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

Jérôme Ermont et Emmanuel Chaput

IRIT - Toulouse INP/ENSEEIHT



L'écran

- L'écran est composé de 25 lignes de 80 colonnes
 - Zone mémoire qui commence à l'adresse 0xB8000
 - Tableau de 25× 80 cases
- Chaque case est composée de 2 octets : le caractère à afficher et ses attributs (cli = clignotage, couleurs de fond et de texte)

Format d'affichage						Caractère									
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
cli.	fond			texte				code ascii							

Les couleurs

• 16 couleurs sont possibles

Num.	Couleur	Num.	Couleur		
0	noir	8	gris foncé		
1	bleu	9	bleu clair		
2	vert	10	vert clair		
3	cyan	11	cyan clair		
4	rouge	12	rouge clair		
5	magenta	13	magenta clair		
6	marron	14	jaune		
7	gris	15	blanc		

Affichage

- Position à l'écran : $0xB8000 + (80 \times ligne + colonne) \times 2$
- 2 façons d'afficher un caractère à l'écran :
 - Écran = tableau d'octets, exemple écriture de A à la position (3, 10) :

```
uint8_t *scr_tab= (uint8_t *) 0xB8000;
int pos= (80*10+3)*2;
scr_tab[pos]= 'A';
scr_tab[pos+1]= 0x0F; // fond noir; texte blanc
```

• Écran = tableau de 2 octets, même exemple :

Le curseur

- Le curseur est géré par la carte graphique
- Gestion à l'aide de 2 ports E/S : 0x3D4 (commande) et 0x3D5 (valeur)
- Utilisation des commandes out du processeur Intel
 - void outb(uint8 val, uint8 port); de n70S/cpu.h
 - pos du curseur : 80 × ligne + colonne
 - envoi de la position en 4 étapes
 - 1 envoi de la commande 15 (0xF)
 - 2 envoi de la valeur du poids faible de la position
 - 3 envoi de la commande 14 (0xE)
 - envoi de la valeur du poids fort

La console

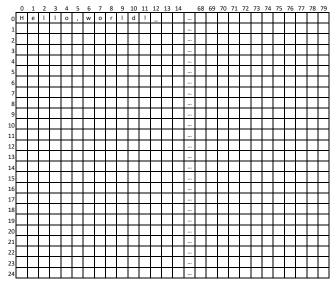
- void console_putbytes(const char *s, int len);
 écrit la chaine de caractères à l'écran
 - utilisée par la fonction printf (pour le moment)
- seuls les caractères compris entre 32 et 127 seront affichés
- quelques caractères de contrôle à prendre en compte

Code	Commande	Car. C	Action
8	Backspace (BS)	'\b'	Déplace le curseur d'une colonne en arrière
9	Horizontal Tab (HT)	'\t'	Ajoute un espace de 8 caractères
10	Line Feed (LF)	'∖n'	Déplace le curseur à la ligne suivante, colone 0
12	Form Feed(FF)	'\f'	Efface l'écran et revient à la colonne 0 de la ligne 0
13	Carriage Return (CR)	'\r'	Déplace le curseur à la colonne 0 de la ligne courante

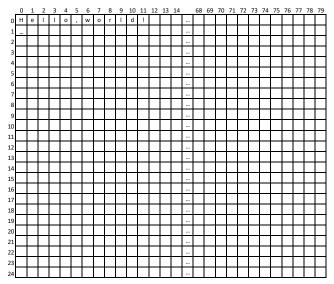
Affichage d'un caractère

```
void console_putchar(const char c) {
   if (c > 31 && c < 127)(
        // affiche le caractere a l'ecran
   } else if (c == '\n') { // ou (c == 10)
        // curseur au debut de la ligne suivante
   } else ...
   // et ainsi de suite
}</pre>
```

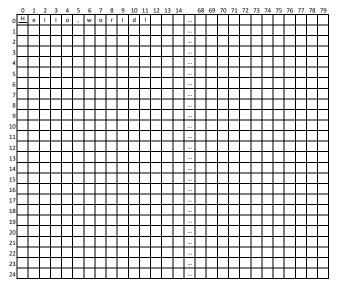
console_bytes("Hello, world!", 12);



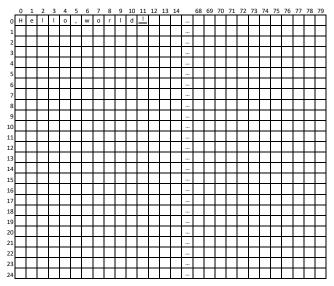
console_bytes("Hello, world!\n", 13);



• console_bytes("Hello,world!\r", 13);



console_bytes("Hello, world!\b", 13);



Travail à faire

- Modifier le fichier console.c pour :
 - 4 ajouter une fonction qui permet de position le curseur,
 - 2 ajouter une fonction console_putchar qui affiche le caractère passé en argument ou demande le déplacement du curseur,
 - ompléter la fonction console_putbytes.
- Tester votre code en affichant des caractères à l'écran (via printf)
 - tester les caractères de contrôle.