Projet : développement d'un mini système d'exploitation pour PC x86

Le Timer

Jérôme Ermont et Emmanuel Chaput

IRIT - Toulouse INP/ENSEEIHT

PIT : Programmable Interval Timer

- Aussi appelé 8253/8254
- Timer matériel qui comprend :
  - un oscillateur de 1,19 MHz
  - un pré-scaleur
  - 3 ports de sortie

Ports

- Disposent d'un diviseur d'horloge
- Channel 0 : connecté à l'entrée 0 du PIC
- Channel 1 : permet de gérer la mémoire
- Channel 2 : utilisé par le haut parleur du PC
- Configurés par un registre de contrôle/commande

Num. Port	Utilisé pour
0x40	Channel 0
0x41	Channel 1
0x42	Channel 2
0x43	Registre de contrôle/commande

Registre de contrôle/commande

Bits	Signification
7 et 6	Selection du Channel 0 0 : Channel 0 0 1 : Channel 1 1 0 : Channel 2 1 1 : Utilisé pour la lecture du registre de contrôle/commande
4 et 5	Mode d'accès 0 0 : Mode enregistrement valeur 0 1 : Poids faible seul 1 0 : Poids fort seul 1 1 : Poids faible/Poids fort
de 1 à 3	Mode de fonctionnement 0 0 0 : Interruption sur compte à rebours 0 0 1 : Compte à rebours contrôlé 0 1 0 : Générateur d'impulsion 0 1 1 : Générateur de signal carré 1 0 0 : Impulsion à expiration d'un timer 1 0 1 : Idem 1 1 0 : Générateur d'impulsion 2 1 1 1 : Générateur de signal carré 2
0	Mode BCD (1)/Binaire (0)

Exemple de configuration

1. outb(0x34, 0x43)
   Channel 0, accès poids faible/poids fort, générateur d'impulsion, fréquence définie en binaire
2. outb(FREQUENCE&0xFF, 0x40)
   Affectation de la fréquence, poids faible, au Channel 0
3. outb(FREQUENCE>>8, 0x40)
   Affectation de la fréquence, poids faible, au Channel 0

- Calcul de la fréquence
  - Oscillateur : f_osc=1,19 MHz (0x1234BD)
  - FREQUENCE = f_osc/HORLOGE
  - HORLOGE : fréquence de l'horloge à générer

PIC : Programmable Interrupt Controller

- Permet de géréer les interruptions provenant de différents matériels (Timer, Clavier, ...)
- Table de correspondance

Port d'entrée	Numéro d'IT	Description
IRQ0	0x20	Timer
IRQ1	0x21	Clavier
IRQ2	0x22	Cascade pour le PIC esclave
IRQ3	0x23	Port série 2
IRQ4	0x24	Port série 1
IRQ5	0x25	Port parallèle 2
IRQ6	0x26	Lecteur de disquette
IRQ7	0x27	Port parallèle 1
IRQ8/IRQ0	0x28	CMOS RTC
IRQ9/IRQ1	0x29	CGA
IRQ10/IRQ2	0x2A	Reservé
IRQ11/IRQ3	0x2B	Reservé
IRQ12/IRQ4	0x2C	PS/2
IRQ13/IRQ5	0x2D	FPU
IRQ14/IRQ6	0x2E	Controleur de disque dur
IRQ15/IRQ7	0x2F	Reservé

Gestion des interruptions via le PIC

- Le PIC dispose de 2 ports I/O pour le contrôler
  - 0x20 : Commande
  - 0x21 : Données de configuration
- Masquage de l'IT :
  outb(inb(NumPortPIC)|(1<<NumPortIRQ), NumPortPIC)
  Par ex : outb(inb(0x21)|1, 0x21) : désactivation de l'IT du Timer
- Démasquage de l'IT :
  outb(inb(NumPortPIC)&~(1<<NumPortIRQ), NumPortPIC)
  Par ex : outb(inb(0x21)&0xfe, 0x21) : activation de l'IT du Timer
- Acquittement de l'IT :
  outb(0x20, NumPortPIC)
  Par ex : outb(0x20, 0x20) : ack de l'IT du PIC Master

Travail à réaliser

Écrire le module `time.c` qui permet de gérer l'horloge système à 1 kHz (1 ms) et qui permet de faire la conversion vers les heures, minutes et secondes.

1. Écrire la fonction qui initialise le timer pour la fréquence de 1 kHz et la ligne d'IT correspondante (penser à avoir un fichier assembleur de traitement de l'IT) ;
2. Écrire en C la fonction de traitement de l'IT reçue. Cette fonction acquitte l'IT et incrémente le compteur système.
3. Écrire une fonction de conversion du compteur système vers un format hh :mm :ss
4. Afficher la durée d'exécution du système en haut et à droite de la console.