

Debugging with GDB

Most of the commands below can be executed simply by typing their first letter (*e.g.*, `p` for `print`).
Functions' and variables' names can be auto-completed by hitting `Tab`.

Getting started...

<code>% gdb foo [core]</code>	debug <code>foo</code> (and optionally its core dump)
<code>run</code>	start <code>foo</code>

Breakpoints...

<code>break bar</code>	temporarily stops execution as soon as <code>bar</code> is called
<code>break baz.c:n</code>	temporarily stops execution just before line <code>n</code> in <code>baz.c</code>
<code>info breakpoints</code>	list breakpoints and their numbers
<code>delete i</code>	delete breakpoint <code>i</code>
<code>delete</code>	delete all breakpoints

Walking through code...

<code>next</code>	execute the next line of code
<code>next n</code>	execute the next <code>n</code> lines of code
<code>step</code>	step into the next line of code
<code>continue</code>	continue execution from wherever you are to next breakpoint, if any
<code>finish</code>	finish the current function's execution
<code>list n</code>	print out some lines of code surrounding line <code>n</code>
<code>backtrace</code>	display the frame stack (<i>i.e.</i> , the series of function calls that got you there)
<code>bt</code>	display the frame stack (<i>i.e.</i> , the series of function calls that got you there)
<code>frame n</code>	change the current frame to frame <code>n</code>

Printing variables...

<code>print v</code>	print the value in <code>v</code>
<code>display v</code>	continually display the value in <code>v</code>
<code>print a[i]</code>	print the <code>i</code> th element of array <code>a</code>
<code>print *ptr</code>	print the value stored at address <code>ptr</code> (<i>i.e.</i> , dereference <code>ptr</code>)
<code>print obj.b</code>	print <code>obj</code> 's <code>b</code> data member
<code>print ptr->b</code>	print data member <code>b</code> in object pointed to by <code>ptr</code>