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。

用GDB调试程序（一）

分类： 编程工具 2003-07-02 23:13 88108人阅读 评论(30) 收藏 举报

程序调试工具 file 工具 图形 unix path

用GDB调试程序

GDB概述

GDB是GNU开源组织发布的一个强大的UNIX下的程序调试工具。或许，各位比较喜欢那种图形界面方式的，像VC、BCB等IDE的调试，但如果你是在UNIX平台下做软件，你会发现GDB这个调试工具有比VC、BCB的图形化调试器更强大的功能。所谓“寸有所长，尺有所短”就是这个道理。

一般来说，GDB主要帮忙你完成下面四个方面的功能：

1、启动你的程序，可以按照你的自定义的要求随心所欲的运行程序。
2、可让被调试的程序在你所指定的调置的断点处停住。（断点可以是条件表达式）
3、当程序被停住时，可以检查此时你的程序中所发生的事。
4、动态的改变你程序的执行环境。

从上面看来，GDB和一般的调试工具没有什么两样，基本上也是完成这些功能，不过在细节上，你会发现GDB这个调试工具的强大，大家可能比较习惯了图形化的调试工具，但有时候，命令行的调试工具却有着图形化工具所不能完成的功能。让我们一一看来。

一个调试示例

源程序：tst.c

```
 1 #include <stdio.h>
 2
 3 int func(int n)
 4 {
 5         int sum=0,i;
 6         for(i=0; i<n; i++)
 7         {
 8                 sum+=i;
 9         }
10         return sum;
11 }
12
13
14 main()
15 {
```

(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。。。。

```
    16          int i;
    17          long result = 0;
    18          for(i=1; i<=100; i++)
    19          {
    20                  result += i;
    21          }
    22
    23         printf("result[1-100] = %d /n", result );
    24         printf("result[1-250] = %d /n", func(250) );
    25 }
```

编译生成执行文件：（Linux下）

```
    hchen/test> cc -g tst.c -o tst
```

使用GDB调试：

```
hchen/test> gdb tst  <---------- 启动GDB
GNU gdb 5.1.1
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB.  Type "show warranty" for details.
This GDB was configured as "i386-suse-linux"...
(gdb) l     <-------------------- l命令相当于list，从第一行开始例出原码。
1       #include <stdio.h>
2
3       int func(int n)
4       {
5               int sum=0,i;
6               for(i=0; i<n; i++)
7               {
8                       sum+=i;
9               }
10              return sum;
(gdb)       <-------------------- 直接回车表示，重复上一次命令
11      }
12
13
14      main()
15      {
16              int i;
17              long result = 0;
18              for(i=1; i<=100; i++)
19              {
20                      result += i;
(gdb) break 16    <-------------------- 设置断点，在源程序第16行处。
Breakpoint 1 at 0x8048496: file tst.c, line 16.
(gdb) break func  <-------------------- 设置断点，在函数func()入口处。
Breakpoint 2 at 0x8048456: file tst.c, line 5.
(gdb) info break  <-------------------- 查看断点信息。
Num Type           Disp Enb Address    What
1   breakpoint     keep y   0x08048496 in main at tst.c:16
2   breakpoint     keep y   0x08048456 in func at tst.c:5
(gdb) r           <-------------------- 运行程序，run命令简写
```

```
Starting program: /home/hchen/test/tst

Breakpoint 1, main () at tst.c:17    <---------- 在断点处停住。
17              long result = 0;
(gdb) n          <---------------------- 单条语句执行，next命令简写。
18              for(i=1; i<=100; i++)
(gdb) n
20                      result += i;
(gdb) n
18              for(i=1; i<=100; i++)
(gdb) n
20                      result += i;
(gdb) c          <---------------------- 继续运行程序，continue命令简写。
Continuing.
result[1-100] = 5050       <----------程序输出。

Breakpoint 2, func (n=250) at tst.c:5
5               int sum=0,i;
(gdb) n
6               for(i=1; i<=n; i++)
(gdb) p i        <---------------------- 打印变量i的值，print命令简写。
$1 = 134513808
(gdb) n
8                       sum+=i;
(gdb) n
6               for(i=1; i<=n; i++)
(gdb) p sum
$2 = 1
(gdb) n
8                       sum+=i;
(gdb) p i
$3 = 2
(gdb) n
6               for(i=1; i<=n; i++)
(gdb) p sum
$4 = 3
(gdb) bt        <---------------------- 查看函数堆栈。
#0  func (n=250) at tst.c:5
#1  0x080484e4 in main () at tst.c:24
#2  0x400409ed in __libc_start_main () from /lib/libc.so.6
(gdb) finish    <---------------------- 退出函数。
Run till exit from #0  func (n=250) at tst.c:5
0x080484e4 in main () at tst.c:24
24              printf("result[1-250] = %d /n", func(250) );
Value returned is $6 = 31375
(gdb) c     <---------------------- 继续运行。
Continuing.
result[1-250] = 31375    <----------程序输出。

Program exited with code 027. <--------程序退出，调试结束。
(gdb) q     <---------------------- 退出gdb。
hchen/test>
```

好了，有了以上的感性认识，还是让我们来系统地认识一下gdb吧。

使用GDB

一般来说GDB主要调试的是C/C++的程序。要调试C/C++的程序，首先在编译时，我们必须要把调试信息加到可执行文件中。使用编译器（cc/gcc/g++）的 -g 参数可以做到这一点。如：

```
> cc -g hello.c -o hello
> g++ -g hello.cpp -o hello
```

如果没有-g，你将看不见程序的函数名、变量名，所代替的全是运行时的内存地址。当你用-g把调试信息加入之后，并成功编译目标代码以后，让我们来看看如何用gdb来调试他。

启动GDB的方法有以下几种：

1、gdb <program>
program也就是你的执行文件，一般在当然目录下。

2、gdb <program> core
用gdb同时调试一个运行程序和core文件，core是程序非法执行后core dump后产生的文件。

3、gdb <program> <PID>
如果你的程序是一个服务程序，那么你可以指定这个服务程序运行时的进程ID。gdb会自动attach上去，并调试他。program应该在PATH环境变量中搜索得到。

GDB启动时，可以加上一些GDB的启动开关，详细的开关可以用gdb -help查看。我在下面只例举一些比较常用的参数：

-symbols <file>
-s <file>
从指定文件中读取符号表。

-se file
从指定文件中读取符号表信息，并把他用在可执行文件中。

-core <file>
-c <file>
调试时core dump的core文件。

-directory <directory>
-d <directory>
加入一个源文件的搜索路径。默认搜索路径是环境变量中PATH所定义的路径。

下一页->

（版权所有，转载时请注明作者和出处）

分享到：

上一篇：《自适应软件开发》—从奴隶主到领袖< 1 序言 >
下一篇：用GDB调试程序（二）

查看评论

29楼 xj34324 2012-12-20 18:22发表

有点不明白，next命令执行后，是执行之前的指令还是之后的指令？
比如：
Breakpoint 1, main () at tst.c:17 <---------- 在断点处停住。
17 long result = 0;
(gdb) n <---------- 单条语句执行，next命令简写。
18 for(i=1; i<=100; i++)
输入n之后，是执行赋值还是for语句？

28楼 chj90220 2012-12-11 21:20发表

学习了。呵呵

27楼 hhhggg321 2012-12-03 19:31发表

很好。

26楼 lifj07 2012-10-11 18:15发表

写的通俗易懂~赞！

25楼 Zyong812 2012-09-02 16:15发表

楼主慷慨！！！！！！

24楼 geek_snail 2012-07-28 13:01发表

2003的古文啊！很好的文章！

23楼 straing 2012-07-27 17:14发表

谢谢楼主，之前看的一直有点不懂，把你这篇看完了，突然醒悟了，转载了哈。楼主。

22楼 eis_snail 2012-07-19 14:48发表

如果是一个大的项目中包含很多文件，需不需要写Makefile啊，如果写，这个调试的Makefile于不需要调试的Makefile有什么区别啊？

21楼 wangzhong1979 2012-06-06 14:37发表

原创的七篇文章，博主太给力了！十分感谢！很有帮助。

20楼 xiaoyao3857 2012-06-06 11:38发表

看到网上介绍GDB调试程序的还算少的，博主这篇博客言简意赅，学习起来也容易入门！

19楼 javacookie 2012-05-28 22:07发表

楼主，你好，对于用makefile编译的文件，我应该在哪里加入-g呢

Re: byxbai1989 2012-10-23 16:45发表

回复javacookie：可以直接在Makefile中写，至于详细语法见Makefile。

18楼 sherwinkoo 2012-05-04 18:27发表

学习了，谢谢！

17楼 leiwuhen12 2012-04-26 14:24发表

新人看着很有鸭梨

16楼 tietao 2012-03-19 15:12发表

谢谢博主。

15楼 pauper 2012-02-09 11:42发表

不错，之前一直不看好gdb，现在有点感觉了····

14楼 varyall 2012-02-05 01:43发表

不错，第一次找文章就找到你这里了，顶一下哈，哈哈

13楼 ivension 2012-01-31 10:56发表

多谢你了。。。
对我很有帮助

12楼 deping_chen 2011-10-21 20:24发表

第一次看到在命令行调试。当然WinDbg也是用命令调试。
但在文章中没有看出比IDE调试的强大之处。但我相信这是必然的，毕竟IDE调试还是基于命令行调试。但是很明显，命令行调试很不方便，如非必须，我还是用IDE调试。

11楼 minibee88 2011-02-10 22:31发表

[e01][e01][e01]

10楼 raymond323 2010-09-15 16:42发表

[e03][e04][e05]

9楼 ddskyfuyu 2010-03-31 09:38发表

[e01][e01][e01][e01][e01]

8楼 chenhongyi 2009-07-13 21:51发表

非常感谢陈大虾的贴子,的确受益匪浅.能告诉我到哪里能下载gdb吗?chinaunix提供的gdb软件在aix环境下好象不能用.请多多指教.谢谢.

7楼 wsnpy88 2009-05-21 13:54发表

谢谢，真不错，

6楼 jhj117 2008-11-06 21:43发表

陈先生,你的程序的运算结果应该是31125,而不是31375?差了最后一个250的值.
虽然是小事,但是作程序的严谨来说应该更正..

5楼 zhongyijun159 2008-11-01 08:30发表

这篇文章我确实看到过几遍了 ~

4楼 bluehouse1985 2008-09-24 14:11发表

Linux 环境下的多核调试
— Intel + Totalview 强强联合!
目前，在软件开发行业，各种性能优异的调试工具层出不穷。但是，它们中的绝大部分都只支持windows环境。即使能支持linux平台，操作起来也很不方便。因此，对于长期在linux上编写程序的开发人员来说，如何调试就成了一个令人头痛的问题！Intel软件 和 Totalview Debugger 正是在这种情况下应运而生！
Intel软件可以在英特尔架构上产生出色的应用程序性能，并可以利用最新英特尔多核处理器的各项先进功能。TotalView Debugger与Intel软件的结合将会掀起一场linux下调试工具的革命！
TotalView Debugger是一个linux平台并行环境下的调试工具，它的IDE环境、多线程(进程)调试能力、内存调试能力、集群调试能力在业界都是无与伦比的!
XLsoft携手Intel、TotalView公司于2008年10月30日在上海举行“Linux 环境下的多核调试”免费培训讲座。我们非常荣幸地邀请您参加,并提供免费软件试用光盘！

一、报名方式：
电话：021-62128912/010-84492749
Email:Marketing@xlsoft.com.cn

二、讲座内容:
1. Linux 平台下程序调试工具概述
2. Intel 软件功能介绍
3. Totalview Debugger功能介绍

三、讲座时间：
2008年10月30日（星期四）14：00～17：00

四、讲座地点：
上海青松城大酒店3楼长悦厅
（徐家汇肇家浜路777号东安路口，距衡山路站约15分钟路程）

四、活动详情：
联系人:王娟
Tel:021-62128916 Mobile: 15000262606
E-mail:kiko.wang@xlsoft.com.cn

咨询热线:
021－62128912 010－84492749
更多的服务信息，请联系我们Marketing@xlsoft.com.cn or 联系方式。

上海世全软件信息技术有限公司
联系电话 上海:021－62128912 北京:010－84492749

3楼 finlinden 2008-04-25 17:01发表

我来自台湾，读了阁下的【用GDB调试程序】之后，登时感觉胸中有了许多丘壑，非常感谢，
但那也是在那文章之末，哥哥就曾说过：【会写一篇GDB调试C/C++的进阶使用】，等到现在，一盼几年，尚未谋面
真是期望能早点看到这么一篇进阶读物——哪怕没有理论，只是单纯几个工作中用GDB 来debug的实战用例，相信也会对别人有很大的启发
期待~！！

2楼 gdb_learner 2006-10-27 14:20发表

GNU gdb 6.1
Copyright 2004 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i686-pc-linux-gnu"...Using host libthread_db library

 "/lib/tls/libthread_db.so.1".

(gdb) list
8 sum += i;
9 }
10 return sum;
11 }
12
13
14 int main(void)
15 {
16 int i;
17 long result = 0;
(gdb)
18 for(i=1; i<100; i++)
19 {
20 result += i;
21 }
22
23 printf("result[1-100] = %d /n",result);
24 printf("result[1-250] = %d /n",func(250));
25 }
26

怎么显示的不太一样呀

1楼 鲁不死 2006-07-17 11:32发表

正气十足
强烈支持

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