Licence Informatique **Initiation à la Programmation**
Université Bordeaux I **Travaux Dirigés**

Zones de mémoire

- Mettre en évidence les différentes zones de mémoire.
- Mettre en œuvre des pointeurs de pointeur.

Recopier à votre racine le répertoire `~wacren/IP/5_memoire` ainsi que son contenu (commande `cp -r ~wacren/IP/5_memoire ~/IP`).

Exercice 1.

1. Compléter la fonction `strcpy` du fichier `copions.c`.
2. Compiler le programme `copions.c` via la commande `gcc -g -o copions copions.c`.
3. Exécuter le programme, l'exécution avorte sur une erreur de segment.
4. Pour comprendre l'origine de la célèbre expression *core dumped*, entrer la commande shell `ulimit -c unlimited` afin d'autoriser la production de fichier *core* (copie de la mémoire d'un processus réalisée au moment de sa mort brutale) et relancer le programme `copions`.
5. Lancer la commande `gdb copions core` dans un shell, déterminer la cause de l'arrêt brutal du programme (on pourra utiliser la commande `bt full` pour voir l'état de la pile d'appel des fonctions).

Le tandem core + gdb est très efficace pour débugger les programmes dont le comportement est difficilement reproductible – en particulier pour débugger les programmes interactifs.

Exercice 2. Le programme `cartographie.c` affiche les adresses de quelques objets en mémoire.

1. Analyser le résultat de ce programme. On compilera (via `gcc -g cartographie.c`) et exécutera le programme (`./a.out`).
2. Retirer le commentaire devant l'appel à la fonction et utiliser gdb pour déterminer l'adresse début du tas.

Exercice 3. Compiler le programme `examiner_pile.c` afin de tracer son exécution sous gdb. Sous gdb, entrer la commande `b 10 if n == 0` pour placer un point d'arrêt conditionnel dans la procédure récursive. Lancer le programme via la commande `r`.

Examiner la pile d'appel via la commande `bt full`. Une vue plus hexadécimale de pile peut être obtenue tapant la commande `p {void *} $esp@20` afin de visualiser la pile sous forme d'un tableau de 20 pointeurs - (voir aussi `p /a {void *} $esp@20`).

Trouver la périodicité de l'empilement du paramètre de la fonction.

Deviner les indices des cases contenant des adresses de retour ; on pourra taper la commande `disassem` *adresse de retour* pour chaque adresse de retour et observer le code assembleur.

Exercice 4. Écrire un programme `tous_ul` qui retourne `EXIT_SUCCESS` si tous ses paramètres sont des entiers correspondant au type de base unsigned long et `EXIT_FAILURE` sinon (la commande unix `test` utilise ce formalisme).

Exercice 5. Rependre le TD précédent (`cp -r ~wacren/IP/4_pointeur ~/IP`) ; mettre en œuvre votre `strstr()`.