陈皓专栏 【空谷幽兰，心如皓月】

芝兰生于深谷，不以无人而不芳；君子修道立德，不为困穷而改节。

我的BLOG

陈皓专栏（技　术）(RSS)
酷壳（编程和技术）(RSS)

个人资料

haoel

访问： 2334645次
积分： 17381分
排名： 第91名

原创： 120篇 转载： 6篇
译文： 15篇 评论： 4710条

用GDB调试程序（五）

分类： 编程工具 2003-07-09 08:30 14231人阅读 评论(1) 收藏 举报

查看运行时数据

在你调试程序时，当程序被停住时，你可以使用print命令（简写命令为p），或是同义命令inspect来查看当前程序的运行数据。print命令的格式是：

```
print <expr>
print /<f> <expr>
```

<expr>是表达式，是你所调试的程序的语言的表达式（GDB可以调试多种编程语言），<f>是输出的格式，比如，如果要把表达式按16进制的格式输出，那么就是/x。

一、表达式

print和许多GDB的命令一样，可以接受一个表达式，GDB会根据当前的程序运行的数据来计算这个表达式，既然是表达式，那么就可以是当前程序运行中的const常量、变量、函数等内容。可惜的是GDB不能使用你在程序中所定义的宏。

表达式的语法应该是当前所调试的语言的语法，由于C/C++是一种大众型的语言，所以，本文中的例子都是关于C/C++的。（而关于用GDB调试其它语言的章节，我将在后面介绍）

在表达式中，有几种GDB所支持的操作符，它们可以用在任何一种语言中。

```
@
```

是一个和数组有关的操作符，在后面会有更详细的说明。

```
::
```

指定一个在文件或是一个函数中的变量。

```
{<type>} <addr>
```

表示一个指向内存地址<addr>的类型为type的一个对象。

二、程序变量

在GDB中，你可以随时查看以下三种变量的值：

1、全局变量（所有文件可见的）
2、静态全局变量（当前文件可见的）

(62817)
再谈“我是怎么招聘程序员
(54256)
一些重要的算法
(51996)
C++ 对象的内存布局（上
(46037)
哥是玩程序的
(43844)
清华大学出版社“抄袭事件
(43597)
恐怖的C++语言
(39271)

评论排行

清华大学出版社“抄袭事件 (323)
再谈“我是怎么招聘程序员 (264)
“清华大学出版社抄袭事件 (233)
我是怎么招聘程序员的 (233)
C++ 虚函数表解析 (203)
6个变态的C语言写的Hel (201)
恐怖的C++语言 (198)
优秀程序员的十个习惯 (165)
Java构造时成员初始化的 (146)
惹恼程序员的十件事 (130)

推荐文章

最新评论

跟我一起写 Makefile（一）
wangjilebeishang: 我的神啊，可以，顶《《《《学习
五个方法成为更好的程序员
雨中风铃_: 受教！
C++ 虚函数表解析
Walkerinwind: 第一个例子结束实际运行"经"果如下或许应该是结果?
“抄袭事件”判决书
fengyishang: 很遗憾，时隔七年，我才看到七年前这一起关于著作权的纠纷；但很幸运，因为最后的结果正是我们想要的，虽然...
C++ 虚函数表解析
zhiweiarm: 受益匪浅，学习了
22条经典的编程引言
雨中风铃_: 太经典了！
C++ 对象的内存布局（下）
Huang850615: cout << " "; pFun(); //D::f();这个地方出错了，把它改成这样：cout...
五个方法成为更好的程序员
冰冻牙膏: 寻找不同观点，取长补短
跟我一起写 Makefile（一）
luomuxiaoxiao98: Mark！
其实Unix很简单
lovelock: @yhmhappy2006:呵呵，我真不知道你干了什么。。。。

3、局部变量（当前Scope可见的）

如果你的局部变量和全局变量发生冲突（也就是重名），一般情况下是局部变量会隐藏全局变量，也就是说，如果一个全局变量和一个函数中的局部变量同名时，如果当前停止点在函数中，用print显示出的变量的值会是函数中的局部变量的值。如果此时你想查看全局变量的值时，你可以使用“::”操作符：

```
file::variable
function::variable
```

可以通过这种形式指定你所想查看的变量，是哪个文件中的或是哪个函数中的。例如，查看文件f2.c中的全局变量x的值：

```
gdb) p 'f2.c'::x
```

当然，“::”操作符会和C++中的发生冲突，GDB能自动识别“::”是否C++的操作符，所以你不必担心在调试C++程序时会出现异常。

另外，需要注意的是，如果你的程序编译时开启了优化选项，那么在用GDB调试被优化过的程序时，可能会发生某些变量不能访问，或是取值错误码的情况。这个是很正常的，因为优化程序会删改你的程序，整理你程序的语句顺序，剔除一些无意义的变量等，所以在GDB调试这种程序时，运行时的指令和你所编写指令就有不一样，也就会出现你所想象不到的结果。对付这种情况时，需要在编译程序时关闭编译优化。一般来说，几乎所有的编译器都支持编译优化的开关，例如，GNU的C/C++编译器GCC，你可以使用“-gstabs”选项来解决这个问题。关于编译器的参数，还请查看编译器的使用说明文档。

三、数组

有时候，你需要查看一段连续的内存空间的值。比如数组的一段，或是动态分配的数据的大小。你可以使用GDB的“@”操作符，“@”的左边是第一个内存的地址的值，“@”的右边则你你想查看内存的长度。例如，你的程序中有这样的语句：

```
int *array = (int *) malloc (len * sizeof (int));
```

于是，在GDB调试过程中，你可以以如下命令显示出这个动态数组的取值：

```
p *array@len
```

@的左边是数组的首地址的值，也就是变量array所指向的内容，右边则是数据的长度，其保存在变量len中，其输出结果，大约是下面这个样子的：

```
(gdb) p *array@len
$1 = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40}
```

如果是静态数组的话，可以直接用print数组名，就可以显示数组中所有数据的内容了。

四、输出格式

一般来说，GDB会根据变量的类型输出变量的值。但你也可以自定义GDB的输出的格式。例如，你想输出一个整数的十六进制，或是二进制来查看这个整型变量的中的位的情况。要做到这样，你可以使用GDB的数据显示格式：

x　按十六进制格式显示变量。
d　按十进制格式显示变量。
u　按十六进制格式显示无符号整型。
o　按八进制格式显示变量。
t　按二进制格式显示变量。

a 按十六进制格式显示变量。
c 按字符格式显示变量。
f 按浮点数格式显示变量。

```
(gdb) p i
$21 = 101

(gdb) p/a i
$22 = 0x65

(gdb) p/c i
$23 = 101 'e'

(gdb) p/f i
$24 = 1.41531145e-43

(gdb) p/x i
$25 = 0x65

(gdb) p/t i
$26 = 1100101
```

五、查看内存

你可以使用examine命令（简写是x）来查看内存地址中的值。x命令的语法如下所示:

```
x/<n/f/u> <addr>
```

n、f、u是可选的参数。

n 是一个正整数，表示显示内存的长度，也就是说从当前地址向后显示几个地址的内容。
f 表示显示的格式，参见上面。如果地址所指的是字符串，那么格式可以是s，如果地十是指令地址，那么格式可以是i。
u 表示从当前地址往后请求的字节数，如果不指定的话，GDB默认是4个bytes。u参数可以用下面的字符来代替，b表示单字节，h表示双字节，w表示四字节，g表示八字节。当我们指定了字节长度后，GDB会从指内存定的内存地址开始，读写指定字节，并把其当作一个值取出来。

<addr>表示一个内存地址。

n/f/u三个参数可以一起使用。例如:

命令: x/3uh 0x54320 表示，从内存地址0x54320读取内容，h表示以双字节为一个单位，3表示三个单位，u表示按十六进制显示。

六、自动显示

你可以设置一些自动显示的变量，当程序停住时，或是在你单步跟踪时，这些变量会自动显示。相关的GDB命令是display。

```
display <expr>
display/<fmt> <expr>
display/<fmt> <addr>
```

expr是一个表达式，fmt表示显示的格式，addr表示内存地址，当你用display设定好了一个或多个表达式

后，只要你的程序被停下来，GDB会自动显示你所设置的这些表达式的值。

格式i和s同样被display支持，一个非常有用的命令是：

```
display/i $pc
```

$pc是GDB的环境变量，表示着指令的地址，/i则表示输出格式为机器指令码，也就是汇编。于是当程序停下后，就会出现源代码和机器指令码相对应的情形，这是一个很有意思的功能。

下面是一些和display相关的GDB命令：

```
undisplay <dnums...>
delete display <dnums...>
```

删除自动显示，dnums意为所设置好了的自动显式的编号。如果要同时删除几个，编号可以用空格分隔，如果要删除一个范围内的编号，可以用减号表示（如：2-5）

```
disable display <dnums...>
enable display <dnums...>
```

disable和enalbe不删除自动显示的设置，而只是让其失效和恢复。

```
info display
```

查看display设置的自动显示的信息。GDB会打出一张表格，向你报告当然调试中设置了多少个自动显示设置，其中包括，设置的编号，表达式，是否enable。

<-上一页 下一页->

分享到：

上一篇：用GDB调试程序（五）

下一篇：用GDB调试程序（七）

查看评论

1楼 程序猿__int64Ago 2012-02-22 10:38发表

x <addr> 真的很有用哦～真为这个事着急呢

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场