

# TP Adresses Variables / Fonctions

## 1°) Ecrire le programme suivant :

```
#include <stdio.h>
int i;
int j = 2;
extern int _end;
extern int _etext;
extern int _edata;
extern int __bss_start;
extern char **environ;
void fonc(int val)
{
    int k2;
    static int m;
    printf("val = %d\n", val);
    printf("adr de val = %p\n", &val);
    printf("adr de k2 = %p\n", &k2);
    printf("adr de m = %p\n", &m);
    printf("adr de fonction = %p\n", &fonc);
}
int main(int argc, char *argv[0], char *arge[0])
{
    int k1;
    fonc(5);
    printf("adr de main = %p\n", main);
    printf("adr de _etext = %p\n", &_etext);
    printf("adr de j = %p\n", &j);
    printf("adr de _edata = %p\n", &_edata);
    printf("adr de __bss_start = %p\n", &__bss_start);
    printf("adr de i = %p\n", &i);
    printf("adr de _end = %p\n", &_end);
    printf("adr de i = %p\n", &i);
    printf("adr de k1 = %p\n", &k1);
    printf("adr de premier argument argv[0] = %p\n", &argv[0]);
    printf("adr de la premiere variable d'environnement = %p\n", &arge[0]);
    printf("adre  argc = %p\n", &argc);
    printf("adresse du tableau de pointeur sur les argv = %p\n", argv);
    printf("adresse du tableau de pointeur sur les arge = %p\n", arge);
    for(;;);
    return 0;
}
```

## 2°) Compiler et exécuter le programme

Utiliser g++ pour compiler le programme et générer le programme mesadresses.exe  
Exécuter mesadresses.exe → ./mesadresses.exe  
Quel est le PID de ce programme ?

## 3°) LDD : List Dynamic Dependencies

Utiliser la commande suivante :  
ldd mesadresses.exe

## 4°) maps

Utiliser la commande suivante :  
cat /proc/MONPID/maps

## 5°) Remplissez le schéma ci-dessous :

