

# Bibliothèques

---

Les objectifs du TP sont les suivants :

- Apprendre à utiliser une bibliothèque C.
- Apprendre à créer une bibliothèque C.

## 1 Header C

Un fichier d'en-tête C, ou *header*, est un fichier portant l'extension `.h` qui contient généralement les prototypes de fonctions définies dans un fichier `.c`. Par exemple voici le fichier `test.h` :

```
1 #ifndef TEST_HPP
2 #define TEST_HPP
3
4 int quaranteDeux();
5
6 #endif
    associé au fichier test.c :
7
8 #include "test.h"
9
10 int quaranteDeux()
11 {
12     return 42;
13 }
14
15 int fact(int n)
16 {
17     return n <= 1 ? 1 : n * fact( n-1 );
18 }
19 }
```

1. Créer un fichier nommé `fcts.c` contenant les fonctions `fact` et `fibonacci` et le fichier en-tête associé.
2. Créer un fichier nommé `main.c` faisant appel à ces fonctions.

## 2 Utilisation d'une bibliothèque existante

La bibliothèque DevIL <http://openil.sourceforge.net/> permet de manipuler des images, le code suivant montre son utilisation :

```
1 #include <IL/il.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     unsigned int image;
```

```

8 // Allocation d'une image de 100x100 pixels en RGB
9 unsigned char* data = (unsigned char*) malloc(100*100*3);
10
11 // Dessin d'une diagonale
12 for(int i = 0 ; i < 100 ; ++i) {
13     data[ 3*(100*i + i) ] = 255;
14     data[ 3*(100*i + i) + 1 ] = 255;
15     data[ 3*(100*i + i) + 2 ] = 255;
16 }
17
18 ilInit(); // Initialisation de la bibliothèque
19 ilGenImages(1, &image); // Génération de l'image
20 ilBindImage(image); // Activation de l'image
21
22 // Définition de l'image : 100x100 pixels, 3 composantes RGB, données
23 iTexImage(100, 100, 1, 3, IL_RGB, IL_UNSIGNED_BYTE, data);
24
25 iEnable(IL_FILE_OVERWRITE); // Ecrase l'image si elle existe déjà
26 iSaveImage("out.jpg"); // Sauvegarde de l'image dans le fichier out.jpg
27 iDeletImages(1, &image); // Libération des ressources liées à l'image
28 free(data); // Libération du tableau image
29
30 return 0;
31 }

```

Afin d'utiliser les fonctions de la bibliothèque il convient d'inclure le header `il.h` contenant la définition des fonctions disponibles de la bibliothèque.

Pour compiler l'exemple : `gcc -o image image.c -lIL -std=c99`. L'option `-lIL` permet de lier le programme au fichier de la bibliothèque nommé `libIL.so` qui doit se trouver dans un emplacement standard tel que `/usr/lib`.

Essayer de dessiner un carré au milieu de l'image.

### 3 Réalisation d'une bibliothèque

1. Créer une bibliothèque statique nommée `libfcts.a` à partir du fichier `fcts.c` précédent.
2. Créer un fichier nommé `main.c` faisant appel aux fonctions de votre bibliothèque et tester son fonctionnement.
3. Créer une bibliothèque dynamique nommée `libfcts.so` à partir du fichier `fcts.c`.
4. Est-il nécessaire de changer le code du fichier `main.c` pour l'utiliser ?

### 4 Plugin

Les plugins sont des bibliothèques dynamiques mais qui sont chargées à la demande. Reprendre la bibliothèque dynamique précédente et utilisez là comme un plugin.