国嵌**嵌入式课程学习笔记文档**

# 网络编程学习笔记

## 第二课 gdb的调试

第二课 gdb的调试

编译c代码

gcc -g test.c -o test

(其中-g这个命令是给调试用的，如果要调试就要加-g，-o指明编译出的文件名称)

启动GDB

gdb test

3. 在main函数处设置断点

break main

运行程序

run

[root@localhost lesson2]# gcc -g gdb.c 
root@localhost lesson2]# Is 
gdb.c 
root@localhost lesson2]# gdb gdb 
-o gdb 
GNU gdb (GDB) Red Hat Enterprise Linux (7.2-56.e16) 
Copyright (C) 2010 Free Software Foundation, Inc. 
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/license 
This is free software: you are free to change and redistribute it. 
There is NO WARRANTY, to the extent permitted by law. 
Type "show 
and "show warranty" for details. 
This GD8 was configured as "i686-redhat-linux-gnu" 
For bug reporting instructions, please see: 
<http : ,//vm•l.gnu.org/software/gdb/bugs/>. . 
Reading symbols from /home/S3-ARM/network/lesson2/gdb... done. 
(gdb) break main 
Breakpoint 1 at Ox80483e9: file gdb.c, line 11. 
(gdb) run 
Starting program: /home/S3-ARM/network/lesson2/gdb 
Breakpoint 1, main ( ) at gdb.c 11 
11 
long result 
Missing separate debuginfos, 
(gdb) 
use: debuginfo-install glibc-2.12-1.8 

[network/les. 
Zinclude <stdio . 
void myprintf(long res) 
printf( 
",res ) , 
void main() 
int i; 
long result 
result -F 1; 
myprintf ( result) ; 
"gdb.c" 
17L, 
184C 

运用更加丰富的GDB命令对程序进行调试

|  |  |
| --- | --- |
|  | GDB命令 |

list (l) 查看函数

break (b) 函数名 在某函数入口处添加断电

break (b) 行号 在指定行添加断点

break (b) 文件名：行号 在指定文件的指定行添加断点

info break 查看所有设置的断点

delete 断点编号 删除断点

next (n) 单步运行程序（不进入子函数）

step (s) 单步运行程序（进入子函数）

continue ( c ) 继续运行程序

print (p) 查看指定的变量值

set var=value 设置变量的值

quit (q) 退出gdb

在网上查了一篇文章：http://blog.csdn.net/liigo/article/details/582231/

GDB的命令很多，但我们只需要掌握10个常用的命令就可以完成调试

|  |  |  |
| --- | --- | --- |
| **命令** | **解释** | **示例** |
| file <文件名> | 加载被调试的可执行程序文件。  因为一般都在被调试程序所在目录下执行GDB，因而文本名不需要带路径。 | (gdb) file gdb-sample |
| r | Run的简写，运行被调试的程序。  如果此前没有下过断点，则执行完整个程序；如果有断点，则程序暂停在第一个可用断点处。 | (gdb) r |
| c | Continue的简写，继续执行被调试程序，直至下一个断点或程序结束。 | (gdb) c |
| b <行号>  b <函数名称>  b \*<函数名称>  b \*<代码地址>  d [编号] | b: Breakpoint的简写，设置断点。两可以使用“行号”“函数名称”“执行地址”等方式指定断点位置。  其中在函数名称前面加“\*”符号表示将断点 设置在“由编译器生成的prolog代码处”。如果不了解汇编，可以不予理会此用法。  d: Delete breakpoint的简写，删除指定编号的某个断点，或删除所有断点。断点编号从1开始递增。 | (gdb) b 8  (gdb) b main  (gdb) b \*main  (gdb) b \*0x804835c  (gdb) d |
| s, n | s: 执行一行源程序代码，如果此行代码中有函数调用，则进入该函数；  n: 执行一行源程序代码，此行代码中的函数调用也一并执行。  s 相当于其它调试器中的“Step Into (单步跟踪进入)”；  n 相当于其它调试器中的“Step Over (单步跟踪)”。  这两个命令必须在有源代码调试信息的情况下才可以使用（GCC编译时使用“-g”参数）。 | (gdb) s  (gdb) n |
| si, ni | si命令类似于s命令，ni命令类似于n命令。所不同的是，这两个命令（si/ni）所针对的是汇编指令，而s/n针对的是源代码。 | (gdb) si  (gdb) ni |
| p <变量名称> | Print的简写，显示指定变量（临时变量或全局变量）的值。 | (gdb) p i  (gdb) p nGlobalVar |
| display ...  undisplay <编号> | display，设置程序中断后欲显示的数据及其格式。  例如，如果希望每次程序中断后可以看到即将被执行的下一条汇编指令，可以使用命令  “display /i $pc”  其中 $pc 代表当前汇编指令，/i 表示以十六进行显示。当需要关心汇编代码时，此命令相当有用。  undispaly，取消先前的display设置，编号从1开始递增。 | (gdb) display /i $pc  (gdb) undisplay 1 |
| i | Info的简写，用于显示各类信息，详情请查阅“help i”。 | (gdb) i r |
| q | Quit的简写，退出GDB调试环境。 | (gdb) q |
| help [命令名称] | GDB帮助命令，提供对GDB名种命令的解释说明。  如果指定了“命令名称”参数，则显示该命令的详细说明；如果没有指定参数，则分类显示所有GDB命令，供用户进一步浏览和查询。 | (gdb) help display |

*来自 <*[*http://blog.csdn.net/liigo/article/details/582231/*](http://blog.csdn.net/liigo/article/details/582231/)*>*