

Debugging

Techniques de base

TP 1

Steven Costiou
RMoD, Inria Lille - Nord Europe
steven.costiou@inria.fr

Novembre 2019

Dans ces exercices, nous allons partir d'un rapport de *bug* fourni par un utilisateur pour enquêter sur un problème dans un programme. Pour cela nous appliquerons des techniques de *debugging*. La première étape consiste à reproduire et à observer le problème. Nous instrumenterons ensuite le programme pour obtenir plus d'informations.

1 Mise en place de l'environnement de travail

À partir de la page <https://kloum.io/costiou/teaching/debugging.html> :

- Télécharger les sources du programme *buggy*, qui sont `buggy.c` et `buggy.h`.
- Télécharger les sources du programme *anagrams*, qui sont `anagrams_IAGL.c` et `anagrams_IAGL.h`.

2 Tracer l'exécution du programme *buggy*

Rapport de *bug*. Je lance le programme *buggy* et rien ne s'affiche.

2.1 Reproduction et observation du problème

1. Compiler et exécuter le programme *buggy*. Que fait le programme ?
2. Observer le code de `buggy.c`. Quel est le comportement attendu ?
3. Le rapport de *bug* vous semble-t-il assez précis ? Si ça n'est pas le cas, préciser le rapport.

2.2 Insertion de traces

1. Tracer l'évolution du compteur dans *buggy.c*. Compiler et exécuter. Observer l'exécution du programme.

2. Tracer l'évolution du passage au carré dans *buggy.c*. Compiler et exécuter. Observer l'exécution du programme.
3. À partir des observations effectuées, préciser le rapport de *bug*.
Quelles hypothèses peut-on émettre concernant l'origine du problème ?
Essayer de confirmer les hypothèses et de corriger le programme.

3 Investigation avec gdb

Le *debugger* gdb permet de contrôler l'exécution du programme pas-à-pas à partir de points d'arrêt. Les points d'arrêts doivent être placés dans le programme au préalable, puis une fois que le programme est arrêté il est possible d'avancer l'exécution instruction par instruction. Cette opération est appelée *step*.

1. Configurer gdb pour *debugger* le programme.
2. Placer un point d'arrêt dans la boucle principale.
3. Une fois arrêté, utiliser la commande *next* pour exécuter la prochaine instruction dans le flot de contrôle. Cette technique est-elle suffisante pour mettre en lumière le problème reporté ?
4. Placer un point d'arrêt conditionnel pour arrêter le programme au moment où le problème devient visible. Utiliser les commandes d'exécution pas-à-pas pour avancer dans l'exécution du programme. Reporter le comportement observé.

4 Bugs dans le program anagrams

"Une anagramme (le mot est féminin) [...] est une construction fondée sur une figure de style qui inverse ou permute les lettres d'un mot ou d'un groupe de mots pour en extraire un sens ou un mot nouveau. Il est sous-entendu que ce nouveau mot existe dans un dictionnaire."^a

Le programme `anagrams_IAGL.c` est une implémentation du premier exercice "Anagrammes" du livre *Programmation efficace*.^b Dans cet exercice, nous prenons comme entrée la phrase suivante :

le chien marche vers sa niche et trouve une limace de chine
nue pleine de malice qui lui fait du charme

De cette phrase, nous allons extraire chaque mot qui est une anagramme d'un ou plusieurs autres mots contenus dans cette phrase. Tous les mots étants anagrammes l'un de l'autre sont regroupés et imprimés sur la sortie standard. Ainsi, si l'exercice est implémenté correctement, le résultat de l'exécution du programme devrait être le suivant :

```
{chien niche chine}  
{marche charme}  
{une nue}  
{limace malice}
```

^a<https://fr.wikipedia.org/wiki/Anagramme>

^bDürr, Christoph, and Jill-Jënn Vie. 2016. *Programmation efficace: Les 128 algorithmes qu'il faut avoir compris et codés dans sa vie*. Ellipses

4.1 Travail à rendre

Malheureusement, le programme contient des **bugs**!

Pour chaque bug :

1. Écrire un rapport de bug,
2. enquêter sur le problème en utilisant des traces ou GDB,
3. proposer une correction avec une justification,
4. proposer un test¹ assurant que le problème est corrigé.

Pour chaque étape, expliquer votre démarche de manière concise et précise.

¹Test unitaire, fonctionnel, manuel...