

Systèmes d'Exploitation - ENSIN6U3

Interruptions

Leonardo Brenner ¹ Jean-Luc Massat ²

¹Leonardo.Brenner@univ-amu.fr

²Jean-Luc.Massat@univ-amu.fr

Aix-Marseille Université
Faculté des Sciences

- 1 Rendez-vous asynchrones
- 2 Interruption d'un processus
- 3 Interruption matérielle
- 4 Les appels systèmes
- 5 Déroutements

Table de matière

- 1 Rendez-vous asynchrones
- 2 Interruption d'un processus
- 3 Interruption matérielle
- 4 Les appels systèmes
- 5 Déroutements

Rendez-vous asynchrones

Problème

Prise en compte rapide d'un événement tout en effectuant un travail.

Solution

L'*attente active*. Soit F un indicateur qui signale l'événement

```
...  
si (F = 1) alors  
    F := 0  
    <traiter l'événement>  
fin si  
...
```

Problèmes de la solution

- le test **explicite** est gênant et peu sécurisé,
- perte de temps C.P.U.,
- pas de réaction immédiate et risque de collision.

Table de matière

- 1 Rendez-vous asynchrones
- 2 Interruption d'un processus**
- 3 Interruption matérielle
- 4 Les appels systèmes
- 5 Déroutements

Interruption d'un processus

Interruption

Opération **indivisible** effectuée par la C.P.U. qui change le CO et le MODE. Les interruptions sont déclenchées uniquement sur les points observables (points **interruptibles**).

Vecteur d'interruptions

Chaque interruption a une cause identifiée par un entier. Le **vecteur d'interruptions** (VI) est une table logée dans les adresses basses de la mémoire.

Interruption d'un processus

Prise en compte d'une interruption k

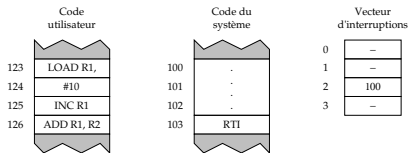
interruption de cause k

```
PUSH CO, MODE;  
MODE := maître;  
CO := vi[k];
```

reprise après interruption

```
POP MODE;  
POP CO;
```

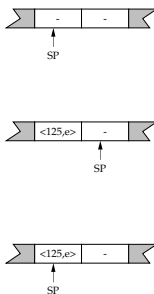
Exemple de traitement d'une interruption



Evolution des registres de la C.P.U.

CO	MODE	SP	R1	R2
123	esclave	200	-	34
LOAD R1, #10				
125	esclave	200	10	34
< interruption de cause n° 2 >				
100	maître	201	10	34
.
103	maître	201	10	34
RTI				
125	esclave	200	10	34
INC R1				
126	esclave	200	11	34

Evolution de la pile



Structure générale d'un traitant

Programme traitant

Le programme exécuté après une interruption est appelé le **traitant** de cette interruption.

Structure d'un traitant

Les traitants ont la structure suivante :

1. sauvegarde du contexte ;
2. traitement de la cause ;
3. restauration du contexte ;
4. retour au processus interrompu (instruction RTI).

Utilisation des interruptions

Types d'interruptions

Il existe trois types d'interruptions :

- **Interruption matérielle** : réaction aux événements extérieurs ;
- **Appel au superviseur** : appel explicite d'une routine système ;
- **Déroutement** : traitement des erreurs et des situations anormales.

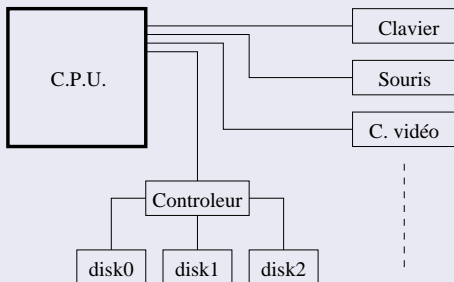
Table de matière

- 1 Rendez-vous asynchrones
- 2 Interruption d'un processus
- 3 Interruption matérielle**
- 4 Les appels systèmes
- 5 Déroutements

Interruption matérielle

Principe

Un signal (électrique) est émis vers le processeur qui provoque une interruption.

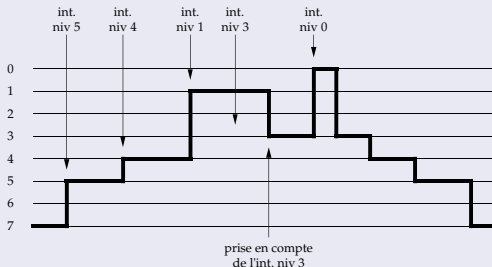


Il y a donc un branchement vers le programme de traitement des interruptions (« le traitant ») avec un **changement de mode**.

Interruption matérielle

Caractéristiques

Le numéro associé à chaque interruption matérielle indique une priorité :



Des instructions privilégiées permettent de :

- masquer des niveaux,
- armer ou désarmer des niveaux,
- déclencher des niveaux.

Table de matière

- 1 Rendez-vous asynchrones
- 2 Interruption d'un processus
- 3 Interruption matérielle
- 4 Les appels systèmes**
- 5 Déroutements

Les appels systèmes

Principe

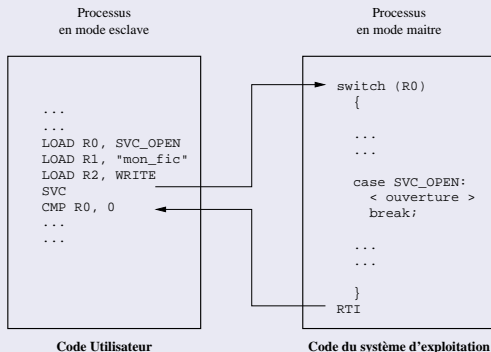
Si les processus s'exécutent en mode « **esclave** », les E/S directes leurs sont interdites.

- Pour réaliser des E/S, les processus doivent envoyer des requêtes au système d'exploitation.
- L'appel de fonction classique pose des problèmes pour réaliser ces requêtes :
 - le nombre de points d'entrée est important,
 - le mode d'exécution reste « esclave »,
 - le code du S.E. est accessible en lecture.

Les appels systèmes

Moyens

Les trappes (« Trap ») : Appel explicite du système par le biais d'une instruction qui déclenche une interruption.



Les appels systèmes

Avantages

Avantages des appels par interruption :

- Changement de mode,
- Le nombre de points d'entrée est limité à un,
- On obtient un sas qui isole le S.E.,
- Définition d'une nouvelle machine (ajout d'instructions),
- Une librairie standard offre :
 - une interface système agréable et simplifiée,
 - une interface indépendante du système d'exploitation,
 - l'implantation réelle des appels systèmes en assembleur.

Les appels systèmes

Structure du traitant

- 1 sauvegarde du contexte,
- 2 vérifier la conformité de la requête,
 - vérifier la nature de la requête,
 - vérifier la nature des arguments,
 - vérifier les droits du processus demandeur.
- 3 exécuter la requête,
- 4 choisir le prochain processus p à exécuter,
- 5 restaurer le contexte de p ,
- 6 relancer le processus p .

Table de matière

- 1 Rendez-vous asynchrones
- 2 Interruption d'un processus
- 3 Interruption matérielle
- 4 Les appels systèmes
- 5 Déroutements**

Déroutements

Principe

L'objectif des déroutements est double :

- traitement **systematique** des erreurs ou des situations anormales (défaut de page),
- **protection** et bonne utilisation de la machine (les processus sont surveillés).

En cas d'erreur

Si l'exécution d'une instruction produit une erreur alors, il y a interruption et branchement vers la routine du système qui traite les erreurs.

Les causes possibles sont :

- données incorrectes (division par zéro),
- opération interdite (instruction privilégiée),
- instruction inconnue,
- accès à une zone mémoire interdite,
- erreur de bus mémoire.

Déroutements

Exemples

- Implantation d'une gestion utilisateur des erreurs,
- Gestion de la mémoire virtuelle,
- Ajout de nouvelles instruction par simulation logicielle,
- Le mode TRACE sur le processeur 68000 :
 - à chaque point interruptible, la C.P.U. génère un déroutement,
 - le S.E. récupère donc le contrôle entre chaque instructions,
 - il peut visualiser l'évolution des registres.