

TP n°10 : Gestion de la mémoire

Objectif : La mémoire virtuelle des processus

La mémoire virtuelle d'un processus est constituée d'un ensemble de *régions*. Une région est une portion contiguë de la mémoire virtuelle d'un processus. A chaque région le système associe des protections et un rôle particulier. Le but de cet exercice est de visualiser et de comprendre ces régions.

Travail à effectuer :

1) Commencez par compiler (avec l'option -static) le programme suivant :

```
#include <sys/types.h>
#include <unistd.h>

#include <stdio.h>
#include <stdlib.h>

int une_globale;
int une_autre=4;
char* alloc;

int main()
{
    int une_locale;

    alloc = malloc(1024L * 1024L * 10L); /* 10mo */
    printf("PID = %d\n", getpid());
    printf("adresse de une_globale      = %8lx\n", (unsigned long) &
    une_globale);
    printf("adresse de une_autre_globale = %8lx\n", (unsigned long) & une_autre);
    printf("adresse de une_locale      = %8lx\n", (unsigned long) & une_locale);
    printf("adresse de alloc          = %8lx\n", (unsigned long)  alloc);
    printf("adresse de main           = %8lx\n", (unsigned long) & main);
    printf("adresse de printf        = %8lx\n", (unsigned long) & printf);

    /* afficher la carte mémoire */
    sprintf(alloc, "cat /proc/%d/maps", getpid());
    system(alloc);

    return 0;
}
```

2) En comparant l'espace adressable de chaque région et les adresses données par le programme, donnez un sens à chaque région.

3) Même exercice mais en compilant le programme sans la directive -static.

Remarque : le système UNIX associe un numéro à chaque fichier sur disque. Ce numéro (appelé le numéro d' *i-node*) peut-être visualisé en utilisant l'option -i de la commande ls.