

LE DÉBOGAGE : EXEMPLES AVEC GDB

Dedebant Louis Saï Ismaël

27/02/2017

Présentation réalisée dans le cadre du module HLIN408

INTRODUCTION

1. Le débogage

- Pourquoi se servir d'un débogueur?
- Le débogueur GDB

2. Utilisations de GDB

- Utilisation basique
- Utilisation avancée

3. Conclusion

QU'EST-CE QU'UN DÉBOGUEUR?

Analyser les bogues d'un programme.

Détecter les erreurs non détectées à la compilation

S'utilise en complément du compilateur

GDB est l'acronyme de GNU DeBugger.

Fonctionne par ligne de commandes

Compatible avec la plupart des système Unix

Utilisé sur plusieurs langages de programmation

Écrit en 1988 par Richard Stallman

Compilation du fichier avant utilisation de GDB

- `gcc -g -Wall exemple.c -o exemple`

Démarrage de gdb dans une fenêtre de commande

- `GDB <NomExecutable>`

```
lsma@lsma-HP-ENVY-15-Notebook-PC:~/Bureau$ gdb toto
GNU gdb (Ubuntu 7.11.1-0ubuntu1-16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
toto: Aucun fichier ou dossier de ce type.
(gdb) █
```

- Erreur de type "Core dumped" : gdb toto dumpfile

- Erreur de type "Core dumped" : `gdb toto dumpfile`
- Deboguer un executable déjà lancé : `gdb toto 1234`

- Erreur de type "Core dumped" : `gdb toto dumpfile`
- Deboguer un executable déjà lancé : `gdb toto 1234`
- Syntaxe type d'une commande : `(gdb) commandes`

- Erreur de type "Core dumped" : `gdb toto dumpfile`
- Deboguer un executable déjà lancé : `gdb toto 1234`
- Syntaxe type d'une commande : `(gdb) commandes`
- Obtenir de l'aide sur une commande : `(gdb) help[NomDeCommande]`

- Lancer le programme : (gdb) run (NomExecutable)

- Lancer le programme : `(gdb) run (NomExecutable)`
- Lancer le programme avec des points d'arrêt : `(gdb) start (NomExecutable)`

Il existe de multiples interfaces graphiques pour gdb

- Nemiver** s'intègre bien à Gnome

- Kdbg** s'intègre bien à KDE

- ddd** très complète

- cgdb** interface de type ncurses

- gdb-mode** pour emacs

Pour les prochains exemples, nous n'utiliserons pas d'interface graphique.

UTILISATIONS DE GDB

CONCLUSION

- + Logiciel très efficace
- + Corrige tout type d'erreur
- + Gratuit et libre d'accès
- Peut rapidement devenir fastidieux

MERCI DE VOTRE ATTENTION