

Programmation des Systèmes

TP 3 : Initiation à l'assembleur

1 But

Le but de ce TP est de vous familiariser avec l'assembleur ARM. Il vous faudra coder plusieurs fonctions d'affichage en assembleur, à savoir :

- Remplissage d'un rectangle
- Dessin d'un cercle
- Affichage d'une ligne

2 Travail à faire

Le mode d'affichage que nous allons utiliser est le mode 3 :

- La taille de l'écran fait 240x160 pixels.
- Un pixel fait 2 octets.
- Chaque pixel a une couleur codée sur 16 bits (5 bits par plan de couleur, Blue Green Red).

2.1 Affichage d'un rectangle en assembleur

La première partie du TP consiste à remplir un rectangle en assembleur. Pour ce faire, dans le fichier *main.c*, vous devez définir une fonction `fillRect` qui va remplir un rectangle. Voici la signature de la procédure

```
extern void fillRect(  
    unsigned int x,  
    unsigned int y,  
    unsigned int width,  
    unsigned int height,  
    unsigned int color);
```

Dans cette signature, le doublon `(x,y)` définit la position du sommet supérieur gauche du rectangle à l'écran, `width` définit sa largeur, `height` est sa hauteur, et `color` est la couleur de remplissage.

Attention : Les quatre premiers paramètres d'une fonction sont dans les registres `r0-r3` et les autres se trouvent au sommet de la pile.

2.2 Dessin d'un contour de cercle

La seconde partie consiste à remplir un cercle de rayon `radius`, de position `x,y` et de couleur `color`. La signature sera donc :

```
extern void drawCircle(  
    unsigned int x,  
    unsigned int y,  
    unsigned int radius,  
    unsigned int color);
```

Pour dessiner le cercle, utilisez l'algorithme de Bresenham.

2.3 Affichage d'une droite

La dernière partie du travail consiste à afficher un segment reliant `x1,y1` et `x2,y2` de couleur `color`. La signature de cette fonction sera :

```
extern void drawLine(  
    unsigned int x1,  
    unsigned int y1,  
    unsigned int x2,  
    unsigned int y2,  
    unsigned int color);
```

Pour afficher la droite, nous vous demandons d'utiliser l'algorithme de Bresenham (cf. indications).

3 Indications

- Les quatre premiers paramètres d'une fonction sont dans `r0-r3`, les autres sont au sommet de la pile.
- La pile utilisée par C est descendante.
- L'assembleur peut effectuer des opérations sous condition, n'hésitez pas les utiliser, cela peut simplifier le code.
- L'assembleur est très peu lisible, n'hésitez pas à beaucoup (trop) commenter le code.
- Bresenham a proposé plusieurs algorithmes pour dessiner des segments ou des cercles. Sur Internet, vous trouverez plusieurs explications sur ses algorithmes.

4 À rendre

Votre code source.