

GDB

Introduction

- GDB : The GNU Project Debugger
- Debugger disponible en ligne de commandes ainsi qu'en interface graphique
- Documentation : <https://www.gnu.org/software/gdb/>

Support

- GDB supporte les langages suivants :
 - Natifs : C/C++
 - Partiel : D/Modula-2/Pascal
 - Objective-C avec un ordinateur Apple

Fonctionnalités

- Permet de lancer votre programme dans un environnement de débog avec des paramètres de départ
- Arrêter le programme à des emplacements spécifiques
- Examiner l'état du programme durant un arrêt ou au moment d'un crash

Liste des commandes de bases

- gdb 'NOM_DU_PROGRAMME'
 - Lance le debugger en lui spécifiant quel programme examiner
- break 'OPTION'
 - Crée un point d'arrêt à l'emplacement 'OPTION'
- run
 - Lance la session de debug
- n, s, c,
 - Permet d'avancer dans le programme
- bt, p
 - Permet d'interroger l'état du programme

Lancer gdb

- Pour lancer gdb il suffit de l'appeler via l'invite de commande
 - `gdb`
- gdb prend en argument le nom du programme à debugger
 - `gdb text.exe`
- Pour pouvoir debugger correctement, le programme doit être compilé en mode debug afin d'obtenir ses symboles
 - `g++ test.cpp -o test_debug.exe -g`

Lancer et arrêter le programme

- Une fois gdb lancer, la command 'run' permettra de commencer le programme
- Un point d'arrêt est automatiquement créé sur la fonction main au lancement
- Pour créer des points d'arrêts il suffit d'utiliser la command 'break'
- Pour lister tous les points d'arrêt courants :
 - info break
- Pour supprimer un point d'arrêt :
 - delete 'numero_du_breakpoint'

La commande break

- La commande break peut prendre plusieurs arguments :
 - break 'nom_de_la_fonction'
 - break 'numero_de_ligne'
 - break 'fichier:numero_de_ligne'

Commandes de navigation

- Ces commandes s'utilisent lorsque l'arrêt d'un programme est causée par un point d'arrêt
- continue (c) :
 - Continue l'exécution du programme jusqu'au prochain point d'arrêt
- next (n):
 - Avance l'exécution à la prochaine ligne du fichier
- step (s):
 - Avance l'exécution à la prochaine ligne du code source

Commandes d'affichage

- Il existe deux commandes pour afficher l'état du programme en cours
- bt (ou backtrace)
 - Affiche le stack du programme à l'état courant
- p (ou print)
 - Affiche la valeur d'une variable définie dans le scope

Exemple