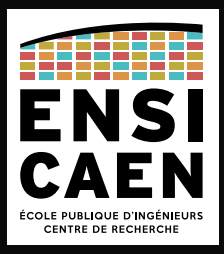

Outils de développement logiciel

TP n° 7 : débogage

Alain Lebret

2023-2024

Objectif

Comprendre et corriger différentes erreurs d'exécution à l'aide du débogueur ***GNU gdb***.
Pré-requis : chapitre 11
Durée estimée : 1 séance

Préliminaire

Comme dans certains TP précédents, nous vous proposons d'utiliser l'application ***asciinema***. Cette application va vous permettre d'enregistrer les commandes que vous entrez dans le terminal et les résultats de ces commandes. Voici par exemple la commande à exécuter dans l'exercice n° 2 pour enregistrer le terminal dans le fichier "tp07-prenom-nom-ex1.cast" que vous remettrez en fin de séance.

```
$ asciinema rec tp07-prenom-nom-ex1.cast -i=1
```

Dans le cas où vous auriez stoppé l'enregistrement avant terme, relancez la commande précédente en ajoutant l'option "--append" de manière à continuer l'enregistrement du fichier.

Pour stopper l'enregistrement d'***asciinema***, cela se fait à l'aide de la commande "`exit`" ou de la séquence Ctrl + d.

1 Introduction

Dans le TP sur la chaîne de compilation, nous avons rencontré un certain nombre d'options de ***GNU gcc***. Nous allons ici mettre en oeuvre l'option **-g** chargée d'ajouter à l'exécutable des informations symboliques qui permettent le débogage, et que nous utiliserons dorénavant lors de la mise au point de tout programme.

Afin de produire l'exécutable "`prog`" utilisable par ***gdb***, on compilera le fichier "prog.c" de la manière suivante :

```
$ gcc -g -o prog prog.c
```

Dans ce TP, on se passera de l'option d'optimisation `-Og` compatible avec `-g`, et on laissera de côté les options `-Wall`, `-Wextra`, `-ansi`, et `-pedantic` de manière à ne pas trop faciliter le débogage.

Nous avons vu qu'il était possible sans quitter ***Vim*** de compiler un fichier, d'exécuter un programme, voire de lancer n'importe quelle commande Unix. De la même manière, il est possible d'exécuter ***gdb*** et de l'associer au fichier en cours. Cela peut être réalisé en entrant les commandes suivantes depuis la fenêtre d'édition de ***Vim*** :

```
:packadd termdebug
:Termdebug
```

Vim découpe alors la fenêtre en trois panneaux dont, le panneau de contrôle du débogueur, celui de la sortie d'exécution du programme et celui d'édition. Le fenêtrage plaçant les trois panneaux verticalement, placez-vous dans le panneau d'édition (Ctrl + w w), puis afin de placer celui-ci à droite comme sur la figure ci-après, utilisez la combinaison : Ctrl + w ⇧ + l ("L" majuscule).

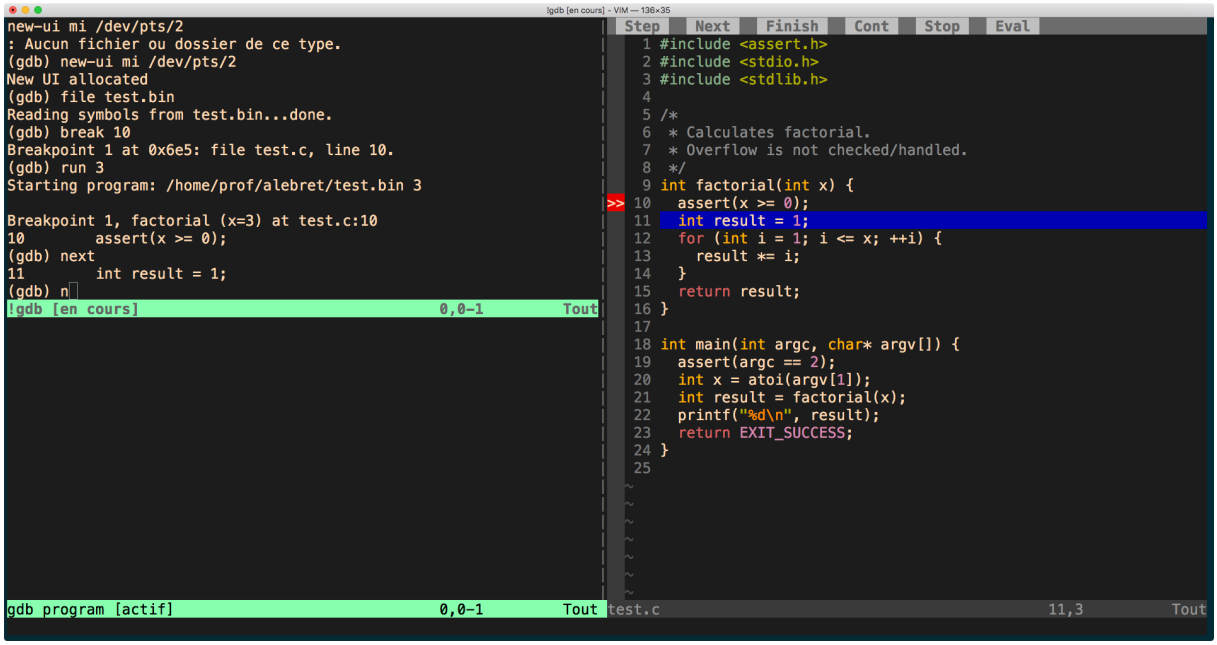

Figure 1: ***Vim*** en mode débogage avec ***GNU gdb***.

Pour chacun des exercices suivants, votre compte-rendu de séance devra comporter les traces de ***gdb*** ainsi qu'une analyse vous ayant permis de corriger la ou les anomalies.

2 Quelques anomalies à corriger

Exercice n° 1 : polynôme

Soit l'expression suivante où y et x sont des nombres réels :

$$y = \frac{x-1}{x} + \frac{1}{2}\frac{(x-1)^2}{x^2} + \frac{1}{3}\frac{(x-1)^3}{x^3} + \frac{1}{4}\frac{(x-1)^4}{x^4} + \frac{1}{5}\frac{(x-1)^5}{x^5}$$

On peut simplifier le calcul en posant :

$$u = \frac{(x-1)}{x}$$

Par exemple, pour x = 2 alors u vaut 0,5 et y = 0,688.

Le programme "polynomial.c" qui vous est fourni est supposé effectuer la résolution.

1. Déplacez-vous dans votre dossier "odl" et créez-y le sous-dossier "tp07", puis placez-vous dans ce dernier. "tp07" sera votre dossier de travail pour cette séance et vous y effectuerez toutes vos actions.
2. Copiez dans le dossier de travail le fichier "polynomial.c" que vous trouverez dans "ressources-odl-fisa/tp07/", puis ouvrez-le avec ***Vim***.
3. Sans quitter l'éditeur, compilez le fichier (attention, il ne faut pas oublier de réaliser l'édition de liens avec la bibliothèque mathématique : option `-lm`), puis exécutez le programme `polynomial` qui vient d'être généré. Apparemment, la valeur affichée ne correspond pas à celle que l'on attendait.
4. Activez ***gdb*** depuis ***Vim*** et associez-le au fichier. Placez un point d'arrêt au début de la fonction `main()`, puis suivez pas à pas l'évolution de vos variables. Corrigez le problème une fois détecté, puis vérifiez le bon fonctionnement.

Vous aurez sans doute à utiliser les commandes suivantes : `file`, `break`, `run`, `print`, `next`).

Exercice n° 2 : somme des *n* premiers entiers

1. Copiez dans le dossier de travail le fichier "sumto.c" que vous trouverez dans "ressources-odl-fisa/tp07/", compilez-le, puis exécutez le programme `sumto` que vous aurez généré. Apparemment, la valeur affichée de la somme des entiers ne correspond pas à celle que l'on attendait.

2. Associez ***gdb*** à l'exécutable et placez des points d'arrêt au début des fonctions `main()` et `sum_to()`.
3. Lancez le programme et vérifiez que la variable x est initialisée à "10". Affectez-lui la valeur 5, puis suivez pas à pas l'évolution des différentes variables dans `sum_to()`. Corrigez le problème et vérifiez le bon fonctionnement.

Vous aurez sans doute à utiliser les commandes suivantes : `file`, `run`, `print`, `next`, `continue` et `set var`.

Exercice n° 3 : aire et volume d'une sphère

1. Copiez dans le dossier de travail le fichier "sphere.c" que vous trouverez dans "ressources-odl-fisa/tp07/".
2. Compilez, puis exécutez le programme `sphere` que vous aurez généré. Apparemment, les valeurs affichées ne correspondent pas à celles que l'on attendait.
3. Associez ***gdb*** à l'exécutable, puis placez des points d'arrêt au début de la fonction `get_radius()` ainsi qu'à la ligne 59 dans la fonction `main()`. Lancez le programme et affectez la valeur 5 à la variable `radius`, puis suivez pas à pas l'évolution des variables dans `main()`. Corrigez le problème et vérifiez le bon fonctionnement.

Vous aurez sans doute à utiliser les commandes suivantes : `break`, `run`, `print`, `next`, `continue` et `set variable`.

Exercice n° 4 : une date changeante

1. Copiez dans le dossier de travail le fichier "year.c" que vous trouverez dans "ressources-odl-fisa/tp07/".
2. Compilez, puis exécutez le programme `year` que vous aurez généré. Apparemment, la valeur affichée pour la variable `year` n'est pas la même au début qu'à la fin.
3. Déverminer ce programme (indice : qu'en est-il de l'**adresse** de vos variables ?).

Exercice n° 5 : un angle pas vraiment moyen

Soit les angles $\alpha_1, \cdots, \alpha_n$, alors l'angle moyen $\bar{\alpha}$ est donné par :

$$\bar{\alpha} = atan2(\frac{1}{n}\sum_{j=1}^{n} sin(\alpha_j), \frac{1}{n}\sum_{j=1}^{n} cos(\alpha_j))$$

Exemples :

- L'angle moyen de l'ensemble { 350°, 10° } est 0.
- L'angle moyen de l'ensemble { 90°, 180°, 270°, 360° } est -90°.
- L'angle moyen de l'ensemble { 10°, 20°, 30° } est 20°.
- L'angle moyen de l'ensemble { 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80°, 90°, 100° } est 55°.

1. Copiez dans le dossier de travail le fichier "mean_angle.c" que vous trouverez dans "ressources-odl-fisa/tp07/".
2. Compilez, puis exécutez le programme `mean_angle` que vous aurez généré. Apparemment, le calcul de l'angle moyen ne s'est pas bien passé.
3. Déverminer ce programme avec ***gdb***.

Exercice n° 6 : un tri peu fiable

1. Copiez dans le dossier de travail le fichier "sort.c" que vous trouverez dans "ressources-odl-fisa/tp07/".
2. Compilez, puis exécutez le programme `sort` que vous aurez généré. Apparemment, le tri ne se déroule pas comme il faut.
3. Déverminer ce programme avec ***gdb***.

3 Livrable

Vérifiez que les fichiers d'animation ("tp07-prenom-nom-ex*.cast") sont valides en initiant leur lecture à l'aide de la commande "`asciinema play ...`" (pas la peine de les lire en entier !)

Déplacez alors l'ensemble des fichiers "*.cast" vers le sous-dossier "anims" que vous aurez préalablement créé, puis compressez ce dossier à l'aide de la commande `zip`. Déposez l'archive compressée sur Moodle.

4 Résumé

Dans ce TP vous avez mis en oeuvre le débogueur ***GNU gdb*** de manière à corriger des erreurs d'exécution.