Watchpoints

/run/media/lijunjie/资料/Learning Video/writing-running-fixing-code/03_testing-and-debugging/03_debugging-tools/07_watchpoints_instructions.html

Another incredibly useful feature of *gdb* is a *watchpoint* —the ability to have gdb stop when the value of a particular expression changes. For example, we can write *watch i,* which will cause *gdb* to stop whenever the value of *i* changes. When *gdb* stops in response to a watchpoint, it will print the old value of the expression and the new value.

Watchpoints can be a particularly powerful tool when you have pointer-related problems, and values of variables are changing through aliases. However, sometimes, the alias we have when we setup the watchpoint may go out of scope before the change we are interested in happens. For example, we may want to *watch *p* , but *p* is a local variable, whose scope ends before the value changes. Whenever we face such a problem, we can *print p* , which will give us the pointer in a *gdb* variable (*e.g., $6*), and then we can use that *$* -variable (which never goes out of scope—it lives until we restart *gdb*) to set our watchpoint: watch **$6* .