

Pratique du système : Mécanisme d'entrées-sorties sur les architectures IA32

09/16/08

1

Plan

- Entrées / Sorties
- Fonctionnement de l'affichage

09/16/08

2

Espace mémoire d'Entrées/Sorties

- Ports d'E/S
 - Adressage séparé pour transférer des données sur/depus des périphériques d'E/S
 - Ports sont
 - des canaux de communication pour lire, écrire ou les deux (canaux bidirectionnels)
 - *hardware wired*
- Espace d'adressage d'E/S
 - Distinct et séparé de l'espace d'adresse mémoire classique
 - Espace de 64K entre 0x0 et 0xFFFF par ports de 8 bits
- Instructions d'utilisation
 - `in` et `out`
 - ne pas utiliser `mov`, cela ne marche pas

09/16/08

3

Exemple d'utilisation des ports d'E/S

- `in` : écrit dans un registre le contenu du port
 - Valeur immédiate
 - 8 bits : `inb <valeur immédiate (@ du port)>, %al`
 - 16 bits : `inw <valeur immédiate (@ du port)>, %ax`
 - 32 bits : `inl <valeur immédiate (@ du port)>, %eax`
 - La valeur du port concerné est dans dx
 - 8 bits : `inb %dx, %eax`
 - etc...
- `out` : écrit dans le port la valeur d'un registre
 - Valeur immédiate
 - 8 bits : `outb %al, <valeur immédiate>`
 - etc...
 - La valeur du port concerné est dans dx
 - 8 bits : `outb %al, %dx`
 - etc...

09/16/08

4

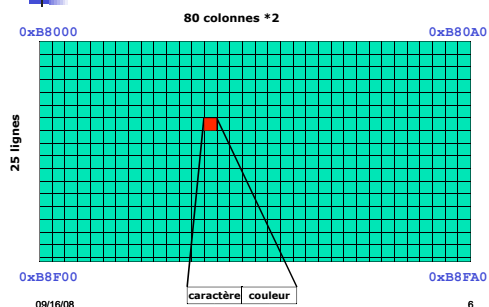
Affichage

- Utilisation synchrone
 - On place dans un endroit particulier de la mémoire (vidéo) des caractères
 - Utilisation du mode texte VGA
 - Ecriture caractère par caractère
 - Caractère = 8 bits
 - Attributs de couleur
 - Couleur du caractère et/ou du fond (8 bits)
- | | | | | | | |
|------|-------------------|---|---|--------------------|---|--|
| Bits | 7 | 6 | 4 | 3 | 0 | |
| Plan | Bits couleur fond | | | Bits couleur texte | | |
- 2 octets associés à un caractère à afficher
 - Positionnement des caractères sur l'écran
 - Matrice 80 colonnes x 25 lignes 'mappée' en mémoire
 - Début à l'adresse mémoire 0xB8000

09/16/08

5

Positionnement sur l'écran



09/16/08

6

Contrôleur Vidéo Standard : CRT MC6845

- Ports d'utilisation
 - Sélection des registres du CRT
 - Port 0x3d4
 - Écriture/Lecture dans le Registre sélectionné
 - Port 0x3d5
- Registres du CRT
 - 18 registres permettant de positionner certaines caractéristiques de l'écran
 - Nb de caractères, synchronisation horiz/verticale,
 - 2 Registres pour gérer la position du curseur
 - Registres 0x0E et 0x0F

09/16/08 7

Gestion du curseur

Position du curseur : $POS = x + (y * 80)$

Poids faible (POS) Poids fort (POS)

Reg. 0x0E Reg. 0x0F

09/16/08 8

Positionnement du curseur

```

#define NB_COLUMN 80
int pos, x, y;

pos = x + y*NB_COLUMN;
outb(0x0F, 0x3D4);
outb((char)pos, 0x3D5);
outb(0x0E, 0x3D4);
outb((char)(pos >> 8), 0x3D5);

```

09/16/08 9

Gestion du défilement

Copie de la ligne (caractère et couleur)

Itération du processus Et caractères invisibles sur toute la dernière ligne

09/16/08 10