

# GDB Reference Sheet

## Preliminary

- compile and link your code with the -g flag to generate an executable with debugging symbols. Don't optimize your code when you're debugging it - optimized code doesn't correspond easily to the source.
- run gdb with the command "gdb <executable> [core dump]"

## Running Your Program

- file <executable> - loads the program you want to debug, if you forgot to specify it as a parameter to gdb
- run [param1 param2 ...] - runs the loaded executable with the given parameters
- start [param1 param2 ...] - same as run, but breaks at the entry to the main() function
- **target remote** <address> - for remote debugging

## Breakpoints

- **break** <position> - sets a new breakpoint. position can be a line number in the current file, a function name, or <file>:<line number>
- **delete** <number> - deletes the breakpoint with the given number. <number> is not the line number.
- **condition** <breakpoint number> <condition> - adds a condition to the breakpoint
- **watch** <variable> - sets a hardware-accelerated breakpoint that stops execution when the value of the variable changes
- **info breakpoints** - shows info about current breakpoints, including their numbers (used in delete)
- **enable** <number> - enable the breakpoint (useful for disabled breakpoints)
- **disable** <number> - disable a breakpoint

## Stepping through your code

- **step** - runs the code until the line number changes. If the statement is a function call, it will go into the call.
- **next** - runs the current statement. If it is a function call, the whole function call is ran.
- **stepi** - runs a single instruction
- **nexti** - runs only a single instruction or, for call instructions, the function call.
- **finish** - runs the current function until it's completion
- **continue** - run the program until the next breakpoint (or signal)

## Displaying Data

- **print**[/<format>] <expression> - prints the value of the expression, which is a C expression.
- **display** <expression> - prints the value of expression every time gdb stops the program
- **undisplay** <number> - cancels display of the expression corresponding to the given number
- **info locals** - prints values of all local variables
- **info registers** - displays the current values of the registers
- **backtrace** - traverses the call stack printing names (and current line numbers) of active functions
- **frame** <number> - allows switching between stack frames (numbers correspond to those listed by backtrace)
- **list** [<position>] - print out a few lines of code at the given position (or, if no position is given, at the current spot in the execution)

## Modifying Execution

- **call** <function>([param1, param2, ...])> - calls a function in your code. (be careful with this!)
- **set** var <variable> = <value>
- **return** <value>

## Ending the program

- **kill** - kills the currently executing program.
- **quit** - exits gdb