

命 令	缩写	用 法	作 用
help	h	h command	显示命令的帮助
run	r	r [args]	运行要调试的程序 args为要运行程序的参数
step	s	s [n]	步进,n为步进次数。如果调用了某个函数,会跳入函数内部。
next	n	n [n]	下一步,n为下一步的次数
continue	c	c	继续执行程序
list	l	l / l+ / l-	列出源码
break	b	b address	在地址address上设置断点
		b function	此命令用来在某个函数上设置断点。
		b linenum	在行号为linenum的行上设置断点。程序在运行到此行之前停止
		b +offset	在当前程序运行到的前几行或后几行设置断点。offset为行号
		b -offset	
		b filename:linenum	在文件名为filename的原文件的第linenum行设置断点
		b filename:function	在文件名为filename的原文件名为function的函数上设置断点。当你的多个文件中可能含有相同的函数名时必须给出文件名
watch	w	w exp	监视表达式的值
kill	k	k	结束当前调试的程序
print	p	p exp	打印表达式的值
		p/fmt exp	fmt为以下值 x 十六进制 d 十进制 u 无符号数 o 八进制 t 二进制 a 十六进制打印 c 字符格式 f 浮点数
output	o	o exp	同print,但是不输出下一行的语句
ptype		ptype struct	输出一个struct结构的定义
whatis		whatis var	命令可以显示某个变量的类型
x		x/(length)(format)(size) addr x/6(o/d/x/u/c/t)(b/h/w) 例如: x/4xw	按一定格式显示内存地址或变量的值
pwd		pwd	显示当前路径
delete	d	d num	删除编号为num的断点和监视
disable		disable n	编号为n的断点暂时无效
enable		enable n	与disable 相反
display		display expr	暂停,步进时自动显示表达式的值
finish			执行直到函数返回 执行直到当前stack返回
return			强制从当前函数返回
where			命令用来查看执行的代码在什么地方中止
backtrace	bt		显示函数调用得所有栈框架(stack frames)的 踪迹和当前函数的参数的值。
quit	q		退出调试程序
frame			
shell		shell ls	执行shell命令
make			不退出gdb而重新编译生成可执行文件
disassemble			显示反汇编代码
thread		thread thread_no	用来在线程之间的切换
set		set width 70	就是把标准屏幕设为70列
		set var=54	设置变量的值。
forward/search		search string	从当前行向后查找匹配某个字符串的程序行
reverse-search			forward/search相反,向前查找字符串。使用格式同上
up/down			上移/下移栈帧,使另一函数成为当前函数
info	i	i breakpoint	显示当前断点列表
		i reg[ister]	显示寄存器信息
		i threads	显示线程信息
		i func	显示所有的函数名
		i local	显示当前函数的所有局部变量的信息
		i prog	显示调试程序的执行状态
		i watch exp	为表达式(变量)exp设置一个观察点。一旦表达式值有变化时,马上停住程序。
		i proc	显示进程的概要信息
		info proc mappings	报告你进程所能访问的地址范围。
		info proc times	你进程和子进程的开始时间,用户时间(user CPU time),和系统CPU时间。
		info proc id	报告有关进程id的信息
		info proc status	报告你进程的一般状态信息。如果进程停止了。这个报告还包括停止的原因和收到的信号
		info proc all	显示上面proc命令这些命令返回的所有信息