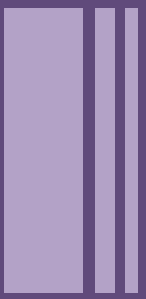
Cours 03:
La Gestion des Interruptions



Plan du cours

- Notion des Interruptions IT
- Gestion des Entrées Sorties (E/S)
- Déroulement d'une IT
- Attente Active (Polling)
- Direct Memory Access (DMA)

Section 1 :
Notion des interruptions (IT)

Les interruptions

Dans la vie courante durant la périodes d'activité:

→ l'interruption est généralement prise en charge aussi vite que possible.

(le téléphone qui sonne),

Les interruptions en informatiques ??

CPU ⟷ Périphérique

Communique ?

→ Les exceptions peuvent être **synchrones** ou **asynchrones**

➢ Besoin d'un système

Cet **aspect** doit être **pris en considération:**

- **Priorités**
- **Temporisation**
- **Ignorer**

Les interruptions en informatique

- Sur une machine, le processeur exécute les programmes chargés en mémoire (ou processus).
- Une autre composante peut demander à l'interrompre pour faire temporairement autre chose.
- **Par exemple:**
  - Périphérique : un paquet réseau arrive, mouvement de la souris
  - Gestion d'erreur : erreur arithmétique, instruction invalide ,

► Il faut donc introduire un mécanisme matériel qui indique au processeur d'arrêter le traitement courant.

Ce mécanisme s'appelle une interruption

Définition de l'interruption

- Une interruption (IT) est un mécanisme qui permet **d'interrompre l'exécution** d'un processus suite à un événement **extérieur** ou **intérieur** et de passer le contrôle à une routine dite « **routine d'interruption** » ou traitement d'interruption.

Un bon système doit réaliser les fonctions:

- **Activer / désactiver** le système d'interruption dans son ensemble
- **Masquer / démasquer** individuellement une interruption.
  - Une **interruption masquée** n'est pas ignorée, **elle est mémorisée** , elle *n'est pris en compte que* lorsqu'elle est *démasquée*
- **Etablir une hiérarchie** entre les sources d'interruption avec plusieurs niveaux de priorité, si possible de façon dynamique
- **Associer un programme spécifique** à chaque interruption

Notion d'interruption

☞ Exercice

On suppose qu'une page de texte contient **50 lignes** de **80 caractères** chacune. On considère une imprimante qui imprime **6 page/mn**. Le temps d'écriture d'un caractère dans le registre de sortie de l'imprimante étant très court, il est ignoré. Cette impression est gérée par des E/S avec IT à chaque caractère.

Quel est le pourcentage d'occupation du processeur par les ITs si chaque interruption prend **50 µs** ?.

Section 2 :
Gestion des Entrées Sorties (E/S)

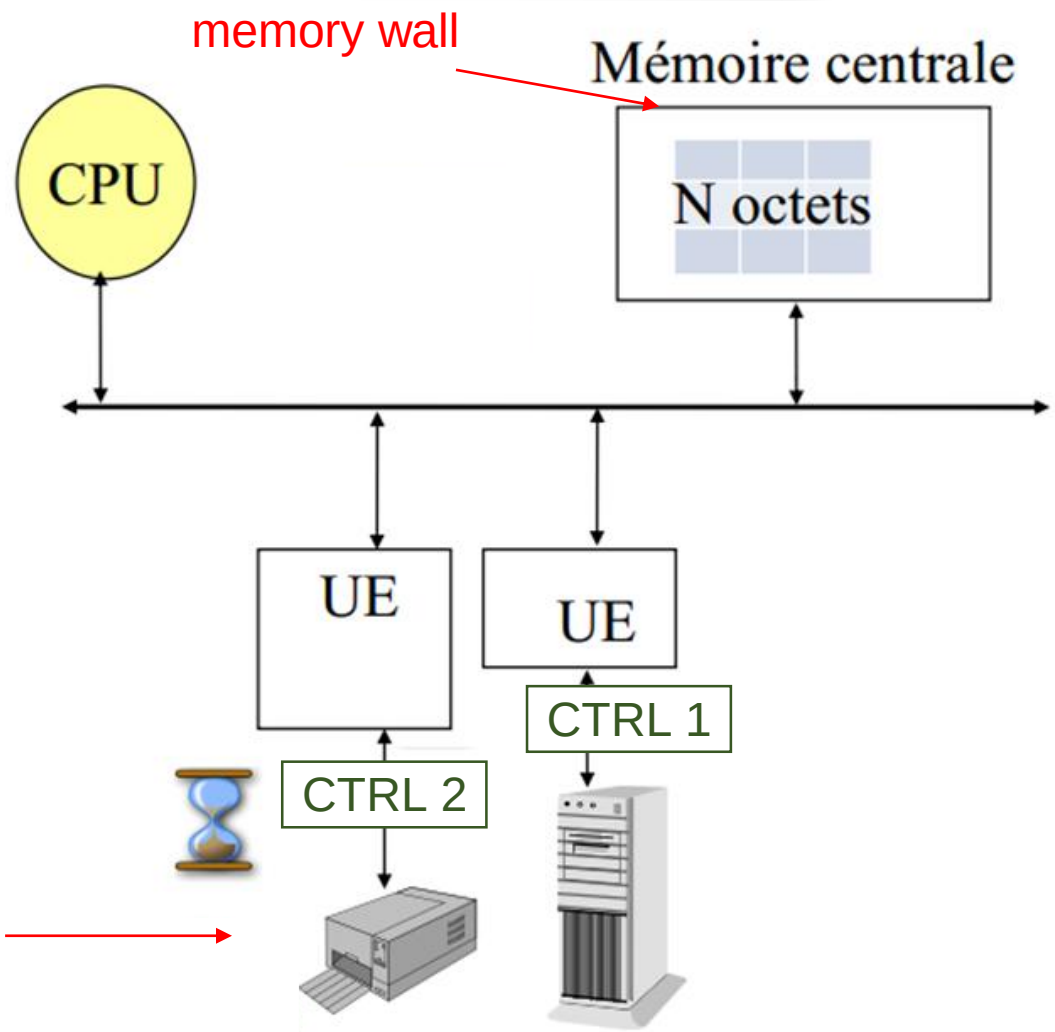
Gestion des Entrées Sorties (E/S)

❑Organisation générale des E/S

- **E/S**: Transfert d'information entre divers niveaux de la hiérarchie de la mémoire:

  registres, mémoire centrale, disques magnétiques …

- Dans l'organisation des E/S, nous distinguons 3 types d'organes :
  - Les périphériques
  - Les contrôleurs CTRL (processeur primaire)
  - Les canaux (UE).

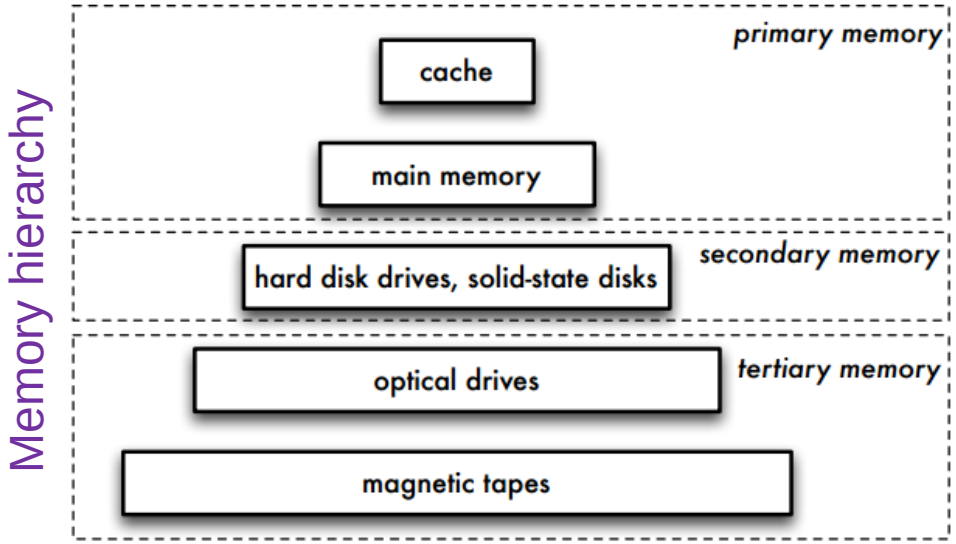


Notion de périphérique

- Les périphériques sont les organes qui assurent les opérations d'Entrées et de Sorties (en abrégé E/S, ou I/O en anglais)
- On désigne par E/S tout transfert d'information depuis ou vers l'ensemble Processeur/Mémoire centrale.

On distingue :

- Transfert d'information entre divers niveaux de la hiérarchie de la mémoire: registres, mémoire centrale, disques magnétiques …
- Transfert d'information depuis ou vers le monde extérieur : périphériques, autres ordinateurs.



Ex. Intel Core i7-965 / RAM 1600MHz

Storage Module	Size	Latency	Transfer Rate
L1-Cache	32KB per core	$1.2ns$	$49GB/s$
L2-Cache	256KB	$3.2ns$	$32GB/s$
L3-Cache	6-8KB	$5.2ns$	$18GB/s$
DDR3 RAM	16-64GB	$11.25ns$	$12.8 - 25.6GB/s$
HDD	1TB	$8.5ms$	$300MB/s$

Notion de contrôleur

- Un contrôleur (on dit parfois coupleur) est un dispositif adapté à chaque type de périphérique.

- Il se connecte à plusieurs périphériques de même type (mais un seul périphérique peut transférer à la fois). Il assure les fonctions logiques des E/S.

- Concrètement, un contrôleur est un processeur primaire comprenant une petite mémoire (dite *tampon* ou *buffer*) et quelques registres spécialisés

Notion de canal

- Appelé parfois unité d'échange (DMA), c'est un processeur spécialisé dans les opérations d'E/S.
- Il ne peut pas être lancé que par un processeur central.
- Il peut interrompre l'UC.
- Son rôle est de *commander* les contrôleurs et les périphériques.

Adressage des périphériques
memory-mapped I/O (MMIO)

- L'adresse d'un périphérique comprend 3 parties :
  - Numéro du canal relié à la mémoire centrale
  - Numéro du contrôleur attaché au canal
  - Numéro du périphérique attaché au contrôleur.

- **Périphérique ?**
  - **Contrôleur ?**
    - **canal ?**

Adresse périphérique
=
[numéro canal, numéro contrôleur, numéro périphérique]

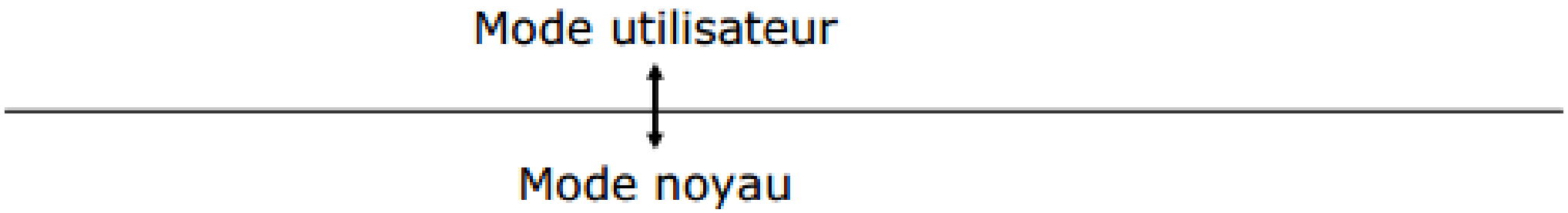
Gestion des Entrées Sorties (E/S)

Comment accéder aux périphériques mappés en mémoire (Memory-Mapped I/O, MMIO) sous Windows et Linux ?

Section 3 :
User mode vs. Kernel mode ?

User mode vs. Kernel mode ?

- Les systèmes d'exploitation d'aujourd'hui possèdent deux modes de fonctionnement
  - Mode utilisateur ou applicatif
  - Mode noyau (kernel)
- Le mode utilisateur permet aux applications de s'exécuter.
- Ces applications accèdent aux ressources du système (ex. matérielles) en utilisant « les services » du noyau du système.
- Le noyau du système garantit l'intégrité de fonctionnement, évitant les plantages dus aux applications mal programmées



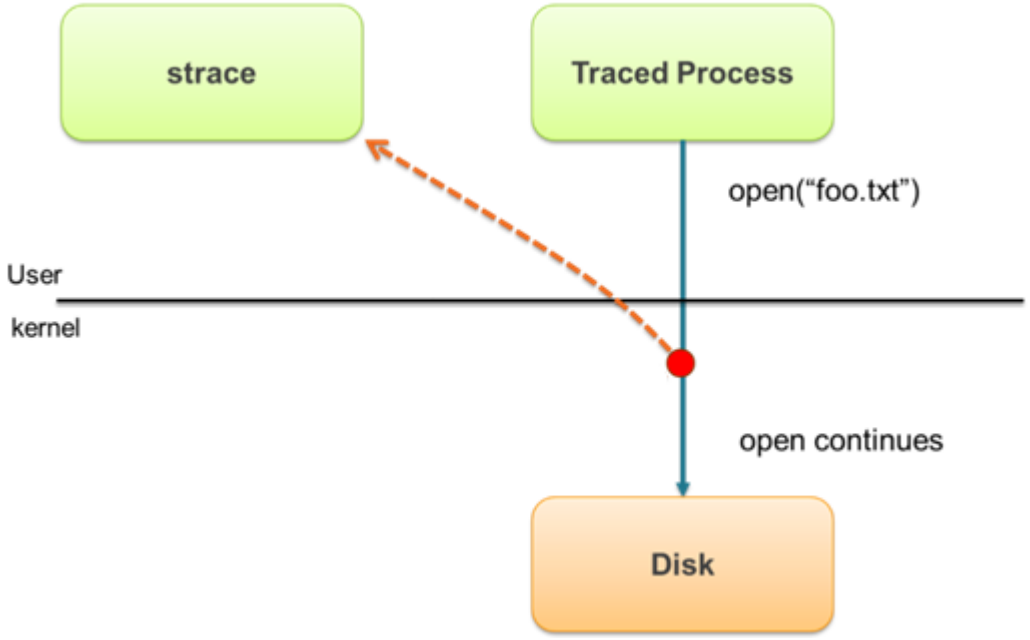
Comment passer en mode noyau ?

⇨Interruption

- Type d'interruptions
  - Interruption Matérielle (IRQ) [Interrupt ReQuest]
    - déclenchées par un périphérique (lecteur, clavier, …)
  - Interruption Logicielle
    - appel système (**appels au superviseur**)
      - ♦ l'utilisateur demande à l'OS un service
      - ♦ permettent à un processus utilisateur de faire un appel au système
      - ♦ appelées par une instruction à l'intérieur d'un programme (*overflow , erreur d'adressage, code opération inexistant, problème de parité, division par zéro* …)

Observer les appels systèmes

- La commande strace intercepte et affiche les appels systèmes d'un programme




```
$ strace echo "coucou"
execve("/bin/echo", ["echo", "coucou"], [/* 54 vars */]) = 0
brk(NULL)                                = 0x25d2000
access("/etc/ld.so.nohwcap", F_OK)       = -1 ENOENT (No such file or directory)
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f619cc01000
access("/etc/ld.so.preload", R_OK)       = -1 ENOENT (No such file or directory)
open("tls/x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("tls/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("x86_64/libc.so.6", O_RDONLY|O_CLOEXEC) = -1 ENOENT (No such file or directory)
open("libc.so.6", O_RDONLY|O_CLOEXEC)    = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
[...]
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 2), ...}) = 0
write(1, "coucou\n", 7coucou
)                 = 7
close(1)                                 = 0
close(2)                                 = 0
exit_group(0)                            = ?
+++ exited with 0 +++
```

- **Autre outils de monitoring (Observation):**

→ strace, tcpdump, netstat, swapon, nicstat, pidstat, …

User mode vs. Kernel mode ?

- Les systèmes Linux fournissent l'utilitaire **strace**
- Les systèmes Solaris et Mac OS X utilisent la commande **dtrace**.
- Les systèmes Windows ne fournissent pas de telles fonctionnalités.

→ Ecrire un programme qui copie le contenu d'un fichier dans un fichier de destination. Ce programme commence par demander à l'utilisateur le nom des fichiers source et de destination.

Écrivez ce programme en utilisant l'API Windows (**Win32**) . Assurez-vous d'inclure toutes les vérifications d'erreur nécessaires

Section 4 :
déroulement d'une interruption

Exécution d'un programme

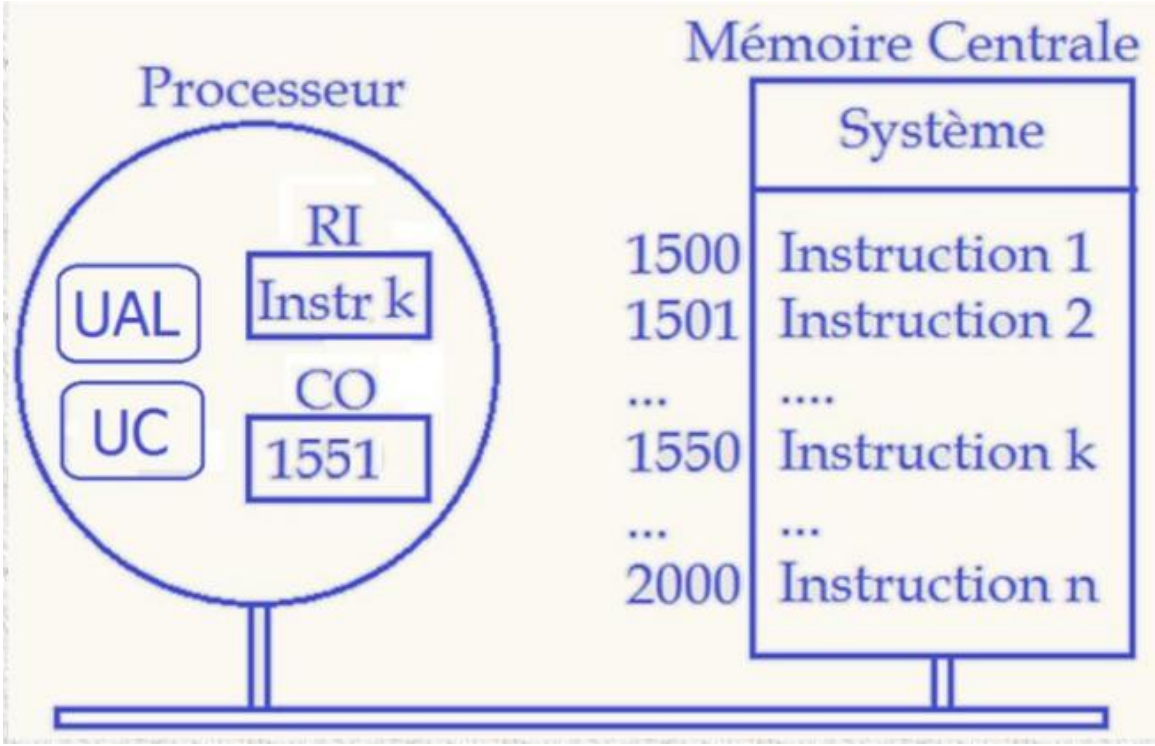
Exécution d'un programme:

- Un programme doit être chargé en MC.
- Le processeur charge l'instruction à exécuter

à partir de la mémoire dans un registre interne appelé Registre Instruction (RI).

- L'adresse de la prochaine instruction/l'instruction

en cours à exécuter se trouve dans un registre spécial appelé Compteur Ordinal (CO).



Mot d'Etat du Processeur PSW

❑ Etat du processeur

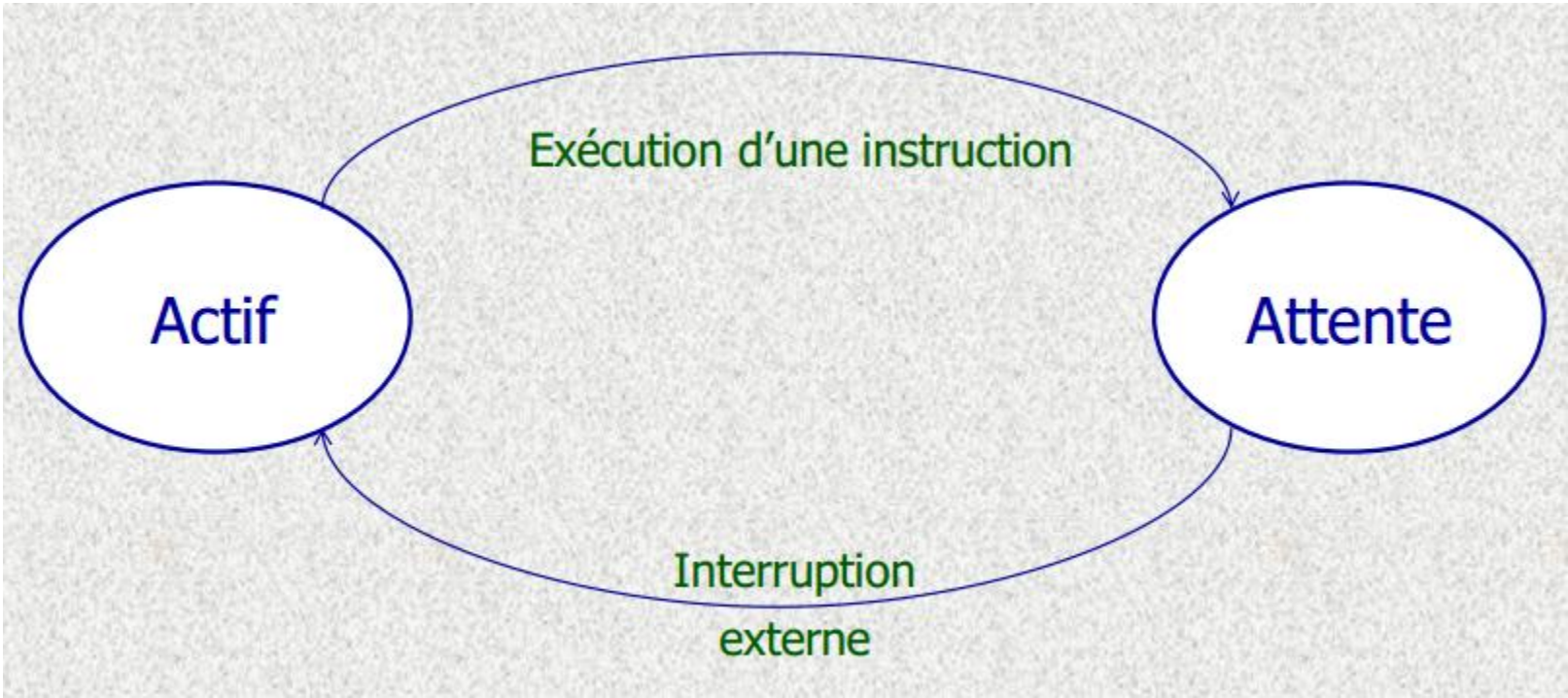
Le SE regroupe des informations-clés sur le fonctionnement du processeur : c'est le mot d'état du processeur (Processor Status Word, PSW)

➔ Il comporte généralement :

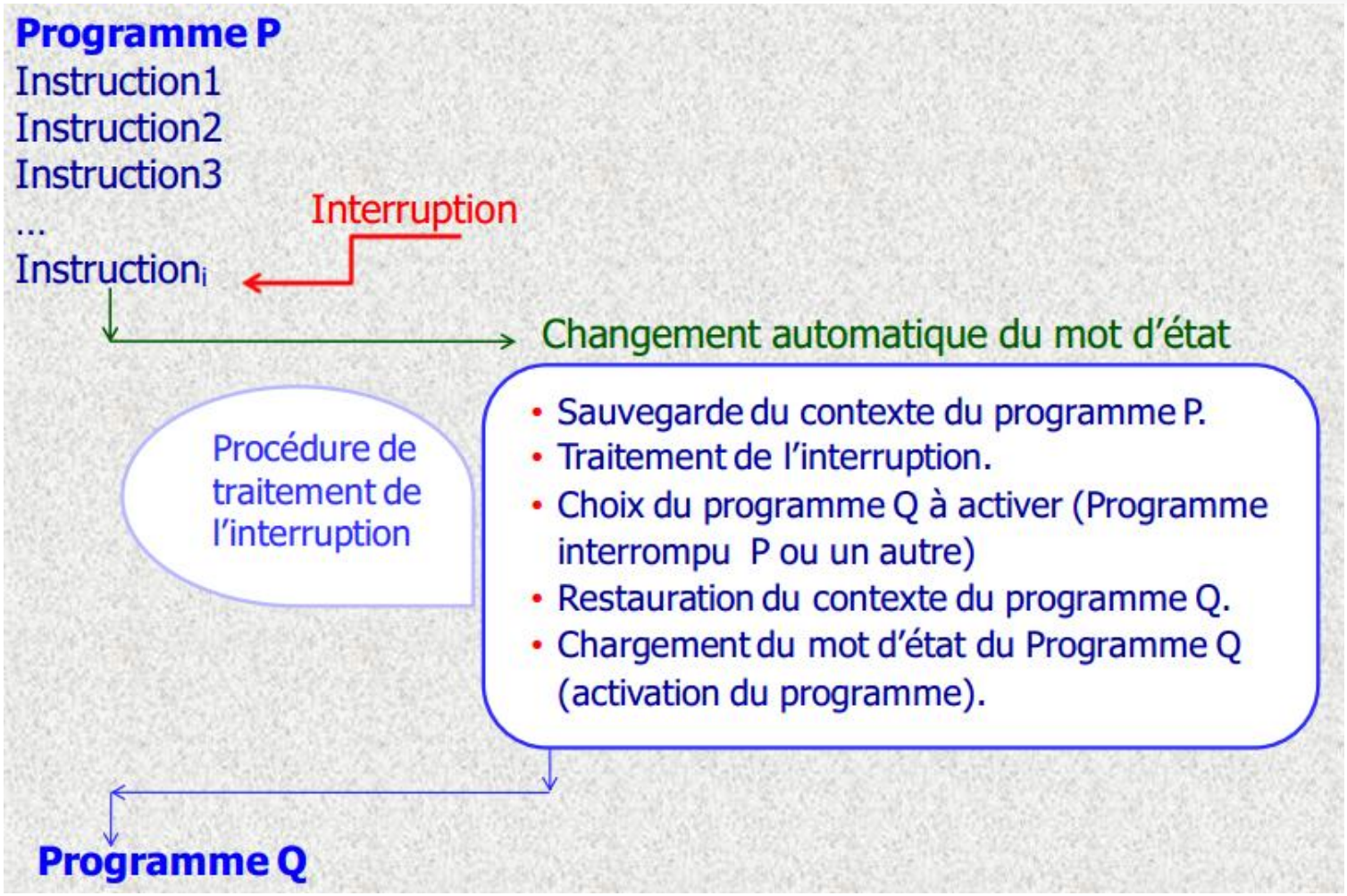
- **Informations sur l'état du processeur**
  - État d'exécution : Actif/Attente
  - Mode d'exécution : Maître/Esclave
  - Masque (informations) sur les interruptions
- **Informations sur les données accessibles et les droits**
  - Table des segments
  - Protection mémoire
- **Informations sur le déroulement de l'activité en cours**
  - Compteur ordinal

Etat d'exécution du processeur

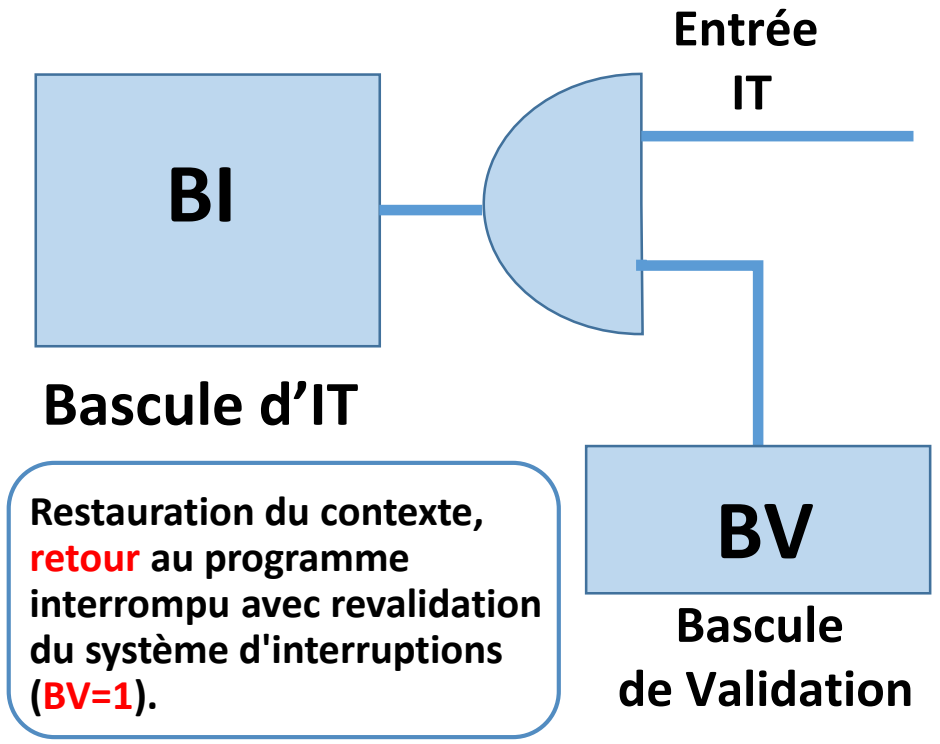
- Le processeur a deux états : **attente** ou **actif**.



Déroulement d'une routine d'interruption



Système d'interruptions simple à une seule entrée:



Restauration du contexte, retour au programme interrompu avec revalidation du système d'interruptions (BV=1).

Interruptions matérielles

Procédure
de traitement des IT

Retrouver l'@ de la **routine** adéquate dans le VI

Périphérique

cartes d'extension
(audio, réseau, cartes SCSI etc.)

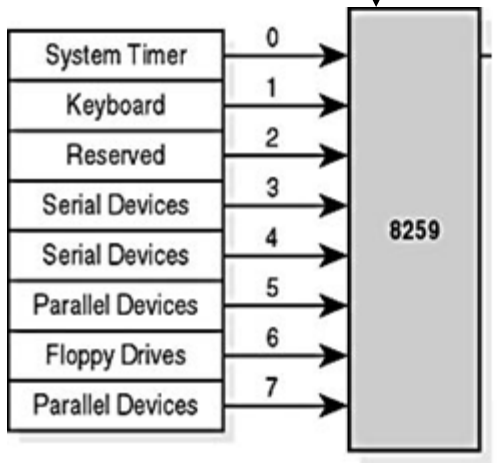
IRQ « Interrupt Request ».

contrôleur d'interruption

Numéro d'origine de signal

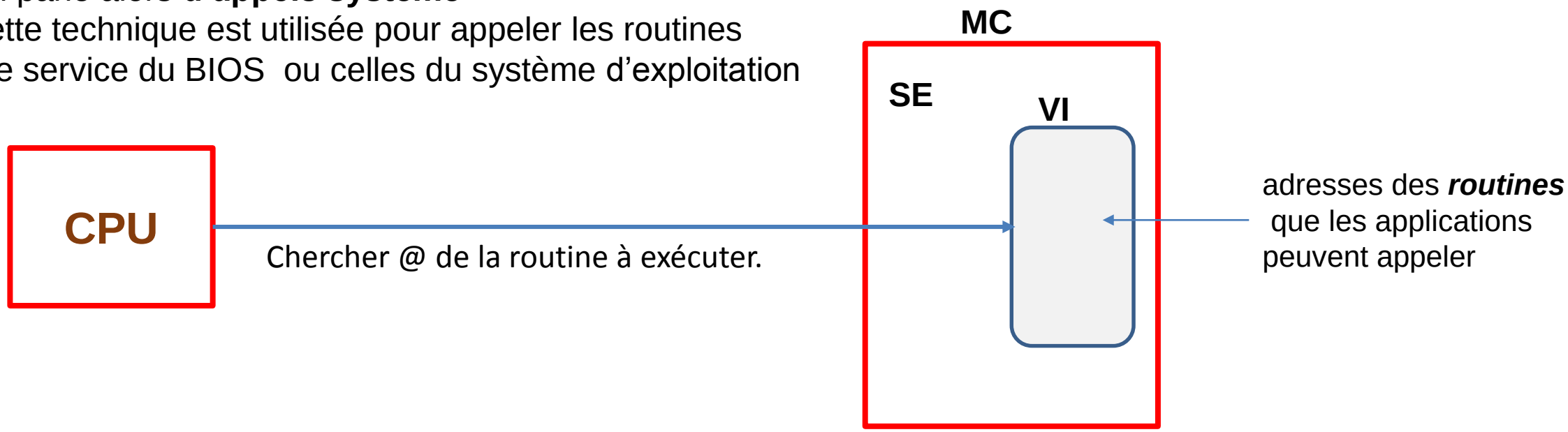
CPU

INT



Interruptions logicielles

- On parle alors **d'appels système**
- Cette technique est utilisée pour appeler les routines de service du BIOS ou celles du système d'exploitation



Le vecteur d'interruption

- Le vecteur d'interruption est une table qui contient les adresses des routines d'interruption.

- Les interruptions sont numérotées et ces numéros (allant de 0 à 255 dans un PC) servent d'index pour rechercher dans le vecteur d'interruption l'adresse de la routine à exécuter.

Le vecteur d'interruption

INTERRUPTS AND BIOS SERVICES		
Services	**Interrupts**	**Pointers**
10H Video services 11H Equipment list 12H Conv. memory size 13H Disk I/O 14H Serial port I/O 15H AT services APM 16H Keyboard I/O 17H Printer I/O 18H ROM-BASIC 19H Bootstrap 1aH Time I/O MRCI hook	00H Division by 0 01H Single-step 02H Non-Maskable 03H Breakpoint 04H Overflow 05H Print screen 06H Invalid opcode 07H no math chip	1bH Keyboard break 1cH User timer interrupt 1dH Video parms 1eH Diskette parms 1fH Graphics chars 22H Terminate addr 23H Ctrl+Break addr 24H Critical Error addr
20H-2fH: DOS Interrupts	08H IRQ 0 Timer 09H IRQ 1 Keyboard 0aH IRQ 2 cascade 0bH IRQ 3 COM 2/4 0cH IRQ 4 COM 1/3 0dH IRQ 5 LPT 2 0eH IRQ 6 diskette 0fH IRQ 7 LPT 1	40H diskette revector 41H hard disk 0 parms 46H hard disk 1 parms
2fH: Multiplex Services		43H EGA font table
31H DPMI services 33H Mouse Support 67H Expanded Memory Fns	70H IRQ 8 RT Clock 71H IRQ 9 redir IRQ 2 72H IRQ 10 (reserved) 73H IRQ 11 (reserved) 74H IRQ 12 (reserved) 75H IRQ 13 math chip 76H IRQ 14 hard disk 77H IRQ 15 (reserved)	4aH User alarm address

Priorité des interruptions

Les systèmes d'IT hiérarchisés

Registre de Mémorisation des niveaux

Regroupement des causes par niveaux.
Association d'une priorité pour chaque niveau.

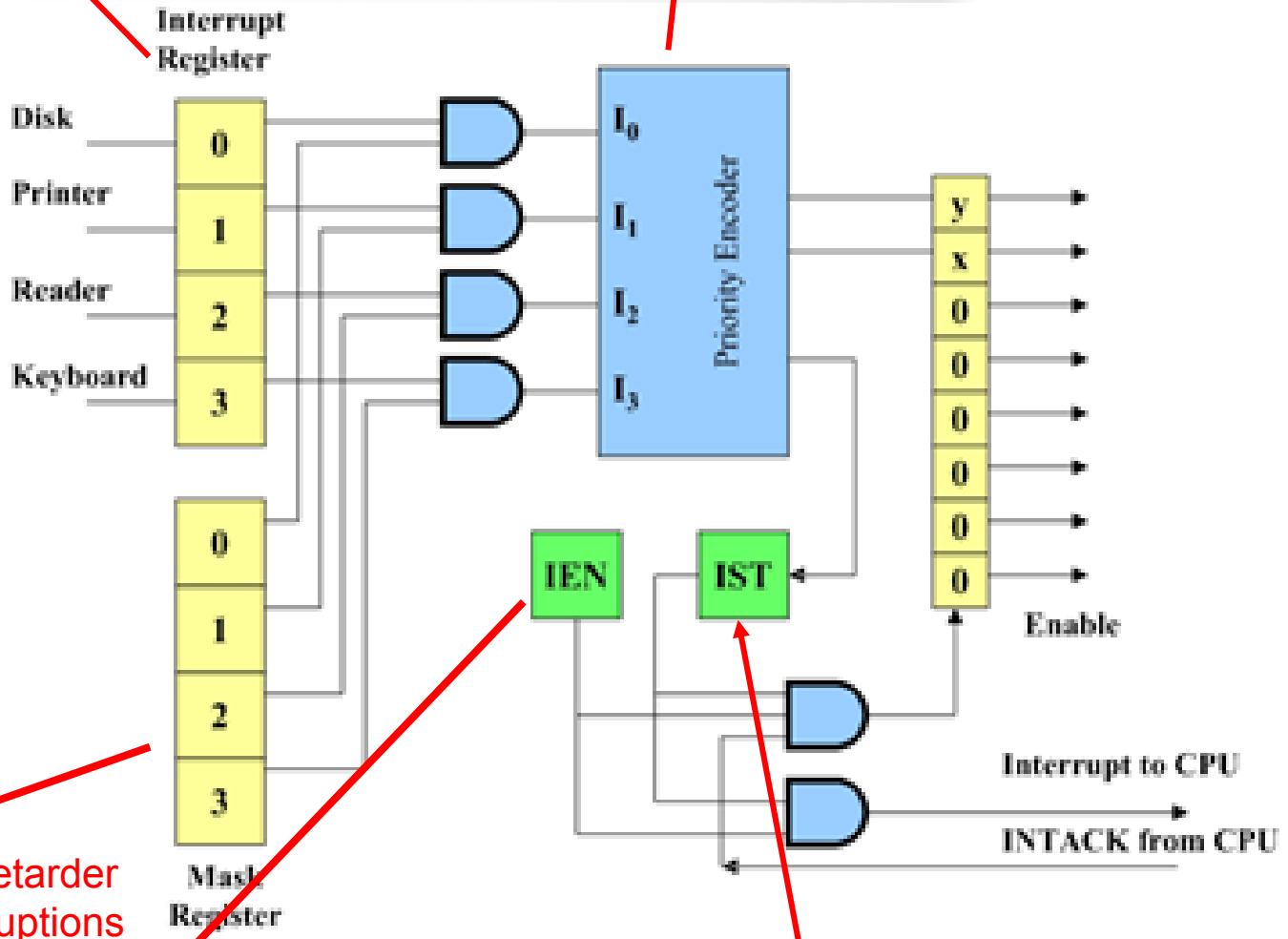
	Nature de l'interruption	fonction de traitement
0	horloge	clockintr
1	disques	diskintr
2	console	ttyintr
3	autres peripheriques	devintr
4	appel system	sottintr
5	autre interruption	otherintr

priorité

Niveau

j est plus prioritaire qu'une interruption de niveau **i**

si **j > i**



Les masques d'interruptions permettent de retarder la prise en compte d'une ou plusieurs interruptions

IEN: Interrupt Enable
Remplir et effacer par des instructions
(Désactiver d'autres interruptions)

output IST [Interrupt Status bits]
(IT prise en compte ?).

Déroulement d'une interruption

Priority Encoder Truth Table

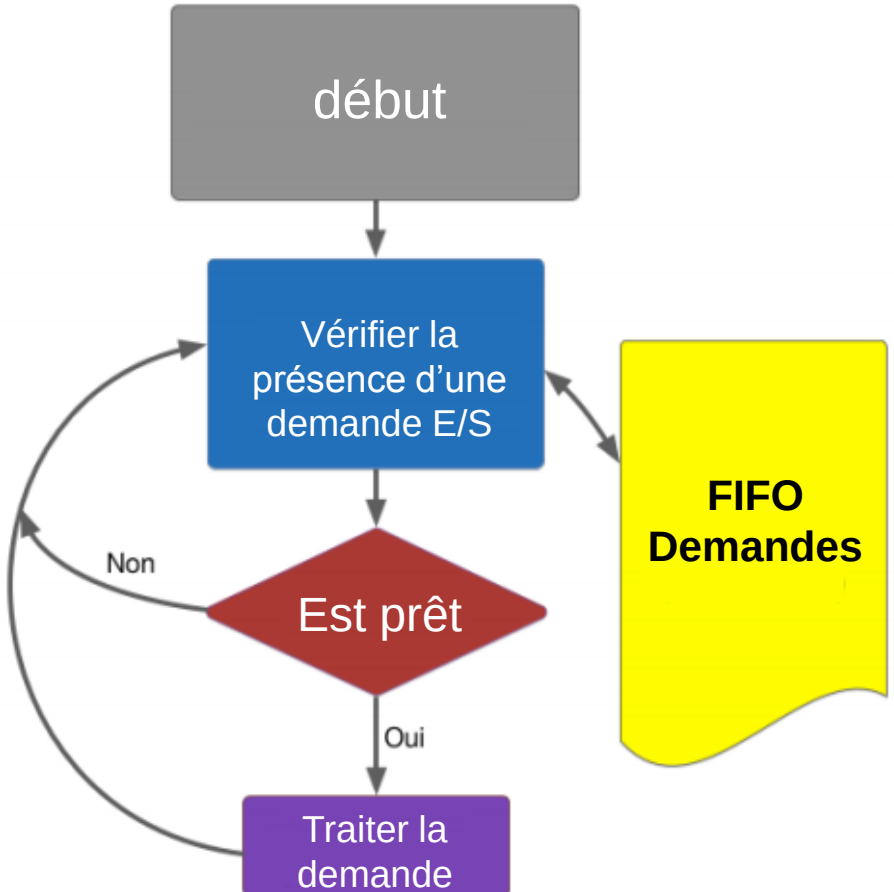
Inputs				Outputs			Boolean functions
I_0	I_1	I_2	I_3	x	y	IST	
1	×	×	×	0	0	1	
0	1	×	×	0	1	1	$x =$
0	0	1	×	1	0	1	$y =$
0	0	0	1	1	1	1	$(IST) =$
0	0	0	0	×	×	0	

Section 5 :
Attente active (Polling)

Attente active (Polling)

L'**attente active**, ou polling est une technique de programmation que les processus utilisent lorsqu'ils vérifient de façon répétée si une condition est vraie, comme l'**attente** d'une entrée ou encore la libération d'un verrou.

- Une alternative au système des interruptions est le polling.
- Chaque élément est testé périodiquement (CPU cycles) pour vérifier s'il demande un service particulier.

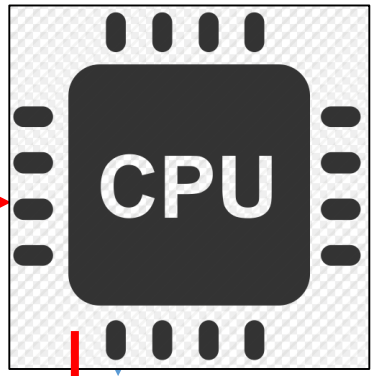


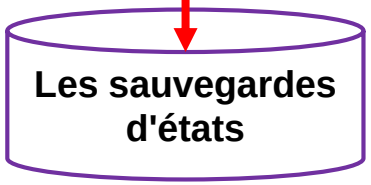
Le polling

☞ Cette méthode n'est pas toujours réalisable ☹

demande des dizaines de milliers d'interventions

(interruptions)





☹ une dégradation des performances

L'usage d'interruptions (IT) par rapport au *polling* permet aussi des économies d'énergie

Section 5 :
Direct Memory Access (DMA)

Exercice

Problème ☹

- Le processeur passe beaucoup de temps à effectuer des instructions de transfert de données
- Beaucoup d'événements d'entrées – sorties nécessitent des transferts de blocs

Solution ☺

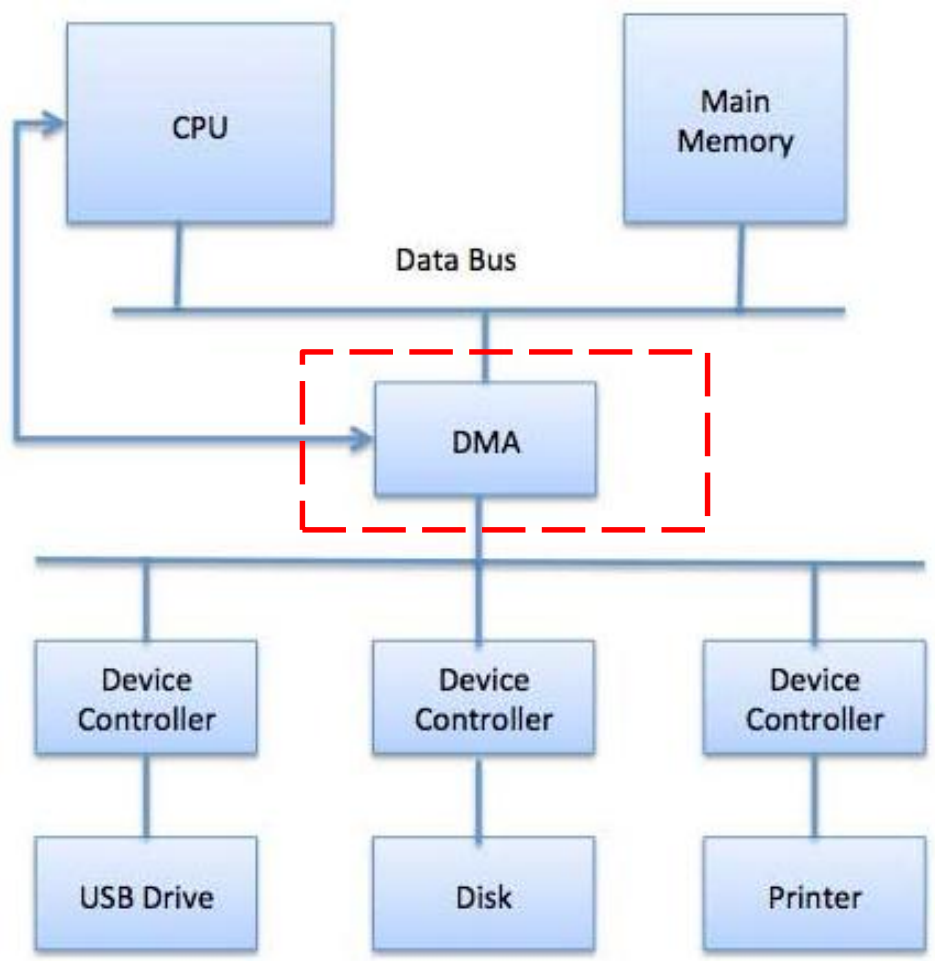
La plupart des ordinateurs possèdent un processeur spécialisé qui gère le dispositif d'accès à la mémoire :

DMA
(Direct Memory Access : accès direct à la mémoire)

Direct Memory Access (DMA)

Délégation de la gestion de l'*échange* (canaux, *DMA*).

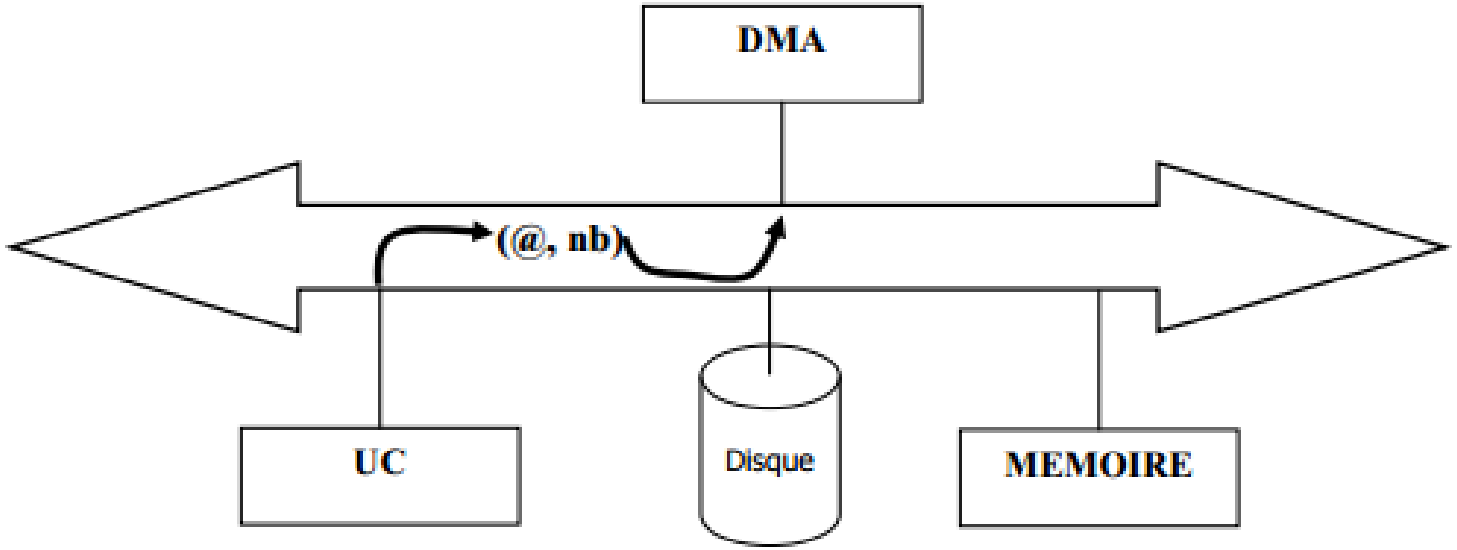
- Le DMA est connecté entre un **contrôleur** de périphérique et le **bus système**, permettant ainsi au périphérique d'accéder à la mémoire sans passer par le CPU.



- Pour transférer de grands blocs de données à grande vitesse
- CPU lancer et conclure le transfert
- La conclusion du transfert ou la disponibilité du périphérique peuvent être signalés par interruption.

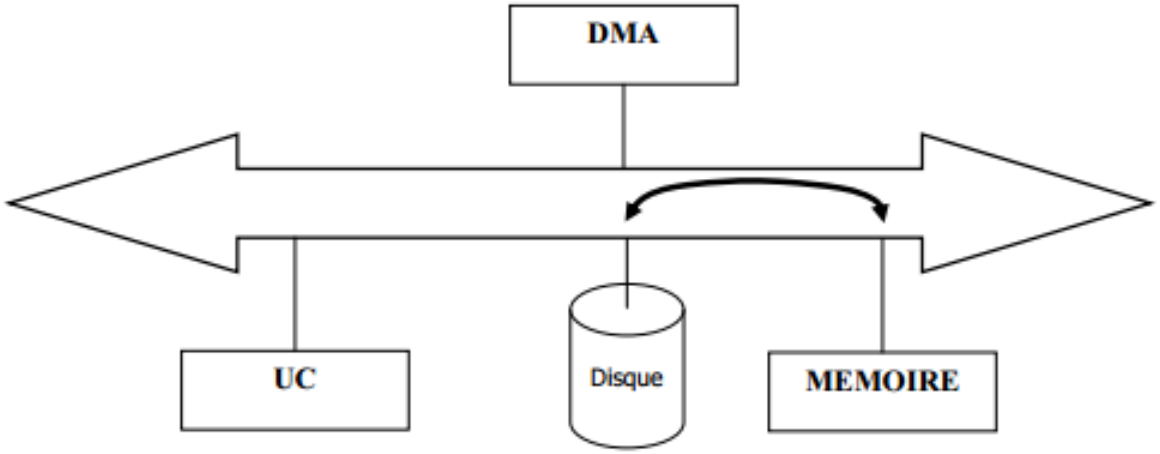
DMA: Etape 1

1 Le processeur initialise les registres du DMA en lui envoyant une adresse mémoire et le nombre d'octets à transférer.



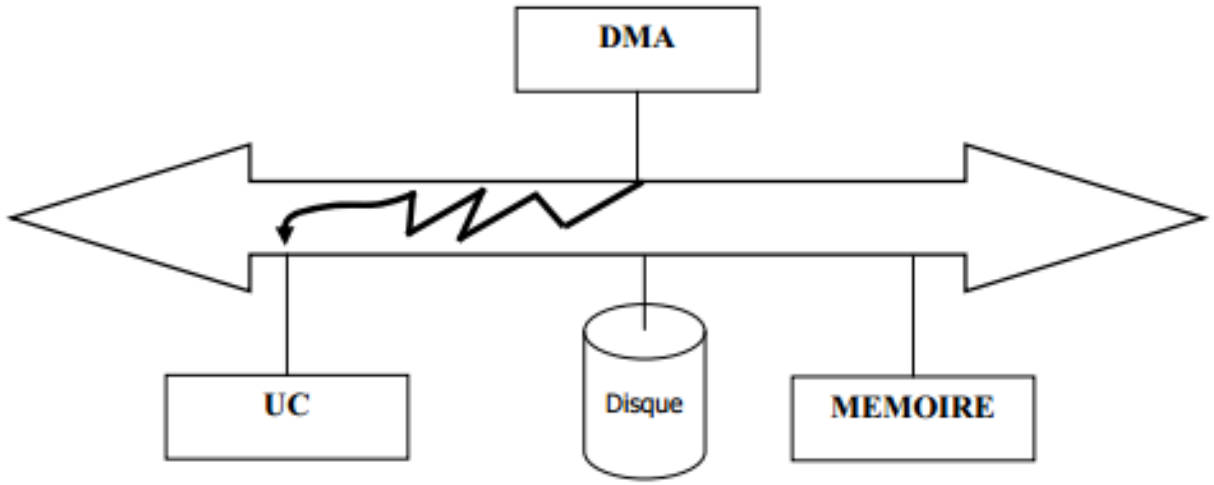
DMA: Etape 2

❷ Le DMA transfert les données entre un disque et la mémoire, pendant ce temps, l'UC travaille autre tache.
Le DMA doit être le maître du bus.



DMA: Etape 3

❸ Une fois le transfert terminé, il y a une génération d'une interruption.



Direct Memory Access (DMA)

Exercice

Écrire un pilote de périphérique.

Pas un vrai périphérique (émulé en C). Néanmoins, très semblable à la manière dont vous programmez un vrai appareil.

1. Le périphérique exporte une série de registres mappés en mémoire.

2. Initialiser les registres selon les spécifications.

3. Interruption du dispositif de configuration dans le système d'exploitation.

4. Copier les données de l'appareil.