

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

Le Timer

Jérôme Ermont et Emmanuel Chaput

IRIT - Toulouse INP/ENSEEIH



- Le principe de lecture d'un caractère est simple
 - ① L'utilisateur appuie sur une touche et génère une interruption clavier
 - ② Le code de la tâche est lue
 - ③ On convertit le code de la touche appuyée en caractère
 - ④ Le caractère obtenu est stocké dans un buffer, prêt à être lu par un appel à `kgetch`

- 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	Controlleur de disque dur
IRQ15/IRQ7	0x2F	Reservé

- 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
- Acquiescement de l'IT :
`outb(0x20, NumPortPIC)`
Par ex : `outb(0x20, 0x20)` : ack de l'IT du PIC Master

- La gestion des interruptions clavier est faite par le PIC
 - Interruption numéro 0x21
 - Ligne 1 du PIC
 - Une interruption est générée lorsqu'une touche est appuyée ou relâchée
- Le port I/O 0x60 permet de récupérer le code de la touche appuyée ou relâchée
 - Le bit 7 du code permet de savoir si la touche a été appuyée (bit 7 = 0) ou relâchée (bit 7 = 1)
 - Une table est nécessaire pour convertir le code de la touche avec le caractère. Par exemple, la touche 0x10 correspond au caractère 'a'.

- ❶ Écrire une fonction d'initialisation du clavier : `init_keyboard`
- ❷ Écrire et gérer le traitement d'une interruption clavier (pour la lecture des caractères)
- ❸ Écrire une fonction `kgetch` qui retourne un caractère stocké dans un buffer
 - Le buffer peut être réalisé à l'aide d'un buffer circulant de taille connue.