

Micro TP: Gestion de la mémoire

Considérons le code

```
1 #include <stdio.h>
2 #include <math.h>
3 // D'après Marc de Falco
4
5 const int a = 42;
6 int b[] = { 1, 2, 3 };
7 int c;
8
9 int f(int x, int y)
10 {
11     int z = x;
12     z = z * y;
13     return z;
14 }
15
16 int main(int argc, char **argv)
17 {
18     const int d = 1664;
19     // c = f(a, d);
20
21     printf("%f\n", cos(d));
22     return 0;
23 }
```

Faisons-en, un fichier objet : `gcc -c memoire.c -o memoire.o` Dans le fichier objet créé, on ne trouve pas mention de la pile et du tas : ils sont créés automatiquement au lancement du programme.

L'utilitaire `objdump` de LINUX donne des informations sur les fichiers objets.

1. Quelles variables/constantes/fonctions vont dans le segment de texte ?
2. Identifier les variables ou constantes qui vont dans le segment de données (et dans les sous-zones en lecture seule (rodata) ou lecture-écriture (data)).
3. Identifier les variables de la pile. Y a-t-il des variables sur le tas ?
4. Identifier les variables ou constantes du segment BSS.
5. Lancer `objdump -x memoire.o`. L'option `-x` affiche tous les en-tête d'information disponible. Répondre à nouveau aux questions précédentes. Le symbole `*COM*` désigne les variables externes non initialisées.