



## Mécanisme d'interruptions sur les processeurs IA32

G.Mounié, S.Nieuviarts, S.Viardot

11/02/08

1



## Plan

- Les interruptions et exceptions
- Le contrôleur d'interruptions
- Le PIT

11/02/08

2



## Interruptions et Exceptions

- Interruption : déclenchée par un autre composant que le processeur
- Exception : déclenchée par le processeur
  - Appel système
  - Erreur d'exécution
- Provoque un déroutement du flot d'exécution

11/02/08

3



## Traitement d'exception

1. Événement cause d'exception
2. Sauvegarde minimale du contexte sur la pile
3. Appel du traitant système
  1. Sauvegarde de registres
  2. Traitement d'exception (si irrécupérable, arrêt du processus ou du système)
  3. Restauration des registres
4. Retour au programme interrompu

11/02/08

4



## Traitement d'interruption

1. Événement signalé au processeur par le contrôleur d'interruptions
2. Dès que le bit **I** de **EFLAGS** passe à **1**, consultation du numéro d'interruption (IRQ) puis traitement comme une exception
3. Au cours de son exécution le traitant envoie un acquittement pour autoriser le signalement de la prochaine IT

11/02/08

5



## Table d'interruptions

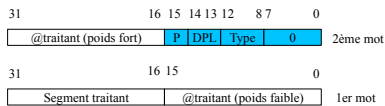
- Table système (IDT) à 256 entrées de (vecteurs) placée à l'adresse 4K
  - 0 à 31 : exceptions du processeur
  - 32 à 47 : correspond aux IRQ (0 à 15)
  - 48 à 255 : libre, utiliser le vecteur **49** pour les appels systèmes de votre noyau

11/02/08

6

## Vecteurs d'interruption

- Appelé aussi descripteur d'IT
- Largeur de 8 octets :  $@ = 8 * no + 4 K$



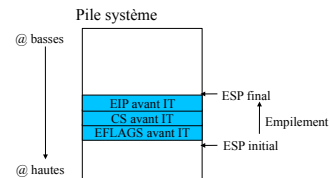
P : 1 si vecteur valide, 0 si vecteur non initialisé  
 DPL : 3 autorise l'appel en mode utilisateur avec INT, sinon 0  
 Type : 14 pour une *Interrupt Gate* ou 15 pour une *Trap Gate*

11/02/08

7

## Appel du traitant

- Sauvegarde et changement de registres
- Description en mode superviseur



EIP courant : adresse de traitant  
 CS courant : segment de traitant  
 EFLAGS courant : bit I selon type de vecteur d'interruption

11/02/08

8

## Fin de traitant

- Effectué par l'appel de l'instruction de retour **IRET**
  - Le processeur dépile **EIP**, **CS** et **EFLAGS** à partir de **ESP**
- L'exécution reprend à partir de l'adresse récupérée
- **EFLAGS** (bit **I** inclus) reprend son ancienne valeur

11/02/08

9

## Masquage d'interruptions

- Les interruptions matérielles peuvent être retardées (masquées) en mettant le bit **I** de **EFLAGS** à 0 (**CLI**)
- Une *Interrupt Gate* masque automatiquement les interruptions
- Pendant l'exécution du noyau il vaut mieux masquer les interruptions pour éviter un état incohérent
- Quand le noyau n'a rien à faire : **STI**

11/02/08

10

## Le contrôleur d'interruptions

- *Programmable Interrupt Controller*
- Mémorise et signale au processeur les interruptions des périphériques
- 15 sources numérotées : 0, 1, 3 à 15
  - Redirigées sur les vecteurs 32 à 47 (vecteur =  $IRQ + 32$ )
- Masquage individuel des sources d'IT
  - Il faut démasquer les composants utilisés

11/02/08

11

## Masque d'IRQ

- Le port d'E/S **0x21** permet de lire ou modifier le masque d'interruption pour les IRQ 0 à 7
  - Bit **N** pour l'IRQ **N**
  - Un bit armé interdit l'IRQ
- Le port **0xA1** contrôle les IRQ 8 à 15
- La modification doit se faire par lecture, modification du bit et écriture

11/02/08

12

## Acquittement d'interruption

- Pour autoriser une nouvelle interruption, écrire au PIC la commande **0x20** sur le port **0x20**
  - Sans acquittement, les interruptions suivantes ne peuvent plus être reçues
- L'acquittement peut se faire en début de traitement si les IT sont masquées

11/02/08

13

## Fonctionnement du PIT

- Un quartz à 1.19 MHz (**0x1234DD** Hz) contrôle plusieurs canaux
- Événements générés de fréquences programmables
  - Un compteur par canal initialisé à une valeur choisie par configuration
  - Compteurs décrémentés à chaque cycle
  - Un compteur à zéro signale un événement

11/02/08

14

## Programmation du PIT

- Port de commande en écriture seulement (**outb**) : **0x43**
- Un port de donnée par canal
  - **0x40** pour le canal 0 qui est utilisé dans le projet
- Programmation en deux temps
  - Écrire un ordre sur le port de commande
  - Lire ou écrire le paramètre correspondant sur le port de donnée

11/02/08

15

## Réglage de fréquence

- Pour générer une IRQ 0 (vecteur 32) à une fréquence choisie **HZ** :
  - Calculer la période :  $P = 0x1234DD / HZ$
  - Envoyer la commande **0x34** (canal 0, poids faible puis poids fort, interruption)
  - Envoyer  $P \% 256$  puis  $P / 256$  sur le port de donnée du canal 0
- La fréquence est une constante du noyau : **CLOCKFREQ**

11/02/08

16

## Utilisation dans le TP PSE

- Initialiser le vecteur d'interruption 32 (pour l'IRQ 0) avec un traitant
- Programmer le canal 0 du PIT à la fréquence désirée
- Démasquer l'IRQ 0 au niveau du PIC (inutile de masquer par la suite)
- Démasquer les IT (**STI**)

11/02/08

17

## Rôle que peut jouer l'IT Timer dans un système

1. Compter le temps
  - Pour réveiller les processus endormis
  - La finesse de comptage dépend de **CLOCKFREQ**
2. Assurer le respect du quantum
  - Changer de processus actif
  - La fréquence dépend de **SCHEDFREQ**
  - Changement de processus actif à la fin de chaque quantum

11/02/08

18