

# GDB Cheat Sheet

## Commandes générales

```
gdb ./programme
gdb --args ./programme[ arg1[ arg2[ ...]]]
```

Démarre GDB pour déboguer le programme passé en paramètre. Les arguments utilisés par le programme à déboguer peuvent être passés au démarrage.

```
run
run[ arg1[ arg2[ ...]]]
```

Démarre (ou redémarre) le programme à déboguer. Si aucun argument n'est donné, les précédents sont utilisés. Si des arguments sont donnés, ils remplacent les précédents.

```
make
```

Effectue un appel à make.

## Gestion des breakpoints/watchpoints

```
break [fichier:]ligne [if condition]
break [fichier:]fonction [if condition]
```

Interrompt l'exécution du programme à déboguer. La position peut être indiquée par un numéro de ligne dans un fichier ou au début d'une fonction. L'interruption peut être effectuée si et seulement si une condition est vraie.

```
rwatch variable
watch variable
awatch variable
```

Interrompt l'exécution du programme à déboguer si la *variable* est lue, modifiée ou les deux.

```
info breakpoints
```

Donne la liste des breakpoints et des watchpoints avec leur numéro (*#breakpoint*).

```
ignore #breakpoint N
```

Ignore un breakpoint ou un watchpoint identifié par son numéro pendant *N* passages dessus.

```
delete #breakpoint
```

Supprime un breakpoint ou watchpoint identifié par son numéro.

## Commandes d'affichage

```
ptype type
ptype variable
ptype fonction
```

Affiche la définition d'un *type*, le type d'une *variable* ou le prototype d'une *fonction*.

```
print /format expression
```

Affiche la valeur de l'*expression* suivant le *format*. Le format est défini par une lettre:

```
c -> caractère (exemple : 65 -> 'a')
d -> entier signé
u -> entier non-signé
f -> flottant
t -> binaire (exemple : 65 -> 01000001)
x -> hexadécimal (exemple : 65 -> 41)
```

## Contrôle du flux d'exécution

```
continue
```

Reprend l'exécution du programme précédemment interrompu.

```
next[ N]
nexti[ N]
```

Exécute *N* lignes du programme ou *N* instructions du programme. Par défaut, *N* vaut 1. Un appel de fonction compte comme une ligne/instruction (on entre pas dans la fonction).

```
step[ N]
stepi[ N]
```

Identique à next/nexti sauf qu'on entre dans les fonctions.

```
finish
```

Termine l'exécution de la fonction en cours.

```
backtrace
```

Affiche la pile d'exécution (et les *#frame*).

```
frame #frame
```

Affiche une frame suivant son numéro.

```
call fonction(args)
```

Appel une fonction du programme.

```
return expression
```

Retourne de la fonction courante avec la valeur de l'*expression*