

TD 7

Du matériel au logiciel

Objectif : organisation en couches - bibliothèques

On veut créer une bibliothèque qui permet de faire des dessins sur un terminal alphanumérique. On travaillera sur la console Bash ou la console tcsh, qui sont tous les deux conformes à l'API des terminaux X. Les fonctions de la bibliothèque sont organisées en couches. Par exemple, les fonctions de base de l'API des terminaux X sont dans la couche 1 (la plus basse), qui vous est fournie dans TPInfo.

```
/* -----  
 * Fichier.....: couche1.c  
 * ----- */  
#define ESC 27  
#include <stdio.h>  
  
void up(void) {  
    printf("%c[%dA",ESC,1);  
} /* fin de up()*/  
  
void down() {  
    printf("%c[%dB",ESC,1);  
} /* fin de down */  
  
void left() {  
    printf("%c[%dD",ESC,1);  
} /* fin de left() */  
  
void right() {  
    printf("%c[%dC",ESC,1);  
} /* fin de right() */  
  
void gotopos(int x, int y) {  
    printf("%c[%d;%dH",27,y,x);  
} /* fin de gotopos() */  
  
int writechar(int c) {  
    int ret ;  
    ret = putchar(c) ;  
    left() ;  
    return(ret) ;  
} /* fin de writechar() */
```

1. Compiler `couche1.c` pour générer un fichier objet `couche1.o`. Écrire un programme `test1.c` qui déclare les fonctions de la couche 1, puis lit en boucle un caractère au clavier :

- Si le caractère est 'u', 'd', 'l', ou 'r', test appelle la fonction de la couche 1 qui commence par cette lettre.
- Si c'est 'w', il lit écrit le caractère '*' avec `writechar`
- Si c'est 'g', il retourne à la position (40,20)
- Si c'est 'q', il se termine.
- Les autres caractères sont ignorés

Compiler `test1.c` pour générer l'exécutable `test1`.

Que fait chacune des 5 fonctions de la couche 1 ?

2. Dans la couche 2, on écrit les fonctions qui produisent un déplacement du curseur en ligne droite (horizontale ou verticale) et celles qui dessinent un trait. L'API est décrite par les règles suivantes :

- Les noms des fonctions de déplacement sont formés avec *mov*. Par exemple,
`void mov_u(int n) ;`
est le prototype de la fonction qui déplace le curseur de *n* positions vers le haut (*up*).
- Les noms des fonctions de dessin de droite sont formés avec *line*. Par exemple,
`void line_l(int n) ;`

est le prototype de la fonction qui dessine n caractères '-' vers la gauche (left). Les droites verticales utilisent '|', les horizontales utilisent '-', les obliques utilisent '/' ou '\' (celui qui correspond à leur direction).

- Toutes les fonctions de dessins de droite respectent la spécification suivante : le premier caractère est dessiné à l'emplacement du curseur avant l'appel, et le curseur est laissé juste après le dernier caractère *dans la direction où on a tracé*. Ainsi, 2 appels `line_l(n)` ont le même résultat qu'un appel `line_l(2n)`.
 - Les directions de déplacement sont u (up), d (down), l (left), r (right). Les directions de dessin sont les 4 précédentes, plus ne (vers le nord-est), se (sud-est), sw (sud-ouest), nw (nord-ouest).
- a. Écrire les 12 fonctions de la couche 2 dans le fichier `couche2.c` et les compiler. On prendra soin de compiler (en générant `couche2.o`) dès qu'on a écrit la première fonction, puis toute les 2 ou 3 fonctions, pour ne pas avoir trop d'erreurs à la fois.
 - b. Écrire les douze prototypes dans un fichier `couche2.h`
 - c. On veut écrire un programme de test des fonctions de dessin qui appelle les 8 fonctions de dessin dans l'ordre, en écrivant pour chacune au bout le numéro de l'appel puis en revenant (à chaque fois) au point (30,10). Commencez par le `#include` qui vous permet de disposer des prototypes de la couche 2. Écrire le programme, le compiler et l'exécuter. Quel est le résultat si on supprime toutes les instructions `gotopos()` (ajouter // devant les lignes correspondantes).

3. Dans la couche 3, on va rassembler les fonctions qui dessinent des rectangles et des losanges (carrés "sur la pointe"). Écrire le fichier `couche3.c` qui déclare les prototypes nécessaires et contient les fonctions `losange` et `rectangle` dont l'API est :

- les prototypes des fonctions sont `void rectangle(int x, int y)` et `void losange(int n)` spécifiée par :
- le rectangle a x caractères '-' horizontalement et y caractères '|' verticalement, plus une étoile '*' à chaque coin (ses dimensions *intérieures* sont x et y, ses dimensions *extérieures* sont x+2 et y+2).
- de même, le losange a n caractères '/' ou '\' suivant le coté, plus des étoiles '*' à chaque coin.
- les deux fonctions laissent le curseur où il était avant leur exécution.
- Par rapport à ce point de départ et d'arrivée, le dessin du rectangle est en bas à droite, le dessin du losange est en bas.

Compilez `couche3.c` et écrivez un fichier `test3.c` qui dessine un rectangle et un losange.

4. Créez une archive de bibliothèque avec les fichiers `couche1.o`, `couche2.o` et `couche3.o`, et copiez-la dans le répertoire `~/lib`. Mettez toutes les déclarations de prototype dans un seul fichier `formes.h` et copiez-le dans le répertoire `~/include`.

Créez un répertoire `td7essai` et copiez-y `test3.c`. Faut-il modifier `test3.c` pour pouvoir le compiler sans rien copier d'autre dans ce répertoire ? et quelle commande faut-il utiliser pour cela ?