

Projet de Conception de Systèmes d'Exploitation

<http://chamilo2.grenet.fr/inp/courses/ENSIMAG4MMPCSEF/document/projet.html>

ISI-1 : Christophe RIPPERT

ISI-2 : Sébastien VIARDOT

ISSC : François BROQUEDIS

SLE : Matthieu BRIEDA

Prenom.Nom@Grenoble-INP.fr

Bilan séance 1

- **2 canaux de communication avec les périphériques (ici : la carte CGA)**
 - **Ports d'entrées-sorties :**
 - Instructions in et out
 - Envoi d'une commande puis lecture ou envoi d'une donnée
 - **Couplage mémoire (*memory mapping*) :**
 - Une plage d'adresses dédiées en mémoire centrale sert d'espace de communication
 - Les périphériques surveillent le bus adresse pour intercepter les échanges de données
 - **Projet MMIPS 1A : leds, 7-segments, etc.**

Cette séance

- **Objectif : afficher le nombre de secondes écoulées depuis le démarrage (*uptime*)**
- **Utilisation de l'horloge matérielle pour gérer la notion de temps dans le noyau**
- **L'architecture x86 intègre une horloge (*Programmable Interval Timer*) qui génère des signaux selon la fréquence voulue**
- **Le PIT envoie un signal via un canal appelé IRQ (*Interrupt Request*) qui déclenche une interruption**

Rappel : interruptions

- **Interruption (IT) : signal (externe ou interne au processeur) qui en interrompt le fonctionnement normal :**
 1. Le processeur exécute un programme
 2. IT → le processeur arrête l'exécution du programme et sauvegarde l'état courant
 3. Il passe la main à un programme appelé traitant d'interruption (*Interrupt Handler*)
 4. A la fin du traitant, le processeur reprend l'exécution du programme initial là où il s'était arrêté (gestion transparente de l'IT)

Interruptions sur l'ia32

- **Masquage / démasquage des IT :**
 - Une IT masquée est ignorée
 - Instructions (fonctions C) : `cli` / `sti`
- **IT générée par le contrôleur d'IT**
(*Programmable Interrupt Controller*)
 - Le PIC reçoit une IRQ (horloge = IRQ 0)
 - Si l'IRQ est démasquée, le PIC envoie l'IT correspondante au processeur (horloge = IT 32)
 - Attention : masquage IT \neq masquage IRQ
- **Adresse du traitant d'IT :**
 - Table des vecteurs d'interruption (*Interrupt Vector Table*) à une adresse connue (ici : 0x1000)

Etapes à réaliser

1. Fonction C qui affiche l'*uptime*

- Position fixe => fonction dédiée (\neq `printf`)

2. Ecrire le traitant de l'IT32 :

- Partie en ASM (sauvegarde / restauration de l'état) donnée (il faut la comprendre)
- Partie en C à écrire (i.e. appel de la fct 1.)

3. Initialiser l'IVT avec l'adresse du traitant

4. Régler la fréquence du PIT (`in` / `out`)

5. Démasquer l'IRQ 0 puis démasquer les IT :

⇒ Ça marche : 😊

⇒ Ça ne fait rien : démasquage ? réglage PIT ?

⇒ Boom : GDB pour localiser les erreurs