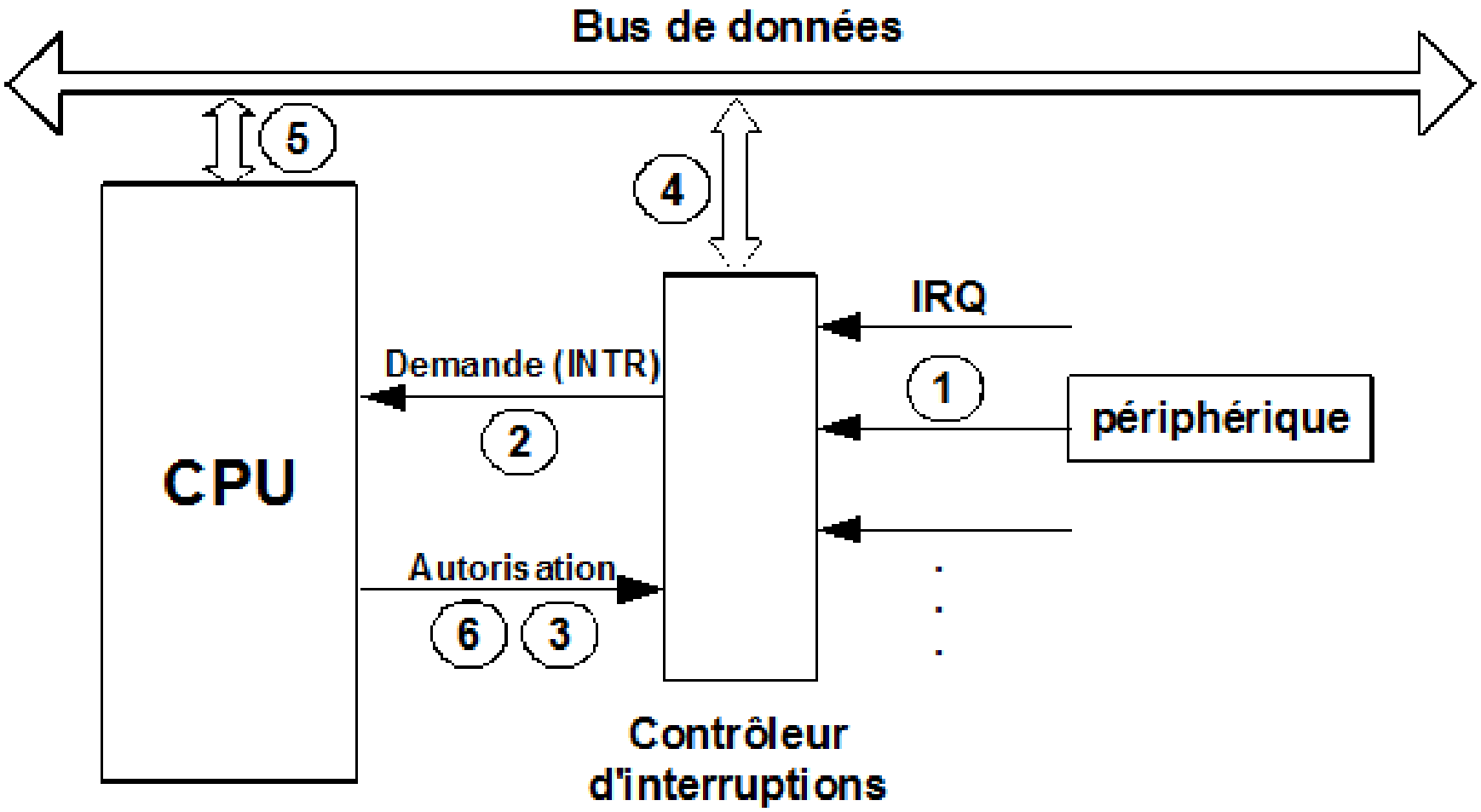
(3) Le processeur reçoit le signal d'interruption et envoi un signal d'autorisation au contrôleur,

(4) Le contrôleur reçoit l'autorisation du processeur et écrit le numéro de l'interruption sur le bus de données.

(5) 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,

(6) Quand le processeur termine l'exécution de la routine d'interruption il envoi un signal au contrôleur d'interruption, celui ci va traiter une autre interruption.

Traitement d'une interruption



- **Exemple de numéro d'interruptions matériels :**
  - IRQ0 (interruption N=° 08h) : Timer,
  - IRQ1 (interruption N=° 09h) : Clavier,
  - IRQ5 (interruption N=° 0Dh) : Disque dur,
  - IRQ6 (interruption N=° 0Eh) : Disquette,
  - ..etc.