

## SYSTEME D'INTERRUPTION

### 1. Définition :

L'interruption est un signal électronique généré par un matériel (exemple : canal, contrôleur,...) ce signal est envoyé au CPU pour provoquer une rupture de séquence afin d'exécuter un programme prioritaire qui va traiter la cause de l'interruption.

Le système d'interruption est un dispositif qui enregistre les signaux d'interruption envoyé au CPU. Ces signaux arrivent à n'importe quel instant, mais ils ne sont pris en charge qu'à la fin de l'opération en cours c'est-à-dire en un point observable.

### 2. Routine d'interruption :

A chaque interruption est associée un traitement ou une routine d'interruption. Chaque routine commence par déterminer la cause afin de lui associer le traitement approprié.

A toute interruption correspond donc un code ou procédure de traitement de la cause d'interruption. Une interruption est provoquée pour demander à l'unité centrale d'effectuer une tâche ou traitement donné. Le code de traitement de la cause d'une interruption est appelé routine. Donc, à chaque cause d'interruption correspond un signal d'interruption et une routine d'interruption.

Lorsqu'un périphérique veut s'adresser au microprocesseur, il provoque une interruption matérielle. Ce qui engendre un arrêt de l'exécution séquentielle des instructions d'un programme. Cette interruption provoque l'exécution d'une routine d'interruption, c'est-à-dire d'un programme de traitement de l'événement survenu et propre à cet événement.

Les signaux d'interruption permettent aux différents composants du système, d'attirer l'attention du microprocesseur afin de déclencher une opération déterminée. Comme plusieurs demandes d'interruption peuvent survenir simultanément, elles sont tout d'abord envoyées au contrôleur d'interruption (PIC : Programmable Interrupt Controller) qui les retransmet ensuite au microprocesseur.

### 3. Causes d'une interruption :

Les événements qui peuvent provoquer une interruption sont de deux (2) catégories :

- Les événements causés par le processus actif (événement synchrone).
  - ➔ provoquent des interruptions dites logicielles
- Les événements extérieurs au processus actif (événement asynchrone).
  - ➔ provoquent des interruptions dites matérielles

Exemples :

Événement synchrones :

- division par zéro
- exécution d'instruction privilégiée ou tout accès mémoire interdit.
- appel à une fonction du S.E.

Événements asynchrones :

- fin d'opération d'E/S
- action de l'opérateur
- signal horloge.

#### 4. Types d'interruptions :

Chaque cause est associée à un niveau d'interruption. Il existe au moins 3 niveaux d'interruption :

##### Les interruptions externes :

dont la cause est extérieur au processus actif (panne, périphérique prêt, erreur E/S, réinitialisation du système)

##### Les déroutements :

situation exceptionnelle ou erreur causé par le processus actif (division par 0, débordement, violation d'accès mémoire,...)

##### Les appels au système (Supervisor Call-SVC) :

Instruction réalisant l'appel d'une fonction du S.E (exemple : demande d'E/S,...)

#### 5. Traitement d'une interruption :

La détection, et éventuellement la reconnaissance du signal d'interruption provoque immédiatement le traitement de l'interruption qui consiste en gros à :

##### 1. arrêter le processus actif

Commutation  
de contexte

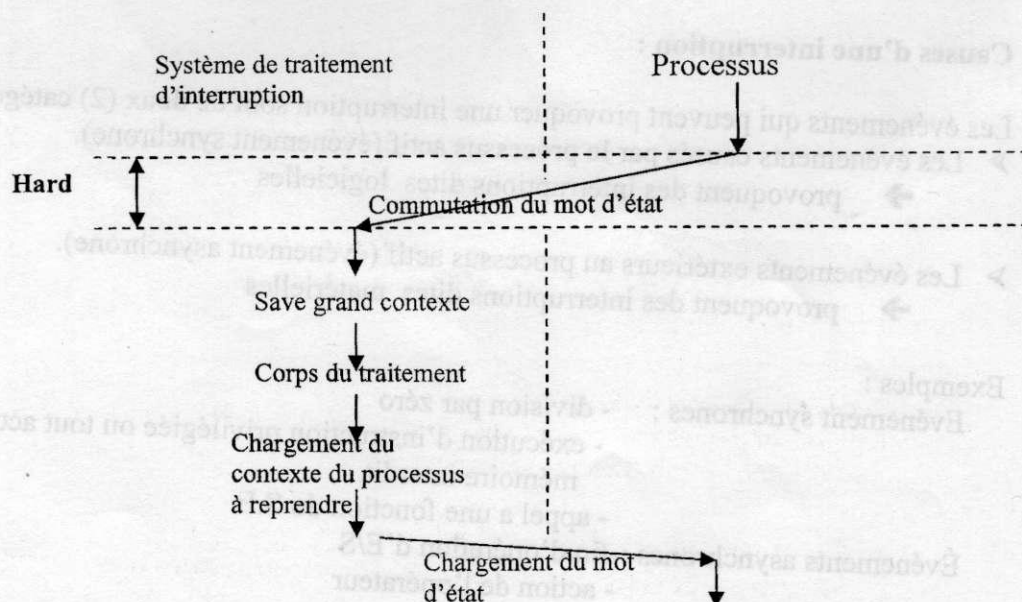
- 2. sauvegarde du contexte du processus
- 3. chargement du contexte programme de traitement de l'interruption

##### 4. exécution du traitement

##### 5. restauration de contexte du processus

##### 6. reprendre l'exécution du processus interrompu

**Remarque :** Il se peut qu'après une interruption le processus à exécuter juste après le traitement de l'interruption ne soit pas celui qui a été interrompu (un processus plus prioritaire est arrivé entre temps)



**Figure1 :** prise en compte d'une interruption

L'opération de commutation de contexte (figure 1) consiste donc à sauvegarder le contexte du processus à interrompre et à le remplacer par le contexte du ~~niveau~~ <sup>nouvel</sup> processus à exécuter. Cette commutation se fait en deux phases :

**Phase1 :**

- commutation du PSW (petit contexte)
- sauvegarde de la valeur du PSW dans une zone précise.
- Chargement d'une nouvelle valeur pour le PSW à partir d'une zone précise.

Remarque : le CO fait parti du PSW en général.

La commutation du PSW est une opération **INDIVISIBLE**.

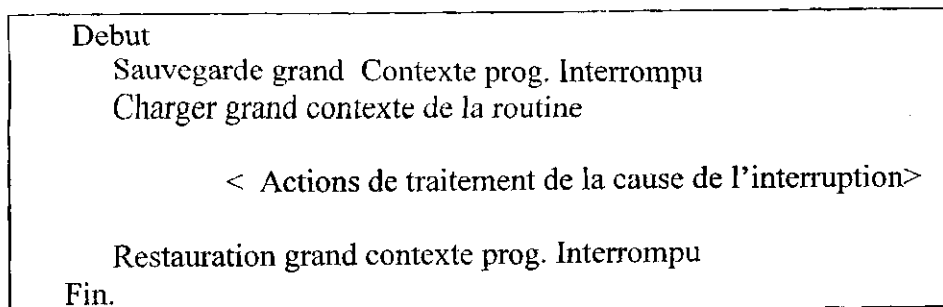
**Phase2 :**

- Les informations qui reste du contexte (grand contexte) peuvent être sauvegardées par le nouveau processus (qui risque de les modifier sinon).
- Cette tâche sera la toute première tâche du nouveau processus qui devient **actif**.

Puisque toutes les informations concernant le processus interrompu on été sauvegardées le dernier pourra reprendre ultérieurement son exécution au point où il l'avait arrêté. Il suffit pour cela de recharger son contexte par une nouvelle opération de commutation de contexte (on parle aussi de restauration de contexte).

Le contexte d'un processus est défini à un point observable (interruptible) où le contexte d'un processus contient des informations stables

A chaque cause d'interruption est associé un code dit Routine d'interruption. Ce code a la forme suivante :



## 6. Système d'interruption hiérarchisée

La plupart des S.E dépose d'un système d'interruption à niveaux de priorité. Ceci permet de répondre à deux problèmes :

- Arrivée simultanée de plusieurs signaux d'interruption (lors de l'exécution d'une instruction)
- Arrivée d'un signal d'interruption pendant le traitement d'une autre.

Les niveaux correspondent à des priorités différentes. Dans chaque niveau, on peut trouver un certain nombre d'interruption. Les priorités sont associées aux différentes causes d'interruption.

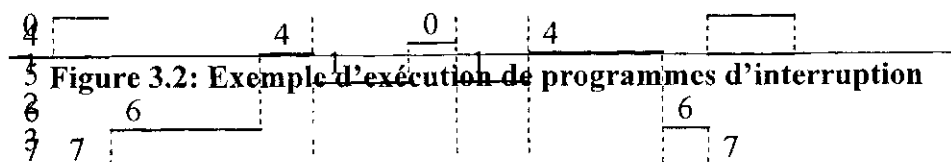


Figure 3.2: Exemple d'exécution de programmes d'interruption

## 7. Masquer / Désarmer une interruption

Le système de traitement qu'on vient de présenter peut être désastreux pour l'exécution des processus si des interruptions arrivent continuellement. Il devient alors nécessaire de pouvoir masquer ou désarmer des niveaux d'interruption.

- ❖ **Masquer** une interruption revient à retarder sa prise en compte. Ceci se fait en positionnant un indicateur dans le PSW. Cet indicateur est consulté avant la prise en compte de l'interruption. On appelle masque d'interruption la suite des valeurs à un moment donné de ces indicateurs.

Le masque d'interruption étant une partie du mot d'état, il peut être modifié lors des commutations de mot d'état causé par une interruption.

- ❖ **Désarmer** un niveau d'interruption → une interruption de niveau est supprimée ou ignorée. On peut aussi réarmer un niveau d'interruption pour prendre en compte les interruptions de ce niveau. Il s'agit encore d'un indicateur à positionner dans le PSW.

Remarque : il est clair qu'on ne peut masquer un déroutement ou la nature de l'interruption liée à l'instruction en cours. Mais on peut le désarmer.

## 8. Activation/Désactivation du S.I :

Dans certains cas, la CPU a besoin d'interdire toute interruption ceci revient à désactiver du S.I global.

Dans ce cas toute interruption est retardée à la prochaine activation du S.I (Système d'Interruption).

Nous avons tenté de faire un tour d'horizon des principaux concepts utilisés pour les systèmes d'interruptions. Mais en fait chaque conception d'ordinateur possède son propre mécanisme d'interruption.