

Chapitre 01: Les interruptions

Cours 01

Plan du cours:

- 1. Modes d'exécution du processeur**
- 2. Comment communique les applications avec le SE ?**
- 3. Qu'est ce qu'une interruption ?**
- 4. Comment communique le matériel avec le SE ?**

Plan du cours:

- 1. Modes d'exécution du processeur**
2. Comment communique les applications avec le SE ?
3. Qu'est ce qu'une interruption ?
4. Comment communique le matériel avec le SE ?

Modes d'exécution du processeur

1. Un programme informatique est une suite d'instructions chargé en mémoire centrale par le SE (le module chargeur)
2. Le processeur exécute le programme instruction par instruction sous le pilotage de l'unité de commande.
3. Les processeurs récents ont au moins deux modes de fonctionnement,
 - **Le mode noyau (système ou superviseur):** le processeur peut exécuter toutes les instructions disponibles du système.
 - **Le mode utilisateur:** certaines instructions sont interdites au processeur pour assurer la sécurité du système.

⇒ Quelles sont ces instructions qui sont interdites au processeur ?

Modes d'exécution du processeur

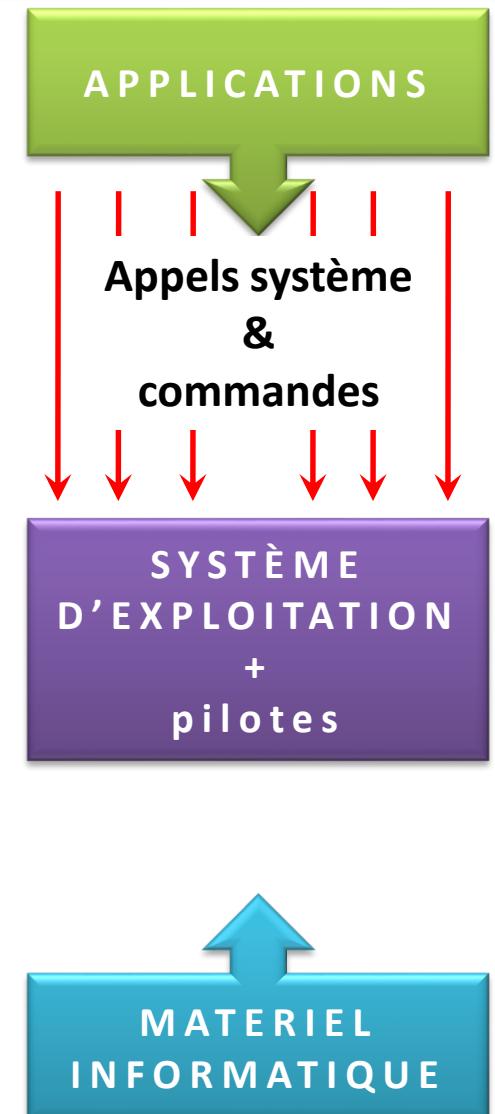
- Ce sont toutes les instructions qui touchent aux ressources et aux données (exemple: lecture d'une donnée à partir du clavier, écrire dans un fichier,...,etc.).
- Elles sont contrôlées par le SE, et ne peuvent être manipulées qu'en mode noyau.
- On distingue deux types de ces instructions: **les appels système** et **les commandes** (interpréteur de commandes).
- Par défaut, Les programmes sont exécutés en mode utilisateur.
- Quand un programme exécute un appel système, il passe en mode noyau.

Plan du cours:

1. Modes d'exécution du processeur
2. Comment communique les applications avec le SE ?
3. Qu'est ce qu'une interruption ?
4. Comment communique le matériel avec le SE ?

Comment communique les applications avec le SE ?

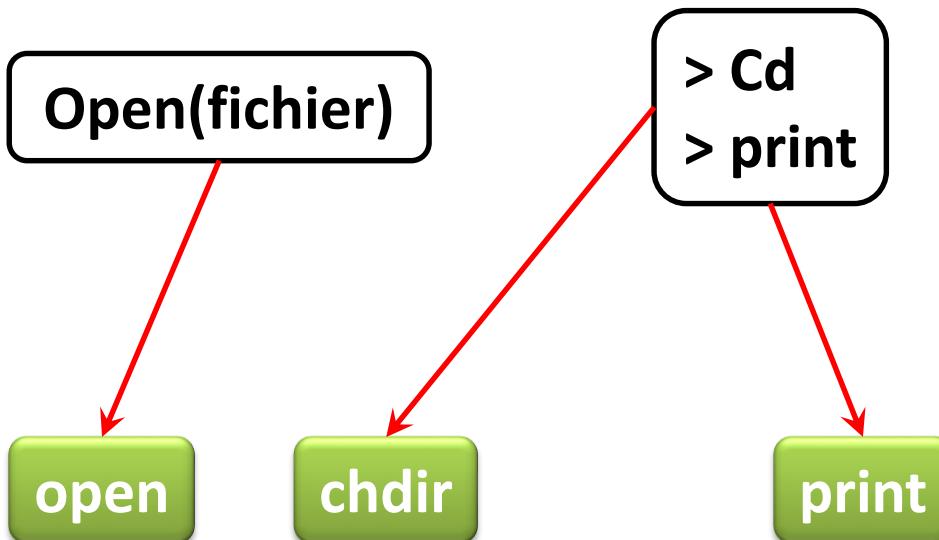
- Le SE se présente donc comme une couche logicielle placée entre la machine et les applications.
- Il s'interface avec les applications par le biais des primitives qu'il offre: **appels système et commandes**



Comment communique les applications avec le SE ?

Programme Utilisateur

Interpréteur de commandes



Appels système

SE

Système de Gestion
de fichier

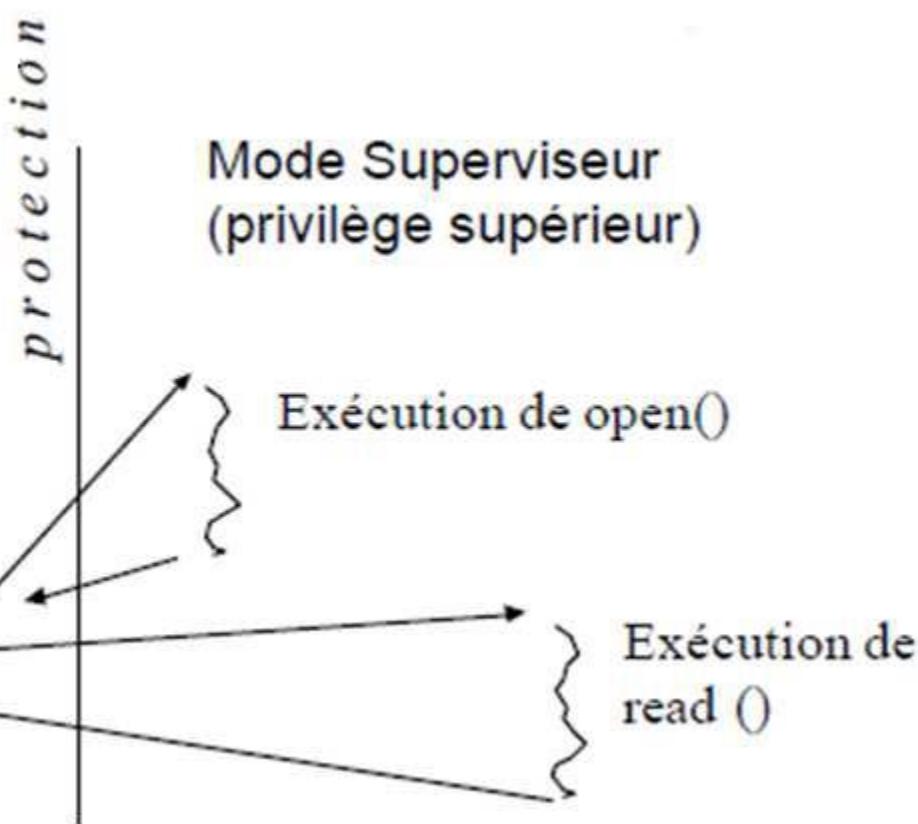
Système de Gestion
des E/S

Comment communique les applications avec le SE ?

L'exécution d'un appel système est similaire à un sous programme.

Mode utilisateur

```
main()
{
    int i, j, fd;
    i = 3;
    fd = open("fichier", "wr");
    read(fd, j, 1);
    j = j / i; }
```

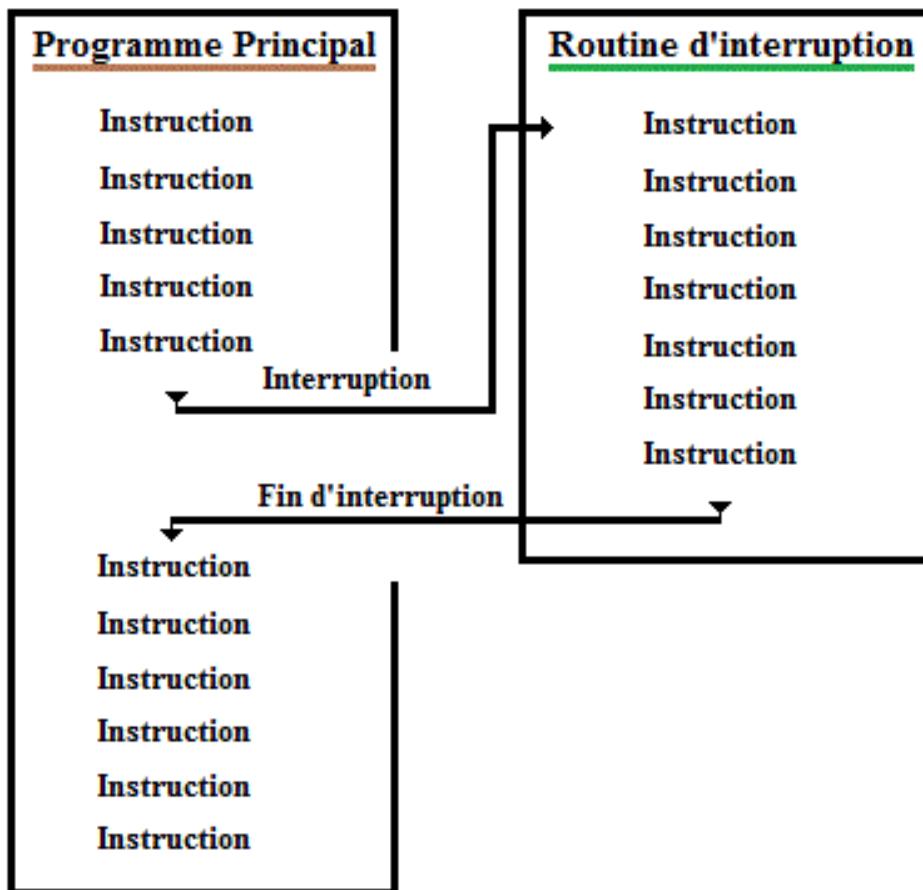


Plan du cours:

1. Modes d'exécution du processeur
2. Comment communique les applications avec le SE ?
3. Qu'est ce qu'une interruption ?
4. Comment communique le matériel avec le SE ?

Qu'est ce qu'une interruption ?

C'est un événement permettant d'arrêter temporairement l'exécution d'un programme pour exécuter un autre programme (appelé routine d'interruption).



Qu'est ce qu'une interruption ?

Les causes d'une interruption peuvent être:

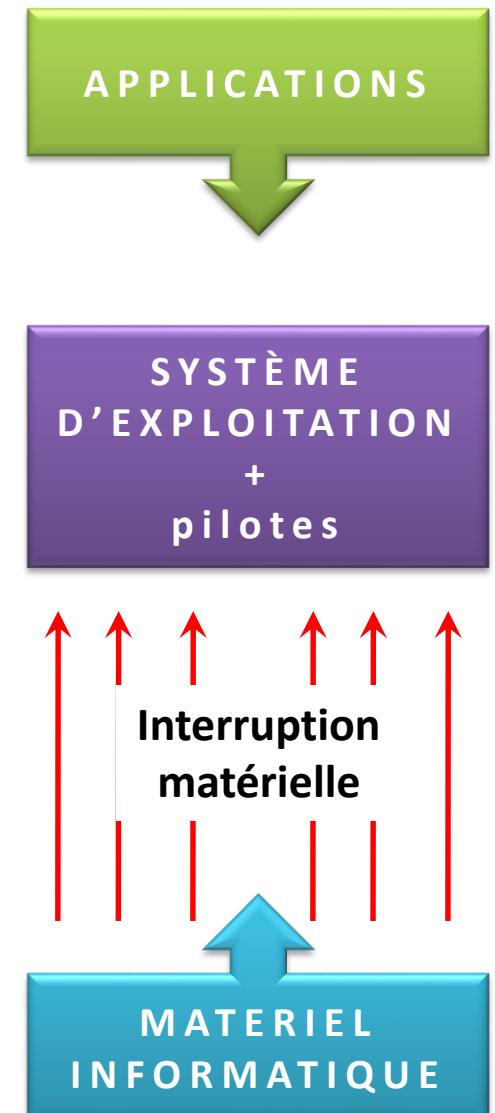
- **Logicielles** (Interne au programme) comme les **appels systèmes** et les **déroutements (trappe)** qui sont provoqués par une opération illicite (ex.: division par zéro), et
- **Matérielles** (externe au programme), générées par le matériel (ex.: bouger la souris, paquet réseau arrive, ...,etc.).

Plan du cours:

- 1. Modes d'exécution du processeur**
- 2. Comment communique les applications avec le SE ?**
- 3. Qu'est ce qu'une interruption ?**
- 4. Comment communique le matériel avec le SE ?**

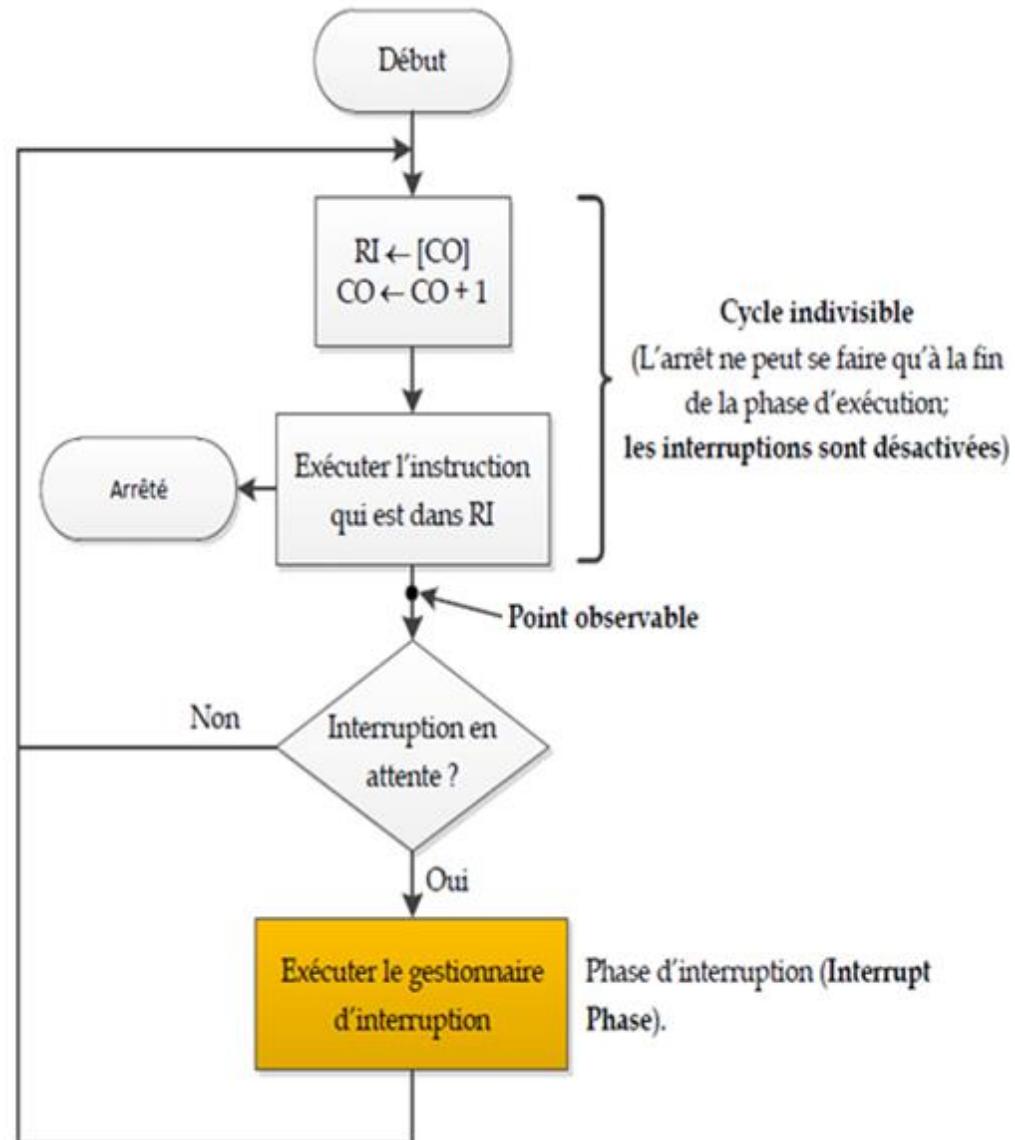
Comment communique le matériel avec le SE ?

- Le SE s'interface avec la couche matérielle, par le biais du mécanisme des **interruptions matérielles**.
- L'interruption est notée par **IRQ***: Interrupt Request où * désigne le numéro spécifié à l'interruption.
- Chaque interruption est associée à un bit qu'il peut modifier, l'ensemble de ces bits forme le **registre d'interruption**.
- Le **vecteur d'interruption** est un ensemble de cases mémoire contenant les adresses des routines d'interruptions.



Comment communique le matériel avec le SE ?

1. Le processeur termine l'exécution d'une instruction,
2. Il détecte que l'interruption **n** est levée,
3. Il sauvegarde le CO (et parfois d'autres registres) en mémoire,
4. Il charge la case **n** du vecteur d'interruption dans le CO,
5. Il exécute la routine d'interruption,
6. Ensuite, il restaure le CO (et autre registres)
7. Enfin, il exécute l'instruction suivante



Comment communique le matériel avec le SE ?

Masquage et priorité des interruptions:

On peut avoir à un moment donné, plusieurs indicateurs d'interruption positionnés en même temps. D'où l'utilisation de **priorités** sur les niveaux d'interruption.

Cependant, certains programmes nécessitent que leur exécution soit complète sans interruption. Le **masquage** répond à ce besoin. Cette action désarme les priorités sur les interruptions. Ainsi:

- En masquant les interruptions, on n'autorise plus l'arrêt d'un processus en cours,
- En démasquant, on arme à nouveau le mécanisme des interruptions.

Comment communique le matériel avec le SE ?

Exercice 1:

Citer les trois grandes causes de passage du mode utilisateur au mode noyau :

Solution:

1. **Appel système:** le programme utilisateur appelle une fonction du système, c'est une demande explicite du passage en mode noyau.
2. **Trappe (Déroutement):** l'exécution par le programme utilisateur d'une opération illicite (division par 0, instruction machine interdite, violation mémoire, etc.). L'exécution du programme utilisateur est alors arrêtée.
3. **Interruption:** sa prise en compte par le matériel et le SE. Le programme utilisateur est alors stoppé et l'exécution de la routine d'interruption associée à l'interruption survenue est exécutée en mode noyau.

