

Command line

<code>gdb [executable]</code>	begin debugging of [executable]
<code>gdb -p [PID]</code>	attach to and debug [PID]
<code>gdb -ex [commands...]</code>	execute [commands] on startup
<code>gdb --args [executable] [arguments...]</code>	begin debugging of [executable] with [arguments]

Breakpoints

<code>break, b [linenum]</code>	set a breakpoint at [linenum]
<code>break if [condition]</code>	set breakpoint here if [condition] is met
<code>break [code location] if [condition]</code>	set breakpoint at [code location] if [condition] is met
<code>info breakpoints</code>	list all breakpoints and their numbers and location
<code>clear</code>	delete all breakpoints
<code>delete [breakpoint num]</code>	delete [breakpoint num]
<code>enable [breakpoint num]</code>	enable [breakpoint num]
<code>disable [breakpoint num]</code>	disable [breakpoint num]

[code location] can be a number, for a line in the main file being debugged, or *filename:linenum*, *function*, or *filename:function*.

Executing

<code>run</code>	run the loaded program
<code>run [args...]</code>	run the loaded program with [args]
<code>next</code>	next line of source code
<code>step</code>	same as next but will dive into function calls
<code>finish</code>	continue until return instruction
<code>continue</code>	continue until next breakpoint

Examining Code

<code>backtrace</code>	show a backtrace
<code>where</code>	show the current call stack
<code>print [variable]</code>	print a representation of variable
<code>display [variable]</code>	same as print but will execute after each step instruction
<code>print/[format] [variable]</code>	print [variable] as [format]
<code>display/[format] [variable]</code>	display [variable] as [format]

Print and display can also show almost any C style expression.

Format

<code>a</code>	pointer
<code>c</code>	character
<code>d</code>	signed int
<code>u</code>	unsigned int
<code>f</code>	floating point number
<code>o</code>	octal
<code>x</code>	hexadecimal



By deoxys314

Not published yet.

Last updated 19th January, 2023.

Page 1 of 1.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>