	,		
命令	缩写	用 法	作 用
help	h	h command	显示命令的帮助
	r	r [args]	运行要调试的程序 args为要运行程序的参数
	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。
	n	n [n]	下一步, n为下一步的次数
	c	c	维续执行程序
	1	1 / 1+ / 1-	列出源码
	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	P-10 0/00
break		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 filename:linenum	在文件名为filename的原文件的第linenum行设置断点
			在文件名为filename的原文件的名为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
		b filename:function	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	р	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	0	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/(length)	(format)(size) addr	
X	x/6(o/d/x/u	/c/t) (b/h/w)	按一定格式显示内存地址或变量的值
	例如: x/4xv	1	
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
			NILEZEDARE INI ENTERIORE
return			强制从当前函数返回
return where			
where	bt		强制从当前函数返回
where backtrace	bt q		强制从当前函数返回命令用来查看执行的代码在什么地方中止
where backtrace			强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。
where backtrace quit		shell ls	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。
where backtrace quit frame		shell 1s	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序
where backtrace quit frame shell		shell 1s	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令
where backtrace quit frame shell make		shell 1s thread thread_no	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件
where backtrace quit frame shell make disassemble thread			强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码
where backtrace quit frame shell make disassemble		thread thread_no	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换
where backtrace quit frame shell make disassemble thread		thread thread_no set width 70	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列
where backtrace quit frame shell make disassemble thread		thread thread_no set width 70 set var=54	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上
where backtrace quit frame shell make disassemble thread set forward/search		thread thread_no set width 70 set var=54	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行
where backtrace quit frame shell make disassemble thread set forward/search reverse-search		thread thread_no set width 70 set var=54	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上
where backtrace quit frame shell make disassemble thread set forward/search reverse-search		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister]	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表
where backtrace quit frame shell make disassemble thread set forward/search reverse-search		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示务存器信息
where backtrace quit frame shell make disassemble thread set forward/search reverse-search		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示线程信息 显示所有的函数名
where backtrace quit frame shell make disassemble thread set forward/search reverse-search		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示线程信息 显示线程信息 显示线程信息
where backtrace quit frame shell make disassemble thread set forward/search reverse-search up/down		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local i prog	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示线程信息 显示线程信息 显示线程信息 显示线程信息
where backtrace quit frame shell make disassemble thread set forward/search reverse-search		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示统程信息 显示所有的函数名 显示当前函数的所有局部变量的信息 显示调试程序的执行状态 为表达式(变量)exp设置一个观察点。一但表达式值有变化时,马上停住程序。
where backtrace quit frame shell make disassemble thread set forward/search reverse-search up/down		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local i prog i watch exp i proc	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示统程信息 显示所有的函数名 显示当前函数的所有局部变量的信息 显示调试程序的执行状态 为表达式(变量)exp设置一个观察点。一但表达式值有变化时,马上停住程序。 显示进程的概要信息
where backtrace quit frame shell make disassemble thread set forward/search reverse-search up/down		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local i prog i watch exp i proc info proc mappings	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示统程信息 显示统程信息 显示统程信息 显示调试程序的执行状态 为表达式(变量)exp设置一个观察点。一但表达式值有变化时,马上停住程序。 显示进程的概要信息 报告你进程所能访问的地址范围。
where backtrace quit frame shell make disassemble thread set forward/search reverse-search up/down		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local i prog i watch exp i proc info proc mappings info proc times	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行she11命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示统程信息 显示统程信息 显示的函数的所有局部变量的信息 显示调试程序的执行状态 为表达式(变量)exp设置一个观察点。一但表达式值有变化时,马上停住程序。 显示进程的概要信息 报告你进程所能访问的地址范围。 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。
where backtrace quit frame shell make disassemble thread set forward/search reverse-search up/down		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local i prog i watch exp i proc info proc mappings info proc times info proc id	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行shell命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示等存器信息 显示线程信息 显示线程信息 显示线程信息 显示调试程序的执行状态 为表达式(变量)exp设置一个观察点。一但表达式值有变化时,马上停住程序。 显示进程的概要信息 报告你进程所能访问的地址范围。 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。 报告有关进程id的信息
where backtrace quit frame shell make disassemble thread set forward/search reverse-search up/down		thread thread_no set width 70 set var=54 search string i breakpoint i reg[ister] i threads i func i local i prog i watch exp i proc info proc mappings info proc times	强制从当前函数返回 命令用来查看执行的代码在什么地方中止 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。 退出调试程序 执行she11命令 不退出gdb而重新编译生成可执行文件 显示反汇编代码 用来在线程之间的切换 就是把标准屏幕设为70列 设置变量的值。 从当前行向后查找匹配某个字符串的程序行 forward/search相反,向前查找字符串。使用格式同上 上移/下移栈帧,使另一函数成为当前函数 显示当前断点列表 显示寄存器信息 显示统程信息 显示统程信息 显示的函数的所有局部变量的信息 显示调试程序的执行状态 为表达式(变量)exp设置一个观察点。一但表达式值有变化时,马上停住程序。 显示进程的概要信息 报告你进程所能访问的地址范围。 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。