GDB GNU debugger

CONCEPTS

Compile with -g (and -O0 for sane stepping), otherwise GDB shows no source lines or local variables.

Break break, watch
↓ `run`
Stopped frame, locals
↓ `step / continue`
Inspect print, x, info
↓ `repeat`
Resume next, finish, continue

Press Enter to repeat the last command, and use short forms: b, r, c, n, s, bt, p, i.

START AND ATTACH

`gdb <prog>`
Load a program

`gdb --args <prog> <args>`
Load a program with arguments

`gdb -p <pid>`
Attach to a running process

`gdb <prog> <core>`
Inspect a core dump

`run [args]`
Start the program

`start`
Run and stop at the start of main

`set args <args>`
Set arguments for later runs

`attach <pid>`
Attach from inside GDB

`detach`
Release the process and keep it running

`quit`
Leave GDB

BREAKPOINTS

`break <func>`
Break at a function

`break <file>:<line>`
Break at a line of a file

`break <line>`
Break at a line of the current file

`break <loc> if <cond>`
Break only when the condition holds

BREAKPOINTS (cont.)

`tbreak <loc>`
One-shot breakpoint, deleted after first stop

`info breakpoints`
List breakpoints and watchpoints

`delete [<n>]`
Delete one breakpoint, or all

`disable <n> / enable <n>`
Turn a breakpoint off / on

`condition <n> <expr>`
Set or change a breakpoint condition

`ignore <n> <count>`
Skip the next hits of a breakpoint

WATCHPOINTS

`watch <var>`
Stop when a value changes

`watch -l <expr>`
Watch the memory address, not the scope

`rwatch <var>`
Stop when a value is read

`awatch <var>`
Stop on read or write

RUN CONTROL

`continue [<n>]`
Resume; optionally skip n-1 further hits here

`next [<n>]`
Step over calls to the next line

`step [<n>]`
Step into functions with debug info

`finish`
Run until the current function returns

`until`
Finish the loop without stepping through it

`until <loc>`
Run to a location in this frame

`Ctrl+C`
Interrupt the running program

STACK

`backtrace`
Show the call stack

STACK (cont.)

`backtrace full`
Include local variables in each frame

`thread apply all bt`
Backtrace of every thread

`frame <n>`
Select a frame

`up [<n>] / down [<n>]`
Move toward the caller / callee

`info locals`
Local variables of the frame

`info args`
Arguments of the frame

INSPECT DATA

`print <expr>`
Evaluate and show an expression

`print/x <expr>`
Print in hex (also d, o, t, c, a)

`print <arr>@<n>`
Show n elements from the first one

`set var <var> = <val>`
Change a variable

`display <expr>`
Show an expression at every stop

`x/<n><f><u> <addr>`
Examine memory

`ptype <expr-or-type>`
Show the full type definition

`set print pretty on`
Print structs one field per line

SOURCE AND SYMBOLS

`list`
Show source around the current line

`info threads`
List threads

TEXT UI

`layout src`
Source window above the command line

`Ctrl+X A`
Toggle the text UI on and off