GDB // quick reference — GNU Debugger

(gdb) ▮

EXECUTION CONTROL

`r / run [args]` Start or restart the program

`starti` Start, break at very first instruction

`c / continue` Resume execution

`n / next` Step over — source line

`ni / nexti` Step over — one instruction

`s / step` Step into — source line

`si / stepi` Step into — one instruction

`finish` Run until current function returns

`jump <loc>` Set PC to location and resume

`q / quit` Exit GDB

INPUT & ARGUMENTS

`set args args` Set argv before running

`r < <file>` Redirect stdin from file

`r <<< "string"` Feed string literal as stdin

REMOTE DEBUGGING

`$ gdbserver :PORT ./bin` Start stub (run this in a shell)

`target remote HOST:PORT` Connect GDB to stub

`target extended-remote HOST:PORT` Connect; allows run/restart

BREAKPOINTS & WATCHPOINTS

`b <loc>` Set breakpoint (addr / func / file:line)

`b <loc> if <expr>` Set conditional breakpoint

`watch <expr>` Break when memory location changes

`info breakpoints` List all breakpoints with their numbers

`condition <n> <expr>` Add/change condition on breakpoint n

`delete <n>` Delete breakpoint n (omit n for all)

`disable / enable <n>` Disable / enable breakpoint n

STACK

`bt / backtrace` Show full call stack

`up` Move one frame up (toward caller)

`down` Move one frame down (toward callee)

INSPECTION

`p <expr>` Print expression (decimal)

`p/x <expr>` Print expression (hex)

`x/s <addr>` Examine memory as C string

`display <expr>` Auto-print expression on every step

`undisplay <n>` Remove auto-display entry n

`set $reg = val` Overwrite register / convenience var

`info locals` Local variables in current frame

PROGRAM INFO

`info file` Entry point and section addresses

`info proc mappings` Memory map with permissions (needs running proc)

PRINT & EXAMINE FORMATS

`p/fmt <expr>` Print expression with format

`x/Nfmtsz <addr>` Examine N units of memory

FORMAT		SIZE (x)	
x	hex	**b**	byte (1)
d	signed decimal	**h**	half (2)
u	unsigned decimal	**w**	word (4)
o	octal	**g**	giant (8)
t	binary		
c	char		
s	string		
i	instructions		
f	float		
a	address		

TUI — TEXT USER INTERFACE

`layout src` Show source code view

`layout asm` Show assembly view

`layout regs` Show registers + assembly

`layout split` Show source and assembly together

`Ctrl-x a` Toggle TUI on / off

`Ctrl-x o` Cycle focus between TUI windows

`Ctrl-L` Refresh / redraw TUI screen