

命令	解释	示例
file <文件名>	加载被调试的可执行程序文件。 因为一般都在被调试程序所在目录下执行 GDB，因而文本名不需要带路径。	(gdb) file gdb-sample
r	Run 的简写，运行被调试的程序。 如果此前没有下过断点，则执行完整程序；如果有断点，则程序暂停在第一个可用断点处。	(gdb) r
c	Continue 的简写，继续执行被调试程序，直至下一个断点或程序结束。	(gdb) c
b <行号> b <函数名称> b *<函数名称> b *<代码地址>  d [编号]	b: Breakpoint 的简写，设置断点。两可以使用“行号”“函数名称”“执行地址”等方式指定断点位置。  其中在函数名称前面加“*”符号表示将断点设置在“由编译器生成的 prolog 代码处”。如果不了解汇编，可以不予理会此用法。  d: Delete breakpoint 的简写，删除指定编号的某个断点，或删除所有断点。断点编号从 1 开始递增。	(gdb) b 8 (gdb) b main (gdb) b *main (gdb) b *0x804835c  (gdb) d
s, n	s: 执行一行源程序代码，如果此行代码中有函数调用，则进入该函数； n: 执行一行源程序代码，此行代码中的函数调用也一并执行。  s 相当于其它调试器中的“Step Into (单步跟踪进入)”； n 相当于其它调试器中的“Step Over (单步跟踪)”。  这两个命令必须在有源代码调试信息的情况下才可以使用（GCC 编译时使用“-g”参数）。	(gdb) s (gdb) n
si, ni	si 命令类似于 s 命令，ni 命令类似于 n 命令。所不同的是，这两个命令（si/ni）所针对的是汇编指令，而 s/n 针对的是源代码。	(gdb) si (gdb) ni
p <变量名称>	Print 的简写，显示指定变量（临时变量或全局变量）的值。	(gdb) p i (gdb) p nGlobalVar
display ...  undisplay <编号>	display，设置程序中断后欲显示的数据及其格式。 例如，如果希望每次程序中断后可以看到即将被执行的下一条汇编指令，可以使用命令 “display /i \$pc” 其中 \$pc 代表当前汇编指令，/i 表示以十六进行显示。当需要关心汇编代码时，此命令相当有用。	(gdb) display /i \$pc  (gdb) undisplay 1

	undisplay, 取消先前的 display 设置, 编号从 1 开始递增。	
i	Info 的简写, 用于显示各类信息, 详情请查阅 “help i”。	(gdb) i r
q	Quit 的简写, 退出 GDB 调试环境。	(gdb) q
help [命令名称]	GDB 帮助命令, 提供对 GDB 各种命令的解释说明。如果指定了 “命令名称” 参数, 则显示该命令的详细说明; 如果没有指定参数, 则分类显示所有 GDB 命令, 供用户进一步浏览和查询。	(gdb) help display