

## TP 4 (2025-11-21) — Débogage mémoire (noté)

Rendu papier à la fin de la séance

Fichiers/liens → [john.glikberg@uvsq.fr](mailto:john.glikberg@uvsq.fr)

Sujet du mail: "T0I/TP4 - <Nom complet>"

Décrivez votre réflexion — rédigez pour être lisible par un humain

---

### Débogage mémoire

#### Exercice 1 : un bug dans mon Laplacien

Dans le répertoire du cours ex1/ examinez le code fourni et tentez de le lancer. Il semble que cela soit du traitement d'image.

1. Que fait ce code ?
  - Expliquez main.c et les différentes images créées.
  - Expliquez le principe de la convolution et du laplacien dans le traitement d'image.
2. Écrivez un Makefile simple (un seul binaire) pour ce code.
3. Le code est-il fonctionnel ?
4. Quel est le bug et comment l'avez-vous trouvé ?
5. Corrigez le bug.

#### Exercice 2 : le Segfault

Ce code ex2/ semble incorrect. En fait, chacune des fonctions process\_data contient une erreur. Cependant elles ne plantent pas forcément de manière identique.

1. Expliquer en quoi chacune des fonctions process\_dataXX est fausse.
2. En commentant et décommentant les différentes fonctions (1,2,3), expliquer l'ampleur des dégâts et les erreurs rencontrées pour ces trois fonctions :
  - process\_data1 : plantage ? explication.
  - process\_data2 : plantage ? explication.
  - process\_data3 : plantage ? explication.
3. Quel outil vous permet de vous prémunir des erreurs de process\_data1 ?  
Inversement, quel autre outil utilisez-vous pour l'erreur de process\_data3 ?
4. Une corruption mémoire mène-t-elle toujours au plantage et pourquoi ?

### Exercice 3 : la fuite

Ce code ex3/ semble avoir un problème de fuite mémoire.

1. Expliquez ce qu'est une fuite mémoire et les conséquences possibles.
2. Trouvez la fuite dans le code avec un outil.
3. Corrigez la fuite.

### Débogage logique

#### Exercice 4 : la somme

1. Expliquez ce que fait ce programme.
2. Le programme fonctionne-t-il ?
3. Utilisez GDB pour afficher la valeur de sum à chaque coup de boucle.
4. Expliquez le bug et corrigez-le ; donnez la valeur finale de sum.

#### Exercice 5 : Condition d'arrêt

Le code de ex5/ contient une erreur qui mène à un crash.

1. Utilisez GDB pour trouver l'erreur.
2. Quelle est l'erreur ?
3. Corrigez l'erreur.

#### Exercice 6 : Quand ça dépasse

Le code de ex6/ contient une fonction vulnérable à un overflow.

1. Expliquez pourquoi ce code est vulnérable
2. Dessinez l'état de la pile lors de l'appel à `vulnerable_function`.
3. Comment pouvez-vous déclencher l'erreur ?
4. Que devez-vous taper dans le buffer pour changer la valeur de la variable `password_is_good` afin d'afficher « Vous avez cassé le MDP ! » ?

#### Exercice 7 : Ça n'avance plus ?

Le code ex7/ ne se termine jamais.

1. Expliquez ce que fait ce code :
  - Qu'est-ce qu'un thread ?
  - Qu'est-ce qu'un mutex ?
  - Que fait le code ?
2. Écrivez un makefile simple pour ce code.
  - Dépend-il d'une bibliothèque système ? Laquelle ?
3. Pourquoi le code ne se termine-t-il jamais ?
  - Comment afficher l'état du programme avec GDB ? Mettre des captures/copies dans le rapport de cet état. Que fait la commande `thread apply all bt` ?
4. Comment corriger ce code ?