

Changing program execution while debugging;↩ Changing program execution while debugging

[arrow.eps](#) ↩ Use **`gdb`** commands to simulate a solution to a bug before building.

Once you find out what's wrong with your program, you might want to test that the solution you've come up with will work before you change the source code and rebuild. For example, what if you set a variable to a different value? Will that solve the problem?

[_ChangingProgramExec.eps](#) ↩

Command	Description
TableHeadRule.eps ↩ <code>call <i>function</i></code> TableRule.eps ↩ <code>jump <i>linenum</i></code>	<p>Executes the <i>function</i>. You can also use this for Objective-C messages.</p> <p>Resume execution at line number <i>linenum</i>. Execution may stop immediately if there's a breakpoint there.</p> <p>The jump command doesn't change the current stack frame, or the stack pointer, or the contents of any memory location or any register other than the program counter. If <i>linenum</i> is in a different function from the one currently executing, the results may be wild if the two functions expect different patterns of arguments or of local variables. For this reason, the jump command requests confirmation if the specified line isn't in the function currently executing.</p>
TableRule.eps ↩ <code>jump <i>*address</i></code> TableRule.eps ↩ <code>set <i>var</i> = <i>exp</i></code> TableRule.eps ↩	<p>Resume execution at the instruction at address <i>address</i>.</p> <p>Perform an assignment .</p>