## Command line

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

## Breakpoints

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

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

## Examining Code

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

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

## Format

| | |
|---|---|
| a | pointer |
| c | character |
| d | signed int |
| u | unsigned int |
| f | floating point number |
| o | octal |
| x | hexadecimal |