

GDB - les principales commandes

Compilation des modules

Utiliser l'option de compilation `-g` pour chaque module d'un programme. Par exemple

```
clang -g prog.c -o prog
```

Syntaxe des commandes

Chaque commande GDB apparaît en gras, les caractères en style normal peuvent être ignorés.

Les arguments de commande apparaissent en italique entre `< >` et les arguments optionnels sont indiqués entre crochets `[]`. Par exemple

break [*<fichier_source>* :] *<num_ligne>*

signifie que la commande peut être utilisée sous l'une des formes suivantes :

`b 14` ou `b prog.c:14` ou `break 14` ou `break prog.c:14`

Les principales commandes

- Démarrage de gdb dans une fenêtre de commande :

```
gdb <nom_executable>
```

Le prompt (gdb) apparaît, les commandes GDB peuvent être exécutées. GDB se place au début de la procédure principale.

- Afficher les lignes du fichier source courant proches de la position courante

```
list
```

- Afficher les lignes d'un fichier source entre les lignes numérotées *<debut>* et *<fin>*

(si *<fin>* non indiqué, alors jusqu'à la fin du fichier)

```
list [<fichier_source> : ] <debut> , [<fin> ]
```

- Placer un point d'arrêt permanent dans un fichier source au début de l'instruction correspondant à la ligne numérotée *<num_ligne>*

(si `<fichier_source>` non indiqué, alors fichier source courant)

break [`<fichier_source>:`]`<num_ligne>`

- Placer un point d'arrêt temporaire dans un fichier source au début de l'instruction correspondant à la ligne numérotée `<num_ligne>`

tbreak [`<fichier_source>:`]`<num_ligne>`

- Afficher les points d'arrêt définis

info **break**

- Supprimer le point d'arrêt numéroté `<num>`

(si `<num>` non spécifié, alors tous les points d'arrêt sont supprimés)

delete [`<num>`]

- Démarrer l'exécution du programme (une liste d'arguments peut être spécifiée), et le programme s'arrête au premier point d'arrêt rencontré.

run [`<liste_arguments>`]

- Démarrer l'exécution du programme (une liste d'arguments peut être spécifiée), et le programme s'arrête au début de la ligne contenant la première instruction.

start [`<liste_arguments>`]

- Exécuter la ligne de programme

next

- Exécuter la ligne de programme en entrant dans les éventuelles procédures ou fonctions de l'instruction

step

- Continuer l'exécution du programme jusqu'au prochain point d'arrêt

continue

- Afficher la position courante

info source

- Afficher la valeur d'une variable nommée `<nom_var>`

print `<nom_var>`

- Afficher la valeur d'un élément d'indice `<ind>` d'un tableau nommé `<nom_tab>`

print `<nom_tab>(<ind>)`

ATTENTION : pas de vérification d'indice

- Afficher les valeurs des éléments d'un tableau nommé `<nom_tab>` entre les indices `<ind1>` et `<ind2>`

print `<nom_tab>(<ind1>..<ind2>)`

ATTENTION : pas de vérification d'indices

- Afficher la valeur d'une expression `<exp>`

print `<exp>`

- Changer la valeur d'une variable nommée `<nom_var>` avec la valeur de l'expression `<exp>`

set var *<nom_var>* := *<exp>*

ATTENTION : pas de vérification d'indice dans le cas de tableau

- Terminer l'exécution du programme (lorsque le programme est stoppé sur un point d'arrêt)

kill

- Afficher l'aide

help

- Quitter GDB

quit