

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。

目录视图 摘要视图 RSS 订阅

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： 2334644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

文章搜索

文章分类

- 技术趋势 (13)
- 抄袭事件 (7)
- 编程工具 (19)
- 编程语言 (58)
- 职业心情 (23)
- 软件开发 (28)
- 项目管理 (9)

文章存档

- 2011年04月 (1)
- 2011年02月 (3)
- 2010年09月 (1)
- 2010年08月 (2)
- 2010年07月 (5)

展开

阅读排行

- 用GDB调试程序（一） (88102)
- 跟我一起写 Makefile（一） (87915)
- C++ 虚函数表解析 (75463)
- 其实Unix很简单

2013年1月当选微软MVP名单揭晓！ CSDN博客频道年终送好礼获奖名单公布！
2012CSDN博客之星评选正式上线 2000元大奖征异构开发博文 2013年全国百所高校巡讲讲师招募

用GDB调试程序（二）

分类：编程工具 2003-07-03 00:20 23821人阅读 评论(6) 收藏 举报

command shell function unix c++ string

GDB的命令概貌

启动gdb后，就你被带入gdb的调试环境中，就可以使用gdb的命令开始调试程序了，gdb的命令可以使用help命令来查看，如下所示：

```
/home/hchen> 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) help
List of classes of commands:
```

```
aliases -- Aliases of other commands
breakpoints -- Making program stop at certain points
data -- Examining data
files -- Specifying and examining files
internals -- Maintenance commands
obscure -- Obscure features
running -- Running the program
stack -- Examining the stack
status -- Status inquiries
support -- Support facilities
tracepoints -- Tracing of program execution without stopping the program
user-defined -- User-defined commands

Type "help" followed by a class name for a list of commands in that class.
Type "help" followed by command name for full documentation.
Command name abbreviations are allowed if unambiguous.
(gdb)
```

gdb的命令很多，gdb把之分成许多个种类。help命令只是例出gdb的命令种类，如果要看种类中的命令，可以使用help <class> 命令，如：help breakpoints，查看设置断点的所有命令。也可以直接help <command>来查看命令的帮助。

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

gdb中，输入命令时，可以不用打全命令，只用打命令的前几个字符就可以了，当然，命令的前几个字符应该要标志着一个唯一的命令，在Linux下，你可以敲击两次TAB键来补齐命令的全称，如果有重复的，那么gdb会把其例出来。

示例一：在进入函数func时，设置一个断点。可以敲入break func，或是直接就是b func
(gdb) b func
Breakpoint 1 at 0x8048458: file hello.c, line 10.

示例二：敲入b按两次TAB键，你会看到所有b打头的命令：
(gdb) b
backtrace break bt
(gdb)

示例三：只记得函数的前缀，可以这样：
(gdb) b make_ <按TAB键>
(再按下一次TAB键，你会看到：)
make_a_section_from_file make_envirom
make_abs_section make_function_type
make_blockvector make_pointer_type
make_cleanup make_reference_type
make_command make_symbol_completion_list
(gdb) b make_
GDB把所有make开头的函数全部例出来给你查看。

示例四：调试C++的程序时，有可以函数名一样。如：
(gdb) b 'bubble(M-?
bubble(double,double) bubble(int,int)
(gdb) b 'bubble(
你可以查看到C++中的所有的重载函数及参数。(注：M-?和“按两次TAB键”是一个意思)

要退出gdb时，只用发quit或命令简称q就行了。

GDB中运行UNIX的shell程序

在gdb环境中，你可以执行UNIX的shell的命令，使用gdb的shell命令来完成：

shell <command string>
调用UNIX的shell来执行<command string>，环境变量SHELL中定义的UNIX的shell将会被用来执行<command string>，如果SHELL没有定义，那就使用UNIX的标准shell: /bin/sh。(在Windows中使用Command.com或cmd.exe)

还有一个gdb命令是make:

make <make-args>
可以在gdb中执行make命令来重新build自己的程序。这个命令等价于“shell make <make-args>”。

在GDB中运行程序

当以gdb <program>方式启动gdb后，gdb会在PATH路径和当前目录中搜索<program>的源文件。如要确认gdb是否读到源文件，可使用l或list命令，看看gdb是否能列出源代码。

在gdb中，运行程序使用r或是run命令。程序的运行，你有可能需要设置下面四方面的事。

1、程序运行参数。

set args 可指定运行时参数。(如: set args 10 20 30 40 50)
show args 命令可以查看设置好的运行参数。

2、运行环境。

path <dir> 可设定程序的运行路径。
show paths 查看程序的运行路径。
set environment varname [=value] 设置环境变量。如: set env USER=hchen
show environment [varname] 查看环境变量。

3、工作目录。

cd <dir> 相当于shell的cd命令。
pwd 显示当前的所在目录。

4、程序的输入输出。

info terminal 显示你程序用到的终端的模式。
使用重定向控制程序输出。如: run > outfile
tty命令可以指定输入输出的终端设备。如: tty /dev/ttyb

调试已运行的程序

两种方法:

- 1、在UNIX下用ps查看正在运行的程序的PID (进程ID), 然后用gdb <program> PID格式挂接正在运行的程序。
- 2、先用gdb <program>关联上源代码, 并进行gdb, 在gdb中用attach命令来挂接进程的PID。并用detach来取消挂接的进程。

暂停 / 恢复程序运行

调试程序中, 暂停程序运行是必须的, GDB可以方便地暂停程序的运行。你可以设置程序的在哪行停住, 在什么条件下停住, 在收到什么信号时停住等等。以便于你查看运行时的变量, 以及运行时的流程。

当进程被gdb停住时, 你可以使用info program 来查看程序的是否在运行, 进程号, 被暂停的原因。

在gdb中, 我们可以有以下几种暂停方式: 断点 (BreakPoint)、观察点 (WatchPoint)、捕捉点 (CatchPoint)、信号 (Signals)、线程停止 (Thread Stops)。如果要恢复程序运行, 可以使用c或是continue命令。

一、设置断点 (BreakPoint)

我们用break命令来设置断点。正面有几点设置断点的方法:

break <function>

在进入指定函数时停住。C++中可以使用class::function或function(type, type)格式来指定函数名。

break <linenum>

在指定行号停住。

break +offset

break -offset

在当前行号的前面或后面的offset行停住。offset为自然数。

break filename:linenum

在源文件filename的linenum行处停住。

`break filename:function`

在源文件filename的function函数的入口处停住。

`break *address`

在程序运行的内存地址处停住。

`break`

break命令没有参数时，表示在下一条指令处停住。

`break ... if <condition>`

...可以是上述的参数，condition表示条件，在条件成立时停住。比如在循环环境中，可以设置

`break if i=100`，表示当i为100时停住程序。

查看断点时，可使用info命令，如下所示：（注：n表示断点号）

`info breakpoints [n]`

`info break [n]`

二、设置观察点 (WatchPoint)

观察点一般来观察某个表达式（变量也是一种表达式）的值是否有变化了，如果有变化，马上停住程序。我们有下列的几种方法来设置观察点：

`watch <expr>`

为表达式（变量）expr设置一个观察点。一旦表达式值有变化时，马上停住程序。

`rwatch <expr>`

当表达式（变量）expr被读时，停住程序。

`awatch <expr>`

当表达式（变量）的值被读或被写时，停住程序。

`info watchpoints`

列出当前所设置的所有观察点。

三、设置捕捉点 (CatchPoint)

你可设置捕捉点来捕捉程序运行时的一些事件。如：载入共享库（动态链接库）或是C++的异常。设置捕捉点的格式为：

`catch <event>`

当event发生时，停住程序。event可以是下面的内容：

1、throw 一个C++抛出的异常。（throw为关键字）

2、catch 一个C++捕捉到的异常。（catch为关键字）

3、exec 调用系统调用exec时。（exec为关键字，目前此功能只在HP-UX下有用）

4、fork 调用系统调用fork时。（fork为关键字，目前此功能只在HP-UX下有用）

5、vfork 调用系统调用vfork时。（vfork为关键字，目前此功能只在HP-UX下有用）

6、load 或 load <libname> 载入共享库（动态链接库）时。（load为关键字，目前此功能只在HP-UX下有用）

7、unload 或 unload <libname> 卸载共享库（动态链接库）时。（unload为关键字，目前此功能只在HP-UX下有用）

`tcatch <event>`

只设置一次捕捉点，当程序停住以后，应点被自动删除。

(版权所有, 转载时请注明作者和出处)

上一篇: [用GDB调试程序 \(一\)](#)

分享到:

下一篇: [用GDB调试程序 \(五\)](#)

查看评论

6楼 [syzcch](#) 2012-11-08 14:10发表



只能膜拜了 总结的非常到位

5楼 [hu_songsong](#) 2012-10-23 10:13发表



光看不行, 看了又忘了。还是要多练习练习。

4楼 [wodexinmingzi](#) 2012-02-20 20:33发表



例子太少

3楼 [ddskyfuyu_bjtu](#) 2010-03-24 22:32发表



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

但希望有些能多列出几个例子[e04][e04][e04]

2楼 [shenyan008](#) 2009-11-25 19:45发表



"gdb会在PATH路径和当前目录中搜索<program>的源文件"

这里好像有问题, show envirement PATH, 返回的是可执行文件路径, show directories 才显示通过dir <direcotry name>添加的目录, 但在环境变量中没有找到和这个路径相关的变量。不知道存在于哪里?

1楼 [bluehouse1985](#) 2008-09-24 14:20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

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

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#)

京 ICP 证 070598 号

北京创新乐知信息技术有限公司 版权所有

世纪乐知(北京)网络技术有限公司 提供技术支持

江苏乐知网络技术有限公司 提供商务支持

✉ 联系邮箱: [webmaster\(at\)csdn.net](mailto:webmaster(at)csdn.net)

Copyright © 1999-2012, CSDN.NET, All Rights Reserved 