GDB Rocks!

Kent Chen

GDB

The GNU Project Debugger

- Kent Chen (chenkaie)
- chenkaie@gmail.com
- http://chenkaie.blogspot.com
- @chenkaie on GitHub
- @chenkaie on SlideShare
- @chenkaie on LinkedIn
- @chenkaie on Twitter

為什麼要學 GDB

Why everybody learns GDB?

非互動式/交談式

Non-Interactive Debugging

strace - system call, signal

ltrace - library call

printf / printk

“打印”久了

也挺煩人的

Debugging by Endless Printing

GDB

Source-Level

Debugger

互動式/交談式
你叫它幹麻它就幹麻

Interactive Debugging

有了Debugger

Coding是彩色的

- by Jserv/宅色夫大大

No Debugger, No Happy Coding

學會了GDB
我有種山頂洞人
學會用火的感動

- by 張至

張至是誰?! 我也不認識， Google到的，某某鄉民吧!

GDB

Front Ends

gdbtui

Shell - Konsole

```
/home/hamberg/Prosjekter/yzis/qyzis/main.cpp
40      int main(int argc, char **argv) {
41              QApplication app(argc, argv);
42
43              app.setOrganizationName("Yzis");
44              app.setOrganizationDomain("yzis.org");
45              app.setApplicationName("QYzis");
46
47              setlocale( LC_ALL, "");
48
49      #ifndef YZIS_WIN32_GCC
50              bindtextdomain( "yzis", QString("%1%2").arg( PREFIX ).arg("
51              bind_textdomain_codeset( "yzis", "UTF-8" );
52              textdomain( "yzis" );
exec No process In:                                    Line: ??   PC: 0x0
Copyright (C) 2006 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 "i486-linux-gnu"...
Using host libthread_db library "/lib/tls/i686/cmov/libthread_db.so.1".
(gdb)
```

Shell

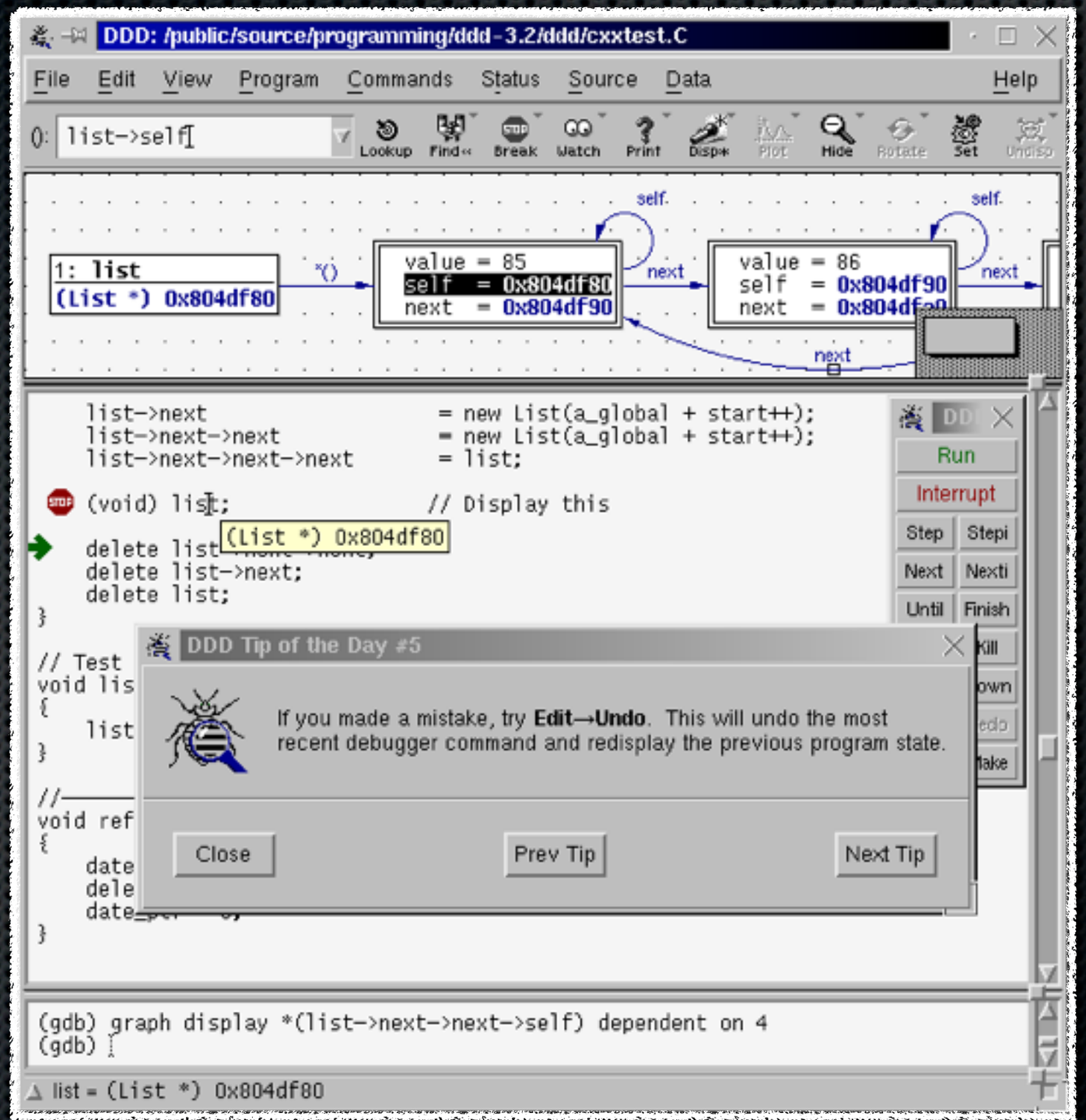
cgdb

rxvt

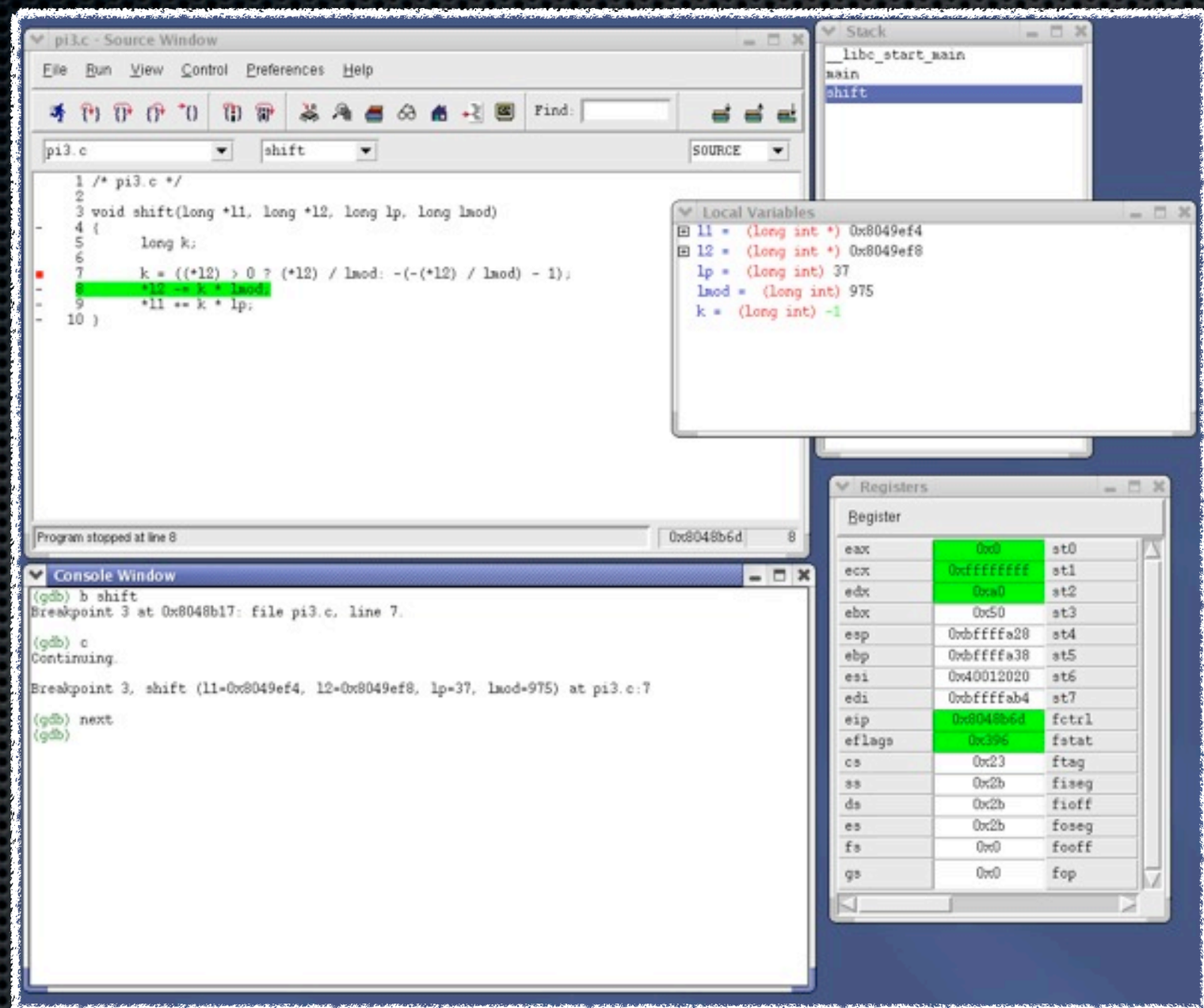
```
1505  int
1506  main (int argc, char *argv[])
1507  {
1508  /* Uncomment to debug and attach */
1509  #if 0
1510    int c;
1511    read (0, &c, 1);
1512  #endif
1513
1514    parse_long_options (&argc, &argv);
1515
1516 -> current_line = ibuf_init ();
1517
1518    if (create_and_init_pair () == -1)
1519      {
1520        fprintf (stderr, "%s:%d Unable to create PTY pair", __FILE__, __LINE
1521        exit (-1);
1522      }
1523
1524    /* First create tgdb, because it has the error log */
1525    if (start_gdb (argc, argv) == -1)
1526      {
/home/mike/dev/cgdb/cgdb/src/cgdb.c
0x00002b4ea9b2fea3 in select () from /lib/libc.so.6
(tgdb)
Breakpoint 2 at 0x4041c1: file cgdb.c, line 1514.
(tgdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
The program being debugged has been started already.
Starting program: /home/mike/dev/cgdb/cgdb/src/cgdb

Breakpoint 2, main (argc=0, argv=0x7fffa6065720) at cgdb.c:1516
(tgdb) p argc
$1 = 0
(tgdb)
```

ddd (Joe's Fav)


DDD: /public/source/programming/ddd-3.2/ddd/cxxtest.C
File Edit View Program Commands Status Source Data Help
list->self
Lookup Find« Break Watch Print Disp« Plot Hide Rotate Set Undisp
1: list
(List *) 0x804df80
value = 85
self = 0x804df80
next = 0x804df90
value = 86
self = 0x804df90
next
list->next = new List(a_global + start++);
list->next->next = new List(a_global + start++);
list->next->next->next = list;
(void) list; // Display this
(List *) 0x804df80
delete list->next;
delete list;
Run
Interrupt
Step Stepi
Next Nexti
Until Finish
DDD Tip of the Day #5
If you made a mistake, try Edit→Undo. This will undo the most recent debugger command and redisplay the previous program state.
Close
Prev Tip
Next Tip
(gdb) graph display *(list->next->next->self) dependent on 4
(gdb)
list = (List *) 0x804df80

insight


pi3.c - Source Window
File Run View Control Preferences Help
Find:
pi3.c
shift
SOURCE
1 /* pi3.c */
2
3 void shift(long *l1, long *l2, long lp, long lmod)
4 {
5 long k;
6
7 k = ((*l2) > 0 ? (*l2) / lmod: -(-(*l2) / lmod) - 1);
8 *l2 -= k * lmod;
9 *l1 += k * lp;
10 }
Program stopped at line 8
0x8048b6d
8
Stack
__libc_start_main
main
shift
Local Variables
l1 = (long int *) 0x8049ef4
l2 = (long int *) 0x8049ef8
lp = (long int) 37
lmod = (long int) 975
k = (long int) -1
Registers
Register
eax 0x0 st0
ecx 0xffffffff st1
edx 0xa0 st2
ebx 0x50 st3
esp 0xbffffa28 st4
ebp 0xbffffa38 st5
esi 0x40012020 st6
edi 0xbffffab4 st7
eip 0x8048b6d fctrl
eflags 0x396 fstat
cs 0x23 ftag
ss 0x2b fiseg
ds 0x2b fioff
es 0x2b foseg
fs 0x0 fooff
gs 0x0 fop
Console Window
(gdb) b shift
Breakpoint 3 at 0x8048b17: file pi3.c, line 7.
(gdb) c
Continuing.
Breakpoint 3, shift (l1=0x8049ef4, l2=0x8049ef8, lp=37, lmod=975) at pi3.c:7
(gdb) next
(gdb)

clewn / vim + gdb

```
#include <execinfo.h>
#include <stdio.h>
#include <stdlib.h>

/* Obtain a backtrace and print it to stdout. */
void print_trace (void)
{
    void *array[10];
    size_t size;
    char **strings;
    size_t i;

    size = backtrace (array, 10);
    strings = backtrace_symbols (array, size);

    printf ("Obtained %zd stack frames.\n", size);

    for (i = 0; i < size; i++)
        printf ("%s\n", strings[i]);

    free (strings);

/* A dummy function to make the backtrace more interesting. */
void dummy_function (void)
{
    print_trace ();
}

int main (void)
{
    dummy_function ();
    return 0;
}

File: backtrace.c [unix,utf-8,C]  [AscII=125] [Hex=7D] [Pos=22,1,62%] [LINE=35]
1: size = 5
2: strings = (char **) 0x601660
4: **strings = 47 '/'
5: i = 5
```

```
    0x0,
    0x3ac7819bc0,
    0x400670,
    0x400473}
(gdb)
(gdb) help gdb
  Undefined command: "gdb".  Try "help".
(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 cla>
  Type "help all" for the list of all commands.
  Type "help" followed by command name for full documentation.
  Type "apropos word" to search for commands related to "word".
  Command name abbreviations are allowed if unambiguous.
(gdb) step
  /home/kent/practice/GDB/kent/backtrace [0x400656]
(gdb) step
(gdb) next
  /lib64/libc.so.6(__libc_start_main+0xf4) [0x3ac7a1d8a4]
(gdb) next
(gdb) next
  /home/kent/practice/GDB/kent/backtrace [0x400519]
(gdb) next
(gdb) print strings
  $23 = (char **) 0x601660
(gdb) print *strings
  $24 = 0x601688 "/home/kent/practice/GDB/kent/backtrace [0x4005de]"
(gdb) print i
  $25 = 5
(gdb)
(gdb)
(gdb)
(gdb) break "/home/kent/practice/GDB/kent/backtrace.c:14"
  Breakpoint 3 at 0x4005e4: file backtrace.c, line 14.
(gdb) break "/home/kent/practice/GDB/kent/backtrace.c:32"
  Breakpoint 4 at 0x400651: file backtrace.c, line 32.
(gdb)
(gdb)
File: gdb-variables [unix,utf-8,]  [AscII=053] [Hex=35] [Pos=4,1,100%] [LINE=4]  File: 4 [unix,utf-8,] [-]  [AscII=040] [Hex=28] [Pos=127,1,100%] [LINE=127]
5: void print_trace (void)
1 VS8102  2 DM365  3 dm365  4 hash  5 VC220P  [6] GDB  7 Dotfiles  8 MISC  9 IRC  10 Top  11 OProfile
```

pyclewn

```
foo.c (~/src/pyclewn_wa/trunk/pyclewn/testsuite) - GVIM
(gdb) step
(gdb) step
(gdb) foldvar 3
(gdb) backtrace
#0  foo (ptr=0xbf865b30 "'a' letter in /home/xavier/src/pyclewn_wa/trunk/py
#1  0x080484f5 in main (argc=1, argv=0xbf865c44) at foobar.c:60
(gdb)»
(clewn)_console [RO]                                    121,6          Bot
[-] var1: (map_t) map =    {...}
   *  var1.key  : (int   ) key   =    1
  [-] var1.value: (char *) value =    0xbf865b30 "'a' letter in /home/xavie
     *  var1.value.*value: (char) *value =    39 '\''
(clewn)_dbgvar [RO]                                     2,1            All
{
    map_t map;
bp  int count = 0;

    map.value = ptr;
    do {
·       map.key = count + 1;
=>      count = bar(&(map.key), map.value + 1);
bp      printf("count: %d\n", count);
·       msleep(100);
    } while (max < 0 || count < max);
}
~/src/pyclewn_wa/trunk/pyclewn/testsuite/foo.c          28,1           Bot
```

gdbmgr

```
prog3.c - (~/.vim/gdbmgr/progs) - VIM
prog3.c  7 prog3.c
" ========================
" Netrw Directory Listing
"   /home/lainme/.vim/gdbm
"   Sorted by      name
"   Sort sequence: [\/]$,\
"   Hiding:        ^\..*
"   Quick Help: <F1>:help
" ========================
../
progs/
| core
| prog1.c
| prog2.c
| prog3.c
| prog4.c
| prog5.c
| prog6.c
| prog7.c
| prog8.c
| prog9.c
| Makefile
| prog1*
| prog2*
| prog3*
| prog4*
| prog5*
< [-] Win#1 Posn[9,1] 全部

  3 starting
  4 breakpoint 1
  5 starting
  6 stopped
Messages [-] Win#2 Posn[6,1] 底端

  1 #0  main (argc=1, argv=0x7fffffffdaa8) at pro
    g3.c:56
FuncStack [-] Win#3 Posn[1,1] 全部

 42
 43 /* ------------------------------------------------------------------- */
 44 /* main: {{{2 */
 45 int main(
 46   int    argc,
 47   char **argv)
 48 {
 49 int x[5];
 50 int y;
 51 int z[5]= {10,20,30,40,50};
 52 int i;
 53
 54 for(i= 0; i < 5; ++i) {
 55         x[i]= z[i];
=> 56         if(i == 3) y= x[i];
 57         }
 58
 59 return 0;
 60 }
prog3.c [-] Win#4 Posn[51,1] 89%

  1 i = 3
Expressions  Win#5 Posn[1,5] 全部

  1
<eakpoints [-] Win#6 Posn[0,0] 全部

  1 w x[3] = 40
Watchpoints  Win#7 Posn[1,1] 全部
```

分享小弟
入門經驗

Sharing my real-world GDB experience

牛刀小試

幼幼班

GDB Beginner’s training

Change memory contents on-the-fly

```
172.16.5.31
1 #include <stdio.h>
2
3 static char *my_str = "Hello World!";
4 int main()
5 {
6     puts(my_str);
=> 7     return 0;
8 }
~
~
~
```

Change memory contents on-the-fly

1. Default (ssh)

```
~|
1| #include <stdio.h>
2|
3| static char buff [256];
4| static char* string;
5|
6| int main ()
7| {
8|     printf ("Please input a string: ");
9|     gets (string);
10|     printf ("\nYour string is: %s\n", string);
11|     return 0;
12| }
 ~|
 ~|
/home/kent/tmp/sample1/input.c
Start it from the beginning? (y or n) y
The program being debugged has been started already.
Starting program: /home/kent/tmp/sample1/input.out
warning: no loadable sections found in added symbol-file system-supplied DSO at 0x2aaaaaaab000

Breakpoint 1, main () at input.c:8
(gdb) set variable string="str"
(gdb) p string
$4 = 0x601010 "str"
(gdb) set $pc=main+31
(gdb) next

Your string is: str
(gdb)
[1] gdb-rocks   2 gdb interpreter   3 Tab3   4 bash          [rd1-2 | 0.21 0.40 0.47 | 03/05 23:33]
```

stack backtrace

```
172.16.5.31

1  #include <stdio.h>
2  void func4() { puts("Output"); }
=> 3  void func3() { func4(); }
4  void func2() { func3(); }
5  void func1() { func2(); }
6
7  int main()
8  {
9      int ret = 0;
10     func1();
11     return ret;
12 }
~
~
~
~
~
~
~
~
~
~
File: <C]  [AscII=118] [Hex=76] [Pos=3,1,25%] [LINE=12]
1
~
~
~
~
~
~
~
File: <,]  [AscII=000] [Hex=00] [Pos=0,1,100%] [LINE=1]

188  ss             0x2b   43
189  ds             0x0    0
190  es             0x0    0
191  fs             0x0    0
192  gs             0x0    0
193  fctrl          0x37f  895
194  fstat          0x0    0
195  ftag           0xffff 65535
196  fiseg          0x0    0
197  fioff          0x0    0
198  foseg          0x0    0
199  fooff          0x0    0
200  fop            0x0    0
201  mxcsr          0x1f80 [ IM DM ZM OM UM PM ]
202 (gdb) info frame
203   Stack level 0, frame at 0x7fffffffcd20:
204    rip = 0x40049c in func4 (stack-backtrace.c:2);>
205    called by frame at 0x7fffffffcd30
206    source language c.
207    Arglist at 0x7fffffffcd10, args:
208    Locals at 0x7fffffffcd10, Previous frame's sp >
209    Saved registers:
210     rbp at 0x7fffffffcd10, rip at 0x7fffffffcd18
211 (gdb) up
212   #1  0x00000000004004b6 in func3 () at stack-bac>
213 (gdb) backtrace
214   #0  func4 () at stack-backtrace.c:2
215   #1  0x00000000004004b6 in func3 () at stack-bac>
216   #2  0x00000000004004c6 in func2 () at stack-bac>
217   #3  0x00000000004004d6 in func1 () at stack-bac>
218   #4  0x00000000004004f1 in main () at stack-back>
219 (gdb)
File: <AscII=040] [Hex=28] [Pos=219,1,100%] [LINE=219]

1 SD81x1  2 SD81x1  3 TC-SIMS  4 VS-SIMS  5 VS-SIMS  6 CISCO  [7] practice  8 dotfiles  9 test  10 Top  [rd
```

Attach to a process

172.16.5.31

```
1  #include <stdio.h>
2  #include <unistd.h>
3  int main(int argc, char **argv)
4  {
5      printf("Hello World!\n");
6      while (1) {
7          sleep(5);
8          printf("Hello World!\n");
9      }
10     return 0;
11 }
~
~
~
~
~
~
~
~
~
~
File: <C]  [AscII=101] [Hex=65] [Pos=6,9,54%] [LINE=11]
1
File: <,]  [AscII=000] [Hex=00] [Pos=0,1,100%] [LINE=1]
```

```
8   and "show warranty" for details.
9   This GDB was configured as "x86_64-unknown-linu>
10  For bug reporting instructions, please see:
11  <http://www.gnu.org/software/gdb/bugs/>.
12 (gdb) file ./a.out
13  Reading symbols from /home/kent/practice/GDB/sa>
14 (gdb) attach 1841
15  Attaching to program: /home/kent/practice/GDB/s>
16  Reading symbols from /lib64/libc.so.6...(no deb>
17  Loaded symbols for /lib64/libc.so.6
18  Reading symbols from /lib64/ld-linux-x86-64.so.>
19  Loaded symbols for /lib64/ld-linux-x86-64.so.2
20  0x0000003ac7a954e0 in __nanosleep_nocancel () f>
21 (gdb) step
22  Single stepping until exit from function __nano>
23  which has no line number information.
24  0x0000003ac7a95334 in sleep () from /lib64/libc>
25 (gdb) step
26  Single stepping until exit from function sleep,
27  which has no line number information.
28  main (argc=1, argv=0x7fff8ae2ed48) at hello-loo>
29 (gdb) continue
30  Continuing.
~
File: < [AscII=032] [Hex=20] [Pos=30,1,100%] [LINE=30]
```

```
@495 attach 1184
@496 bt
@497 b printf
@498 info breakpoints
@499 b 9
@500 attach 1841
@501
File: [Command Line] [unix,utf-8,GDB]  [AscII=000] [Hex=00] [Pos=501,1,100%] [LINE=501]
-- INSERT --
1 SD81x1  2 SD81x1  3 TC-SIMS  4 VS-SIMS  5 VS-SIMS  6 CISCO  7 practice  [8] dotfiles  9 test  10 Top  [rd
```

Jump $pc (program counter)

```
1. Default (ssh)
11| dummy_function (void)
12| {
13|     unsigned char *ptr = 0x00;
14|     test_printf();
15|     fprintf(stderr, "[%s] %d\n", __FUNCTION__, MAX_TMP);
16|     *ptr = 0x00;
17|     fprintf(stderr, "[%s] %c\n", __FUNCTION__, ptr);
18|     fprintf(stderr, "[%s] >>><<< \n", __FUNCTION__);
/home/kent/Junkcode/GDB/gdb-rocks/jump.c
0x0000000000400562 <dummy_function+48>: callq  0x400430 <fprintf@plt>
0x0000000000400567 <dummy_function+53>: mov    -0x8(%rbp),%rax
0x000000000040056b <dummy_function+57>: movb   $0x0,(%rax)
0x000000000040056e <dummy_function+60>: mov    0x2004a3(%rip),%rdi        # 0x600a18 <stderr@@GLIBC_2.2.5>
0x0000000000400575 <dummy_function+67>: mov    -0x8(%rbp),%rcx
0x0000000000400579 <dummy_function+71>: mov    $0x4006dc,%edx
0x000000000040057e <dummy_function+76>: mov    $0x4006f4,%esi
0x0000000000400583 <dummy_function+81>: mov    $0x0,%eax
0x0000000000400588 <dummy_function+86>: callq  0x400430 <fprintf@plt>
0x000000000040058d <dummy_function+91>: mov    0x200484(%rip),%rdi        # 0x600a18 <stderr@@GLIBC_2.2.5>
0x0000000000400594 <dummy_function+98>: mov    $0x4006dc,%edx
0x0000000000400599 <dummy_function+103>:        mov    $0x4006fd,%esi
0x000000000040059e <dummy_function+108>:        mov    $0x0,%eax
0x00000000004005a3 <dummy_function+113>:        callq  0x400430 <fprintf@plt>
0x00000000004005a8 <dummy_function+118>:        leaveq
0x00000000004005a9 <dummy_function+119>:        retq
End of assembler dump.
(gdb) p $pc=dummy_function+60
$2 = (void (*)()) 0x40056e <dummy_function+60>
(gdb) next
(gdb)
[1] gdb-rocks   2 gdb interpreter   3 Tab3   4 bash          [rd1-2 | 0.35 0.50 0.50 | 03/05 22:38]
```

core dump

```
172.16.5.31
core.2803.ptr1          ibm_divide_zero*  ptr1*                    ptr1-static*
[kent]@rd1-2:kent$ ls
a.out*                  core.31311        ibm_divide_zero.c        ptr1-arm-debug*           ptr1-stripped*
backtrace*              cscope.in.out     ibm_divide_zero-dm365*   ptr1-arm-debug-stripped*  ptr_objdump-D
backtrace.c             cscope.out        Makefile                 ptr1-arm-nondebug*        ptr_objdump-DS
backtrace.out*          cscope.po.out     max_crypt*               ptr1.asm                  tags
backtrace.out-dm365*    dump1*            max_crypt.c              ptr1.c                    thread*
core.25295              dump1.c           memorylayout.c           ptr1-grpof*               thread.c
core.26172              gdb.txt           memorylayout-dm365*      ptr1-nondebug*
core.2803.ptr1          ibm_divide_zero*  ptr1*                    ptr1-static*
[kent]@rd1-2:kent$ ./ib
ibm_divide_zero         ibm_divide_zero-dm365
[kent]@rd1-2:kent$ ./ibm_divide_zero
Floating point exception
[kent]@rd1-2:kent$ ul
ul      ulimit
[kent]@rd1-2:kent$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
max nice                        (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 38912
max locked memory       (kbytes, -l) 32
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
max rt priority                 (-r) 0
stack size              (kbytes, -s) 10240
cpu time               (seconds, -t) unlimited
max user processes              (-u) 38912
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
[kent]@rd1-2:kent$ ulimit -c unlimited
1 SD81x1  2 SD81x1  3 TC-SIMS  4 VS-SIMS  5 VS-SIMS  6 CISCO  [7] practice  8 dotfiles  9 test  10 Top  [rd
```

core dump (cont.)

```
172.16.5.31
-rwxr-xr-x 1 kent rd1 8.4K 2010-05-10 11:37 max_crypt*
-rwxr-xr-x 1 kent rd1 9.8K 2010-05-10 11:37 ptr1*
-rwxr-xr-x 1 kent rd1 9.3K 2010-05-10 11:37 thread*
-rw-r--r-- 1 kent rd1  673 2010-05-10 11:37 dump1.c
-rw-r--r-- 1 kent rd1 1.6K 2010-05-10 11:37 tags
-rw-r--r-- 1 kent rd1  510 2010-05-10 11:31 Makefile
[kent]@rd1-2:kent$ gdb ./ib
ibm_divide_zero        ibm_divide_zero-dm365
[kent]@rd1-2:kent$ gdb ./ibm_divide_zero ./cor
core.12375      core.25295      core.26172      core.2803.ptr1  core.31311
[kent]@rd1-2:kent$ gdb ./ibm_divide_zero ./core.12375
GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/kent/practice/GDB/kent/ibm_divide_zero...done.
Reading symbols from /lib64/libc.so.6...(no debugging symbols found)...done.
Loaded symbols for /lib64/libc.so.6
Reading symbols from /lib64/ld-linux-x86-64.so.2...(no debugging symbols found)...done.
Loaded symbols for /lib64/ld-linux-x86-64.so.2
Core was generated by `./ibm_divide_zero'.
Program terminated with signal 8, Arithmetic exception.
#0  0x00000000004004b5 in wib (no1=8, no2=8) at ibm_divide_zero.c:7
7           result = no1 / diff;
(gdb) p diff
$1 = 0
(gdb) p no1
$2 = 8
(gdb)
1 SD81x1  2 SD81x1  3 TC-SIMS  4 VS-SIMS  5 VS-SIMS  6 CISCO  [7] practice  8 dotfiles  9 test  10 Top  [rd
```

core dump (cont.)

```
172.16.5.31
-rw------- 1 kent rd1 152K 2010-05-10 11:39 core.12375
-rw-r--r-- 1 kent rd1  673 2010-05-10 11:37 dump1.c
-rw-r--r-- 1 kent rd1 1.6K 2010-05-10 11:37 tags
-rw-r--r-- 1 kent rd1  510 2010-05-10 11:31 Makefile
[kent]@rd1-2:kent$ ./pt
ptr1                    ptr1-arm-debug-stripped  ptr1-grpof              ptr1-static
ptr1-arm-debug          ptr1-arm-nondebug        ptr1-nondebug           ptr1-stripped
[kent]@rd1-2:kent$ pwd
/home/kent/practice/GDB/kent
[kent]@rd1-2:kent$ ./p
ptr1                    ptr1-arm-debug-stripped  ptr1-grpof              ptr1-static
ptr1-arm-debug          ptr1-arm-nondebug        ptr1-nondebug           ptr1-stripped
[kent]@rd1-2:kent$ ./ptr1
Segmentation fault (core dumped)
[kent]@rd1-2:kent$ gdb ./ptr1 core.27358
GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licens
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 GDB was configured as "x86_64-unknown-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/kent/practice/GDB/kent/ptr1...done.
Reading symbols from /lib64/libc.so.6...(no debugging symbols found)...done.
Loaded symbols for /lib64/libc.so.6
Reading symbols from /lib64/ld-linux-x86-64.so.2...(no debugging symbols found)...done.
Loaded symbols for /lib64/ld-linux-x86-64.so.2
Core was generated by `./ptr1'.
Program terminated with signal 11, Segmentation fault.
#0  0x00000000004004ec in dummy_function () at ptr1.c:9
9           *ptr = 0x00;
(gdb)
1 SD81x1  2 SD81x1  3 TC-SIMS  4 VS-SIMS  5 VS-SIMS  6 CISCO  [7] practice  8 dotfiles  9 test  10 Top  [rd
```

Patch binary file

```
1. Default (ssh)
30| void dummy_function (void)
31| {
32|     char die = 1;
33|     if (die)
34|     {
35|         printf("DEAD!\n");
36|         unsigned char *ptr = 0x00;
/home/kent/Junkcode/GDB/gdb-rocks/dump1.c
0x000000000040087a <dummy_function+8>:  movb   $0x1,-0x11(%rbp)
0x000000000040087e <dummy_function+12>: cmpb   $0x0,-0x11(%rbp)
0x0000000000400882 <dummy_function+16>: je     0x40089f <dummy_function+45>
0x0000000000400884 <dummy_function+18>: mov    $0x400a2b,%edi
0x0000000000400889 <dummy_function+23>: callq  0x4005a8 <puts@plt>
0x000000000040088e <dummy_function+28>: movq   $0x0,-0x10(%rbp)
0x0000000000400896 <dummy_function+36>: mov    -0x10(%rbp),%rax
0x000000000040089a <dummy_function+40>: movb   $0x0,(%rax)
0x000000000040089d <dummy_function+43>: jmp    0x4008be <dummy_function+76>
0x000000000040089f <dummy_function+45>: mov    $0x400a31,%edi
0x00000000004008a4 <dummy_function+50>: callq  0x4005a8 <puts@plt>
0x00000000004008a9 <dummy_function+55>: mov    $0x1,%edi
0x00000000004008ae <dummy_function+60>: callq  0x4005c8 <malloc@plt>
0x00000000004008b3 <dummy_function+65>: mov    %rax,-0x8(%rbp)
0x00000000004008b7 <dummy_function+69>: mov    -0x8(%rbp),%rax
0x00000000004008bb <dummy_function+73>: movb   $0x0,(%rax)
0x00000000004008be <dummy_function+76>: leaveq
0x00000000004008bf <dummy_function+77>: retq
End of assembler dump.
(gdb) x/i $pc
0x40087a <dummy_function+8>:    movb   $0x1,-0x11(%rbp)
(gdb) x/w $pc
0x40087a <dummy_function+8>:    0x01ef45c6
(gdb)
1 gdb-rocks  2 gdb interpreter  3 Tab3  [4] bash                    [rd1-2 | 0.60 0.54 0.64 | 03/06  0:25]
```

Patch binary file (cont.)

1. Default (ssh)

1 dump1.c 2 dump1.out

```
2960   400868:   bf 00 00 00 00          mov    $0x0,%edi
2961   40086d:   e8 46 fd ff ff          callq  4005b8 <exit@plt>
2962
2963
2964
2965
2966
2967
2968 void dummy_function (void)
2969 {
2970   400872:   55                       push   %rbp
2971   400873:   48 89 e5                 mov    %rsp,%rbp
2972   400876:   48 83 ec 20              sub    $0x20,%rsp
2973 /home/kent/Junkcode/GDB/gdb-rocks/dump1.c:32
2974     char die = 1;
2975   40087a:   c6 45 ef 01              movb   $0x1,0xffffffffffffffef(%rbp)
2976 /home/kent/Junkcode/GDB/gdb-rocks/dump1.c:33
2977     if (die)
2978   40087e:   80 7d ef 00              cmpb   $0x0,0xffffffffffffffef(%rbp)
2979   400882:   74 1b                    je     40089f <dummy_function+0x2d>
2980 /home/kent/Junkcode/GDB/gdb-rocks/dump1.c:35
2981     {
2982         printf("DEAD!\n");
2983   400884:   bf 2b 0a 40 00           mov    $0x400a2b,%edi
2984   400889:   e8 1a fd ff ff           callq  4005a8 <puts@plt>
2985 /home/kent/Junkcode/GDB/gdb-rocks/dump1.c:36
2986         unsigned char *ptr = 0x00;
2987   40088e:   48 c7 45 f0 00 00 00     movq   $0x0,0xfffffffffffffff0(%rbp)
2988   400895:   00
2989 /home/kent/Junkcode/GDB/gdb-rocks/dump1.c:37
2990         *ptr = 0x00;
```

master dump1.out.asm 95% < ln 2974:17

$objdump -d -S -l -shrt dump1.out

```
121 0000780: ff48 8b55 f848 8dbd f0fb ffff be00 0400  .H.U.H..........
122 0000790: 00e8 82fe ffff 4885 c075 0abf 0000 0000  ......H..u......
123 00007a0: e813 feff ff48 8b7d f8e8 8afe ffff 488d  .....H.}......H.
129 0000800: ffb8 0000 0000 fc48 8bbd d8f7 ffff f2ae  .......H........
130 0000810: 4889 c848 f7d0 4883 e801 4883 e801 c684  H..H..H...H.....
131 0000820: 05f0 fbff ff00 e8dd fdff ff48 8d8d f0fb  ...........H....
132 0000830: ffff 488d bdf0 f7ff ff41 89c0 baf0 0940  ..H......A.....@
133 0000840: 00be 0004 0000 b800 0000 00e8 48fd ffff  ............H...
134 0000850: 488d bdf0 f7ff ffe8 4cfd ffff 488d bdf0  H.......L...H...
135 0000860: f7ff ffe8 90fd ffff bf00 0000 00e8 46fd  ..............F.
136 0000870: ffff 5548 89e5 4883 ec20 c645 ef01 807d  ..UH..H.. .E...}
137 0000880: ef00 741b bf2b 0a40 00e8 1afd ffff 48c7  ..t..+.@......H.
138 0000890: 45f0 0000 0000 488b 45f0 c600 00eb 1fbf  E.....H.E.......
139 00008a0: 310a 4000 e8ff fcff ffbf 0100 0000 e815  1.@.............
140 00008b0: fdff ff48 8945 f848 8b45 f8c6 0000 c9c3  ...H.E.H.E......
141 00008c0: 5548 89e5 be28 0740 00bf 0b00 0000 e855  UH...(.@.......U
142 00008d0: fdff ffe8 9aff ffff b800 0000 00c9 c390  ................
143 00008e0: f3c3 0f1f 8000 0000 000f 1f80 0000 0000  ................
144 00008f0: 4c89 6424 e04c 896c 24e8 4c8d 2543 0220  L.d$.L.l$.L.%C.
145 0000900: 004c 8974 24f0 4c89 7c24 f849 89f6 4889  .L.t$.L.|$.I..H.
146 0000910: 5c24 d048 896c 24d8 4883 ec38 4189 ff49  \$.H.l$.H..8A..I
147 0000920: 89d5 e849 fcff ff48 8d05 1602 2000 4929  ...I...H.... .I)
148 0000930: c449 c1fc 034d 85e4 741e 31ed 4889 c390  .I...M..t.1.H...
149 0000940: 4883 c501 4c89 ea4c 89f6 4489 ffff 1348  H...L..L..D....H
150 0000950: 83c3 0849 39ec 75e8 488b 5c24 0848 8b6c  ...I9.u.H.\$.H.l
151 0000960: 2410 4c8b 6424 184c 8b6c 2420 4c8b 7424  $.L.d$.L.l$ L.t$
```

Change "ef01" to "ef00"

NORMAL master > dump1.out < unix < utf-8 < xxd 4% ln 134:43

search hit BOTTOM, continuing at TOP

奇技淫巧
進階版

Advanced GDB Tricks

奇技淫巧：

奇異而眩人耳目的技能或事物

1. 奇技淫巧
注音一式 ㄑㄧˊ ㄐㄧˋ ㄧㄣˊ ㄑㄧㄠˇ
奇異而眩人耳目的技能或事物。書經•泰誓下：「郊社不修，宗廟不享，作**奇技淫巧**，以悅婦人。」官場現形記•第四十六回：「臣是天朝的大臣，應該按照國家的制度辦事。什麼火車、輪船，走的雖快，總不外乎**奇技淫巧**。」或作「奇伎淫巧」。

(from 教育部國語辭典)

SIGSEGV + GDB

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <string.h>

void dump(int signo)
{
    char buf[1024];
    char cmd[1024];
    FILE *fh;

    snprintf(buf, sizeof(buf), "/proc/%d/cmdline", getpid());
    if(!(fh = fopen(buf, "r")))
        exit(0);
    if(!fgets(buf, sizeof(buf), fh))
        exit(0);
    fclose(fh);
    if(buf[strlen(buf) - 1] == '\n')
        buf[strlen(buf) - 1] = '\0';
    snprintf(cmd, sizeof(cmd), "gdb %s %d", buf, getpid());
    system(cmd);

    exit(0);
}

void dummy_function (void)
{
    unsigned char *ptr = 0x00;
    *ptr = 0x00;
}

int main (void)
{
    signal(SIGSEGV, &dump);
    dummy_function ();

    return 0;
}
```

```
[kent]@rd1-2:kent$ ./dump1
GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-unknown-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/kent/practice/GDB/kent/dump1...done.
Attaching to program: /home/kent/practice/GDB/kent/dump1, process 28980
Reading symbols from /lib64/libc.so.6...(no debugging symbols found)...done.
Loaded symbols for /lib64/libc.so.6
Reading symbols from /lib64/ld-linux-x86-64.so.2...(no debugging symbols found)
Loaded symbols for /lib64/ld-linux-x86-64.so.2
0x0000003ac7a950e5 in waitpid () from /lib64/libc.so.6
(gdb) bt
#0  0x0000003ac7a950e5 in waitpid () from /lib64/libc.so.6
#1  0x0000003ac7a3be83 in do_system () from /lib64/libc.so.6
#2  0x00000000004007d6 in dump (signo=11) at dump1.c:21
#3  <signal handler called>
#4  0x00000000004007f0 in dummy_function () at dump1.c:29
#5  0x000000000040080d in main () at dump1.c:35
(gdb) frame 4
#4  0x00000000004007f0 in dummy_function () at dump1.c:29
warning: Source file is more recent than executable.
29          *ptr = 0x00;
(gdb)
1 SD81x1  2 SD81x1  3 TC-SIMS  4 VS-SIMS  5 VS-SIMS  6 CISCO   [7] practice   8
```

C interpreter

1. $ gdb `which gdb`
2. (gdb) start
3. Enjoy your world…

- Example:

```
(gdb) p 1 + 2 + abs(-3)
(gdb) p strcmp("VIVOTEK", "AXIS")
(gdb) x/s getenv(“HOME”)
(gdb) p (char*)getenv("HOME")
(gdb) p (char)*getenv("HOME")
(gdb) p printf("%d\n", 12345678)
```

Signal Handler

- Terminal hang / Reboot PC
  - You have to close terminal (e.g., PuTTY, iTerm,...)
- Conventional solution
  - GNU Screen / Tmux
  - nohup
- GDB solution
  - $ gdb [program] [pid]
  - (gdb) handle SIGHUP nopass
  - (gdb) continue

```
(gdb) handle SIGHUP
Signal        Stop      Print   Pass to program Description
SIGHUP        Yes       Yes     Yes             Hangup
(gdb) handle SIGHUP nopass
Signal        Stop      Print   Pass to program Description
SIGHUP        Yes       Yes     No              Hangup

Program received signal SIGHUP, Hangup.
0x0000003ac7a954e0 in __nanosleep_nocancel () from /lib64/libc.so.6
(gdb)
Continuing.
```

經典案例

實戰探討

A real-world case study

案例一、

Case 1

GNU C Library (glibc)

debugging / 除錯

Why?

追求卓越

Pursuit of excellence :)

DieLink

呆者虻

江湖中流傳已久

A well-know issue

某某 Daemon 之死

Process crash issue

dmesg

```
configer: unhandled page fault (11) at 0x00000000, code 0x817
pgd = c130c000
[00000000] *pgd=825c0031, *pte=00000000, *ppte=00000000

Pid: 983, comm:             configer
CPU: 0
PC is at 0x400daa54
LR is at 0x400df960
pc : [<400daa54>]    lr : [<400df960>]    Tainted: PF
sp : 40a70cc8  ip : 00000000  fp : 40a712a4
r10: 40a713d4  r9 : 40a712b0  r8 : 0000000e
r7 : 0000003c  r6 : 40a71358  r5 : 40a712b0  r4 : 40a71398
r3 : 00000000  r2 : 00000000  r1 : 0000003c  r0 : 40a712b0
Flags: Nzcv  IRQs on  FIQs on  Mode USER_32  Segment user
Control: 5317F
Table: 8130C000  DAC: 00000015
[<c00364d4>] (show_regs+0x0/0x50) from [<c003b6bc>] (__do_user_fault+0x5c/0xa4)
 r4 = C87F8040
[<c003b660>] (__do_user_fault+0x0/0xa4) from [<c003b978>] (do_page_fault+0x1f8/0x228)
 r7 = C5994AC8  r6 = C87F8040  r5 = C0348A68  r4 = FFFFFFEC
[<c003b780>] (do_page_fault+0x0/0x228) from [<c003bb38>] (do_DataAbort+0x3c/0xa0)
[<c003bafc>] (do_DataAbort+0x0/0xa0) from [<c0034d88>] (ret_from_exception+0x0/0x10)
 r8 = 0000000E  r7 = 0000003C  r6 = 40A71358  r5 = 40A712B0
 r4 = FFFFFFFF
```

cat /proc/`pidof configer`/maps

```
~ # cat /proc/`pidof configer`/maps
00008000-00012000 r-xp 00000000 1f:01 515        /usr/sbin/configer
0001a000-0001b000 rw-p 0000a000 1f:01 515        /usr/sbin/configer
0001b000-00085000 rwxp 0001b000 00:00 0          [heap]
40000000-4001d000 r-xp 00000000 1f:01 340        /lib/ld-2.5.90.so
4001d000-40022000 rw-p 4001d000 00:00 0
40024000-40026000 rw-p 0001c000 1f:01 340        /lib/ld-2.5.90.so
40026000-4002c000 r-xp 00000000 1f:01 575        /usr/lib/libmessage.so.1.0.1.19
4002c000-40033000 ---p 00006000 1f:01 575        /usr/lib/libmessage.so.1.0.1.19
40033000-40034000 rw-p 00005000 1f:01 575        /usr/lib/libmessage.so.1.0.1.19
40034000-40050000 r-xp 00000000 1f:01 565        /usr/lib/libexpat.so.1.5.2.0
40050000-40058000 ---p 0001c000 1f:01 565        /usr/lib/libexpat.so.1.5.2.0
40058000-4005a000 rw-p 0001c000 1f:01 565        /usr/lib/libexpat.so.1.5.2.0
4005a000-4006e000 r-xp 00000000 1f:01 367        /lib/libpthread-2.5.90.so
4006e000-40075000 ---p 00014000 1f:01 367        /lib/libpthread-2.5.90.so
40075000-40076000 r--p 00013000 1f:01 367        /lib/libpthread-2.5.90.so
40076000-40077000 rw-p 00014000 1f:01 367        /lib/libpthread-2.5.90.so
40077000-40079000 rw-p 40077000 00:00 0
40079000-40190000 r-xp 00000000 1f:01 380        /lib/libc-2.5.90.so
40190000-40198000 ---p 00117000 1f:01 380        /lib/libc-2.5.90.so
40198000-40199000 r--p 00117000 1f:01 380        /lib/libc-2.5.90.so
40199000-4019b000 rw-p 00118000 1f:01 380        /lib/libc-2.5.90.so
4019b000-4019e000 rw-p 4019b000 00:00 0
401e1000-402c2000 rw-s 00000000 00:07 720896     /SYSV63010203 (deleted)
402c2000-402c3000 ---p 402c2000 00:00 0
402c3000-40ac2000 rwxp 402c3000 00:00 0
bed19000-bed2e000 rwxp bed19000 00:00 0          [stack]
```

SIGSEGV

@libc-2.5.90.so

WTF!!

不會吧(驚)

5 5
我有學過
Core dump

無敵の gdb core dump

RD1-2

```
kent@rd1-2:dielink >>> arm-linux-gdb -q 37.121-configer 37.121-core.configer.1318073168.20087.11
Don't know how to attach.  Try "help target".

warning: core file may not match specified executable file.
Error while mapping shared library sections:
/usr/lib/libmessage.so.1: No such file or directory.
Error while mapping shared library sections:
/usr/lib/libexpat.so.1: No such file or directory.
Symbol file not found for /usr/lib/libmessage.so.1
Symbol file not found for /usr/lib/libexpat.so.1
Reading symbols from /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/lib/libpthread.so.0...Reading symbols f
rom /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/usr/lib/debug/lib/libpthread-2.5.90.so.debug...done.
done.
Loaded symbols for /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/lib/libpthread.so.0
Reading symbols from /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/lib/libc.so.6...Reading symbols from /h
ome/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/usr/lib/debug/lib/libc-2.5.90.so.debug...done.
done.
Loaded symbols for /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/lib/libc.so.6
Reading symbols from /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/lib/ld-linux.so.3...Reading symbols fro
m /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/usr/lib/debug/lib/ld-2.5.90.so.debug...done.
done.
Loaded symbols for /home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/lib/ld-linux.so.3
Core was generated by `configer -i /etc/CDF.xml'.
Program terminated with signal 11, Segmentation fault.
#0  0x400daa54 in _IO_strn_overflow (fp=0x40a712b0, c=60) at vsnprintf.c:50
50      vsnprintf.c: No such file or directory.
        in vsnprintf.c
(gdb)
```

backtrace (bt)

RDI-2

```
warning: Unable to find dynamic linker breakpoint function.
GDB will be unable to debug shared library initializers
and track explicitly loaded dynamic code.
Core was generated by `configer -i /etc/CDF.xml'.
Program terminated with signal 11, Segmentation fault.
#0  0x400daa54 in ?? ()
(gdb) set solib-absolute-prefix /home/kent/Project/SD8362_VVTK/release/rootfs
warning: .dynamic section for "/home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1" is not at the expected a
ddress (wrong library or version mismatch?)
Reading symbols from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1...(no debugging symbols found)...do
ne.
Loaded symbols for /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1
Reading symbols from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libexpat.so.1...(no debugging symbols found)...done
.
Loaded symbols for /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libexpat.so.1
Reading symbols from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libpthread.so.0...(no debugging symbols found)...done.
Loaded symbols for /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libpthread.so.0
Reading symbols from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6...(no debugging symbols found)...done.
Loaded symbols for /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
Reading symbols from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/ld-linux.so.3...(no debugging symbols found)...done.
Loaded symbols for /home/kent/Project/SD8362_VVTK/release/rootfs/lib/ld-linux.so.3
(gdb) bt
#0  0x400daa54 in _IO_strn_overflow () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#1  0x400df960 in _IO_default_xsputn_internal () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#2  0x400b59f0 in vfprintf () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#3  0x400daadc in vsnprintf () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#4  0x400bdfa8 in snprintf () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#5  0x4002943c in Message_Compose_Control () from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1
#6  0x4002943c in Message_Compose_Control () from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb)
```

_IO_strn_overflow ()
vfprintf ()

C language !?

WTF!!

不會吧(驚驚)

欲窮千里目

更上一層樓

丐丐我有學過

gdb *frame UP*

frame [index] / up / down

RDI-2

```
No line number known for Message_Compose_Control.
(gdb) bt
#0  0x400daa54 in _IO_strn_overflow () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#1  0x400df960 in _IO_default_xsputn_internal () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#2  0x400b59f0 in vfprintf () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#3  0x400daadc in vsnprintf () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#4  0x400bdfa8 in snprintf () from /home/kent/Project/SD8362_VVTK/release/rootfs/lib/libc.so.6
#5  0x4002943c in Message_Compose_Control () from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1
#6  0x4002943c in Message_Compose_Control () from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb) frame 5
#5  0x4002943c in Message_Compose_Control () from /home/kent/Project/SD8362_VVTK/release/rootfs/usr/lib/libmessage.so.1
(gdb) l Message_Compose_Control
No line number known for Message_Compose_Control.
(gdb) dis
disable      disassemble  disconnect   display
(gdb) disassemble $pc
Dump of assembler code for function Message_Compose_Control:
0x40029330 <Message_Compose_Control+0>: ldrdmi  r10, [r2], -r4
0x40029334 <Message_Compose_Control+4>: ldrdmi  r10, [r2], -r12
0x40029338 <Message_Compose_Control+8>: andmi   r10, r2, r4, ror #25
0x4002933c <Message_Compose_Control+12>:        push    {r4, r5, r6, r7, r8, r9, r10, r11, lr}
0x40029340 <Message_Compose_Control+16>:        ldr     r12, [r3, #8]
0x40029344 <Message_Compose_Control+20>:        mov     r5, r3
0x40029348 <Message_Compose_Control+24>:        sub     r3, r12, #1
0x4002934c <Message_Compose_Control+28>:        sub     sp, sp, #404    ; 0x194
0x40029350 <Message_Compose_Control+32>:        cmp     r12, #5
0x40029354 <Message_Compose_Control+36>:        cmpne   r3, #1
0x40029358 <Message_Compose_Control+40>:        str     r0, [sp, #12]
0x4002935c <Message_Compose_Control+44>:        mov     r9, r1
0x40029360 <Message_Compose_Control+48>:        mov     r10, r2
0x40029364 <Message_Compose_Control+52>:        bhi     0x400295b4 <Message_Compose_Control+644>
```

WTF!!

ARM assembly

組合語言

什麼鬼呀

大學修完課後就通通還給老師了

C Code & ARM assembly

RDI-2

```c
        }
    }

    return S_OK;
}

int Message_Compose_Control(char *pcBuffer, int *piBufferLen, int iNumber, TMessageInfo *ptMessageInfo)
{
    TMessageInfo    *ptCurrMsgInfo;
    char            szTemp[MSG_CONTROL_BUF_SIZE+1], szTemp2[128];
    int             i, iLen, iOffset;

    if ((ptMessageInfo->iType == cmOutputControl) || (ptMessageInfo->iType == cmInputControl) ||
         (ptMessageInfo->iType == cmForceCI))
    {
        if (ptMessageInfo->iStream == 0)
        {
            sprintf(szTemp, "<control id=\"%d\">", ptMessageInfo->iIndex);
        }
        else
        {
            sprintf(szTemp, "<control id=\"%d\" stream=\"%d\">", ptMessageInfo->iIndex, ptMessageInfo->iStream);
        }
    }
    else
    {
        strcpy(szTemp, "<control>");
    }

    iLen = strlen(szTemp);

    for (i=0; i< iNumber; i++)
    {
        ptCurrMsgInfo = ptMessageInfo + i;

        if ((ptCurrMsgInfo->iType <= 0) || (ptCurrMsgInfo->iType > cmNumber) ||
             (ptCurrMsgInfo->aiValue[0] < 0) || (ptCurrMsgInfo->aiValue[0] > ccNumber))
        {
            return S_FAIL;
        }

        if ((ptCurrMsgInfo->iType == cmOutputControl) || (ptMessageInfo->iType == cmInputControl) ||
```

File: message.c [unix,utf-8,C] [AscII=009] [Hex=09] [Pos=940,8,70%] [LINE=1332]

1 SD8362 2 phase2 3 SD83X3 4 SD83X3 5 Phase2 6 Profilemgr 7 SIMS 8 o_o 9 EC2 10 T

RDI-2

```
(gdb) disassemble $pc
Dump of assembler code for function Message_Compose_Control:
0x40029330 <Message_Compose_Control+0>: ldrdmi  r10, [r2], -r4
0x40029334 <Message_Compose_Control+4>: ldrdmi  r10, [r2], -r12
0x40029338 <Message_Compose_Control+8>: andmi   r10, r2, r4, ror #25
0x4002933c <Message_Compose_Control+12>:        push    {r4, r5, r6, r7, r8, r9, r10, r11,
0x40029340 <Message_Compose_Control+16>:        ldr     r12, [r3, #8]
0x40029344 <Message_Compose_Control+20>:        mov     r5, r3
0x40029348 <Message_Compose_Control+24>:        sub     r3, r12, #1
0x4002934c <Message_Compose_Control+28>:        sub     sp, sp, #404    ; 0x194
0x40029350 <Message_Compose_Control+32>:        cmp     r12, #5
0x40029354 <Message_Compose_Control+36>:        cmpne   r3, #1
0x40029358 <Message_Compose_Control+40>:        str     r0, [sp, #12]
0x4002935c <Message_Compose_Control+44>:        mov     r9, r1
0x40029360 <Message_Compose_Control+48>:        mov     r10, r2
0x40029364 <Message_Compose_Control+52>:        bhi     0x400295b4 <Message_Compose_Contro
0x40029368 <Message_Compose_Control+56>:        ldr     r3, [r5, #4]
0x4002936c <Message_Compose_Control+60>:        cmp     r3, #0
0x40029370 <Message_Compose_Control+64>:        beq     0x40029514 <Message_Compose_Contro
0x40029374 <Message_Compose_Control+68>:        add     r11, sp, #16
0x40029378 <Message_Compose_Control+72>:        mov     r0, r11
0x4002937c <Message_Compose_Control+76>:        ldr     r1, [pc, #776]  ; 0x4002968c <Mess
0x40029380 <Message_Compose_Control+80>:        ldr     r2, [r5]
0x40029384 <Message_Compose_Control+84>:        bl      0x400279d4
0x40029388 <Message_Compose_Control+88>:        mov     r0, r11
0x4002938c <Message_Compose_Control+92>:        bl      0x400278cc
0x40029390 <Message_Compose_Control+96>:        cmp     r10, #0
0x40029394 <Message_Compose_Control+100>:       mov     r4, r0
0x40029398 <Message_Compose_Control+104>:       ble     0x400295d0 <Message_Compose_Contro
0x4002939c <Message_Compose_Control+108>:       ldr     r2, [r5, #8]
0x400293a0 <Message_Compose_Control+112>:       sub     r3, r2, #1
0x400293a4 <Message_Compose_Control+116>:       cmp     r3, #8
---Type <return> to continue, or q <return> to quit---bt
0x400293a8 <Message_Compose_Control+120>:       bhi     0x4002955c <Message_Compose_Contro
0x400293ac <Message_Compose_Control+124>:       ldr     r0, [r5, #12]
0x400293b0 <Message_Compose_Control+128>:       cmp     r0, #0
0x400293b4 <Message_Compose_Control+132>:       blt     0x4002955c <Message_Compose_Contro
0x400293b8 <Message_Compose_Control+136>:       cmp     r0, #6
0x400293bc <Message_Compose_Control+140>:       addle   r8, r5, #20
0x400293c0 <Message_Compose_Control+144>:       movle   r1, r5
0x400293c4 <Message_Compose_Control+148>:       movle   r7, #0
0x400293c8 <Message_Compose_Control+152>:       addle   r6, sp, #272    ; 0x110
0x400293cc <Message_Compose_Control+156>:       ble     0x400294ac <Message_Compose_Contro
0x400293d0 <Message_Compose_Control+160>:       b       0x4002955c <Message_Compose_Contro
0x400293d4 <Message_Compose_Control+164>:       cmp     r2, #3
0x400293d8 <Message_Compose_Control+168>:       beq     0x400294c0 <Message_Compose_Contro
0x400293dc <Message_Compose_Control+172>:       sub     r3, r2, #4
0x400293e0 <Message_Compose_Control+176>:       cmp     r3, #1
0x400293e4 <Message_Compose_Control+180>:       bls     0x40029594 <Message_Compose_Contro
0x400293e8 <Message_Compose_Control+184>:       cmp     r2, #6
```

看似

專業 Pro

Looks “GEEK”

In fact

實際上

發現 gcc -O3

TMD

實在太難看了

It's god damn hard to read after gcc -O3

我們需要

Source Level Debugging

Use the

Source

Loser... Orz

May The ***Source*** Be With You

How?

RTFM

Read The
Fucking Manual

load by symbol-file cmd

```
(gdb) set solib-absolute-prefix /workspace/kent/tmp/dielink/rootfsQuit
(gdb) set suQuittute-path /usr/src/debug/glibc-2.5.90/ /home/kent/MySourceForge/glibc-2.5.90/
(gdb) Quit
(gdb) l
40      ldbl_weak_alias (__snprintf, snprintf)
(gdb) f 4
#4  0x400bdfa8 in __snprintf (s=0x40a712b0 "", maxlen=60, format=0x4002ab78 "kickWatchDog") at snprintf.c:35
35        done = __vsnprintf (s, maxlen, format, arg);
(gdb) l
30      {
31        va_list arg;
32        int done;
33
34        va_start (arg, format);
35        done = __vsnprintf (s, maxlen, format, arg);
36        va_end (arg);
37
38        return done;
39      }
(gdb) set soli
solib-absolute-prefix  solib-search-path
(gdb) set solib-absolute-prefix /workspace/kent/tmp/dielink/rootfs Quit
(gdb) set substitute-path /usr/src/debug/glibc-2.5.90/ /home/kent/MySourceForge/glibc-2.5.90/ Quit
(gdb) symbol-file con
configer                          configer_klaymen_static_messagelib
(gdb) symbol-file configer
configer                          configer_klaymen_static_messagelib
(gdb) symbol-file debug/
configer.out.debug                        configer_klaymen_static_messagelib.debug
(gdb) symbol-file debug/configer.out.debug
Load new symbol table from "/workspace/kent/tmp/dielink/debug/configer.out.debug"? (y or n) y
Reading symbols from /workspace/kent/tmp/dielink/debug/configer.out.debug...done.
(gdb)
```

Re-build
debug version
shared library
with "-g"

set solib-absolute-prefix

RDI-2

```
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libc.so.6...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libc.so.6
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/ld-linux.so.3...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/ld-linux.so.3
(gdb) i sharedlibrary
From        To          Syms Read   Shared Object Library
0x400279d8  0x4002aa1c  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libmessage.so.1
0x40036198  0x4004e20c  Yes (*)     /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libexpat.so.1
0x4005dce0  0x4006c3e8  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libpthread.so.0
0x4008d990  0x40173710  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libc.so.6
0x400007e0  0x40018ab4  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/ld-linux.so.3
(*): Shared library is missing debugging information.
(gdb) set solib-absolute-prefix /home/kent/Project/DM36x/sharedlib_debug_symbol/
warning: .dynamic section for "/home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libmessage.so.1" is no
t at the expected address (wrong library or version mismatch?)
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libmessage.so.1...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libmessage.so.1
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libexpat.so.1...(no debugging sy
mbols found)...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libexpat.so.1
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libpthread.so.0...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libpthread.so.0
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libc.so.6...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libc.so.6
Reading symbols from /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/ld-linux.so.3...done.
Loaded symbols for /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/ld-linux.so.3
(gdb) i sharedlibrary
From        To          Syms Read   Shared Object Library
0x400279d8  0x4002aa1c  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libmessage.so.1
0x40036198  0x4004e20c  Yes (*)     /home/kent/Project/DM36x/sharedlib_debug_symbol/usr/lib/libexpat.so.1
0x4005dce0  0x4006c3e8  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libpthread.so.0
0x4008d990  0x40173710  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/libc.so.6
0x400007e0  0x40018ab4  Yes         /home/kent/Project/DM36x/sharedlib_debug_symbol/lib/ld-linux.so.3
(*): Shared library is missing debugging information.
(gdb)
```

Source be with You

RDI-2

```
(*): Shared library is missing debugging information.
(gdb) f 5
#5  0x4002943c in Message_Compose_Control (pcBuffer=0x6b3e2234 <Address 0x6b3e2234 out of bounds>, piBufferLe
975                                     if (snprintf(szTemp2, sizeof(szTemp2), "<source type=\"%d\">%s</sourc
 > sizeof(szTemp2))
(gdb) bt
#0  0x400daa54 in _IO_strn_overflow (fp=0x40a712b0, c=60) at vsnprintf.c:50
#1  0x400df960 in _IO_default_xsputn (f=0x40a712b0, data=<value optimized out>, n=14) at genops.c:480
#2  0x400b59f0 in _IO_vfprintf_internal (s=0x40a712b0, format=0x4002ad74 "<source type=\"%d\">%s</source>", a
p=0x40a713d4) at vfprintf.c:1309
#3  0x400daadc in _IO_vsnprintf (ing=0x40a714e8 "", maxlen=<value optimized out>, format=0x4002ad74 "<source
type=\"%d\">%s</source>", args=0x40a713d4) at vsnprintf.c:116
#4  0x400bdfa8 in   snprintf (s=0x40a712b0 "", maxlen=60, format=0x4002ab78 "kickWatchDog") at snprintf.c:35
#5  0x4002943c in Message_Compose_Control (pcBuffer=0x6b3e2234 <Address 0x6b3e2234 out of bounds>, piBufferLe
n=0x40a71cb4, iNumber=1, ptMessageInfo=0x40a71ca0) at src/message.c:975
#6  0x00009c44 in Swatchdog_WatchPolicy (eWatchPolicy=<value optimized out>) at src/configer.c:113
#7  0x0000b500 in Swatchdog_KickWatchdog (Arg=<value optimized out>) at src/configer.c:147
#8  SendSignalAndRunProcess (Arg=<value optimized out>) at src/configer.c:885
#9  0x4005e8f8 in start_thread (arg=<value optimized out>) at pthread_create.c:300
#10 0x4013a8c8 in clone () from /workspace/kent/tmp/dielink/rootfs/lib/libc.so.6
#11 0x4013a8c8 in clone () from /workspace/kent/tmp/dielink/rootfs/lib/libc.so.6
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb) f 5
#5  0x4002943c in Message_Compose_Control (pcBuffer=0x6b3e2234 <Address 0x6b3e2234 out of bounds>, piBufferLe
n=0x40a71cb4, iNumber=1, ptMessageInfo=0x40a71ca0) at src/message.c:975
975                                     if (snprintf(szTemp2, sizeof(szTemp2), "<source type=\"%d\">%s</sourc
e>", ptMessageInfo->iIndex, g_szControlMessageType[ptCurrMsgInfo->iType-1]) > sizeof(szTemp2))
(gdb) l
970                     {
971                             if ( (ptMessageInfo->aiValue[0] == 0) && (ptMessageInfo->aiValue[1] != 0) )
972                                     snprintf(szTemp2, sizeof(szTemp2), "<source name=\"%s\">%s</source>",
 (char *)ptMessageInfo->aiValue[1], g_szControlMessageType[ptCurrMsgInfo->iType-1]);
973                             else
974                             {
975                                     if (snprintf(szTemp2, sizeof(szTemp2), "<source type=\"%d\">%s</sourc
```

發現傳入
snprintf()
の資料都正確

OMFG!

電梯繼續向下

gdb frame *down*

到了

/lib/libc.so.6

-> libc-2.5.90.so

Shit!

若仿照
上面作法

難不成要自己
build **debug**
版のlibc-2.5.90

Oh No !

使用大廠の

偷偷Solution

你有權利

Say NO

MontaVista

已經幫我們

Build 好了

lib*.*.so.*.debug

```
RD1-2
libdl-2.5.90.so.debug          libnss_wins.so.debug*          libvolume_id.so.debug*
libe2p.so.2.3.debug*           libpamc.so.0.79.debug*         libwrap.so.0.7.6.debug*
libext2fs.so.2.4.debug*        libpam_misc.so.0.79.debug*     security/
libgcc_s.so.1.debug*           libpam.so.0.79.debug*          udev/
libintl.so.3.4.3.debug*        libpcprofile.so.debug
kent@rd1-2:debug_symbol >>> rea
read          readelf       read-ical     read-notepad   read-palmpix
readcd        read-expenses readlink      readonly       read-todos
kent@rd1-2:debug_symbol >>> pwd
/home/kent/Project/DM36x/debug_symbol
kent@rd1-2:debug_symbol >>> read
read          readelf       read-ical     read-notepad   read-palmpix
readcd        read-expenses readlink      readonly       read-todos
kent@rd1-2:debug_symbol >>> readlink -f .
/home/vivotek/mv_pro_5.0/montavista/pro/devkit/arm/v5t_le/target/usr/lib/debug
kent@rd1-2:debug_symbol >>> ls lib/
iptables/                      libm-2.5.90.so.debug           libpcre.so.3.12.0.debug*
ld-2.5.90.so.debug             libmemusage.so.debug           libproc-3.2.7.so.debug*
libanl-2.5.90.so.debug         libncurses.so.5.4.debug*       libpthread-2.5.90.so.debug
libblkid.so.1.0.debug*         libnsl-2.5.90.so.debug         libresolv-2.5.90.so.debug
libBrokenLocale-2.5.90.so.debug libnss_compat-2.5.90.so.debug librt-2.5.90.so.debug
libc-2.5.90.so.debug           libnss_dns-2.5.90.so.debug     libSegFault.so.debug
libcfont.so.0.0.0.debug*       libnss_files-2.5.90.so.debug   libss.so.2.0.debug*
libcidn-2.5.90.so.debug        libnss_hesiod-2.5.90.so.debug  libthread_db-1.0.so.debug
libcom_err.so.2.1.debug*       libnss_ldap-2.3.2.so.debug*    libutil-2.5.90.so.debug
libconsole.so.0.0.0.debug*     libnss_mdns-0.2.so.debug*      libuuid.so.1.2.debug*
libcrypt-2.5.90.so.debug       libnss_nis-2.5.90.so.debug     libvolume_id.so.0.75.0.debug*
libctutils.so.0.0.0.debug*     libnss_nisplus-2.5.90.so.debug libvolume_id.so.0.debug*
libdl-2.5.90.so.debug          libnss_wins.so.debug*          libvolume_id.so.debug*
libe2p.so.2.3.debug*           libpamc.so.0.79.debug*         libwrap.so.0.7.6.debug*
libext2fs.so.2.4.debug*        libpam_misc.so.0.79.debug*     security/
libgcc_s.so.1.debug*           libpam.so.0.79.debug*          udev/
libintl.so.3.4.3.debug*        libpcprofile.so.debug
kent@rd1-2:debug_symbol >>>
1 SD8362  2 phase2  3 SD83X3  4 SD83X3  5 Phase2  6 Profilemgr  7 SIMS  8 o_o  9 EC2  10 Top  11 SD73X3  12 F
```

glibc source level debug

RDI-2

```
30      {
31        va_list arg;
32        int done;
33
34        va_start (arg, format);
35        done = __vsnprintf (s, maxlen, format, arg);
36        va_end (arg);
37
38        return done;
39      }
(gdb) set solib-absolute-prefix /workspace/kent/tmp/dielink/rootfsQuit
(gdb) set suQuittute-path /usr/src/debug/glibc-2.5.90/ /home/kent/MySourceForge/glibc-2.5.90/
(gdb) Quit
(gdb) l
40      ldbl_weak_alias (__snprintf, snprintf)
(gdb) f 4
#4  0x400bdfa8 in __snprintf (s=0x40a712b0 "", maxlen=60, format=0x4002ab78 "kickWatchDog") at snprintf.c:35
35        done = __vsnprintf (s, maxlen, format, arg);
(gdb) l
30      {
31        va_list arg;
32        int done;
33
34        va_start (arg, format);
35        done = __vsnprintf (s, maxlen, format, arg);
36        va_end (arg);
37
38        return done;
39      }
(gdb) set soli
solib-absolute-prefix  solib-search-path
(gdb) set solib-absolute-prefix /workspace/kent/tmp/dielink/rootfs
(gdb) set substitute-path /usr/src/debug/glibc-2.5.90/ /home/kent/MySourceForge/glibc-2.5.90/
```

It's Showtime

Null pointer access issue

RD1-2

```
37  {
38    /* When we come to here this means the user supplied buffer is
39       filled.  But since we must return the number of characters which
40       would have been written in total we must provide a buffer for
41       further use.  We can do this by writing on and on in the overflow
42       buffer in the _IO_strnfile structure.  */
43    _IO_strnfile *snf = (_IO_strnfile *) fp;
44
45    if (fp->_IO_buf_base != snf->overflow_buf)
46      {
47        /* Terminate the string.  We know that there is room for at
48           least one more character since we initialized the stream with
49           a size to make this possible.  */
50        *fp->_IO_write_ptr = '\0';
51
52        INTUSE(_IO_setb) (fp, snf->overflow_buf,
53                          snf->overflow_buf + sizeof (snf->overflow_buf), 0);
54
55        fp->_IO_write_base = snf->overflow_buf;
56        fp->_IO_read_base = snf->overflow_buf;
57        fp->_IO_read_ptr = snf->overflow_buf;
58        fp->_IO_read_end = snf->overflow_buf + sizeof (snf->overflow_buf);
59      }
60
61     fp->_IO_write_ptr = snf->overflow_buf;
62     fp->_IO_write_end = snf->overflow_buf;
/home/kent/MySourceForge/glibc-2.5.90/libio/vsnprintf.c
Display all 385 possibilities? (y or n)
(gdb) p *fp->_IO_wri
p *fp->_IO_write_base
p *fp->_IO_write_end
p *fp->_IO_write_ptr
p *fp->_IO_write_t
(gdb) p *fp->_IO_write_ptr
Cannot access memory at address 0x0
(gdb) p fp->_IO_write_ptr
$3 = 0x0
(gdb)
$4 = 0x0
(gdb) bt
#0  0x400daa54 in _IO_strn_overflow (fp=0x40a712b0, c=60) at vsnprintf.c:50
#1  0x400df960 in _IO_default_xsputn (f=0x40a712b0, data=<value optimized out>, n=14) at genops.c:480
#2  0x400b59f0 in _IO_vfprintf_internal (s=0x40a712b0, format=0x4002ad74 "<source type=\"%d\">%s</source>", ap=0x40a713d4) at vfprintf.c:1309
#3  0x400daadc in _IO_vsnprintf (ing=0x40a714e8 "", maxlen=<value optimized out>, format=0x4002ad74 "<source type=\"%d\">%s</source>", args=0x40a713d4) at vsnprintf.c:116
#4  0x400bdfa8 in __snprintf (s=0x40a712b0 "", maxlen=60, format=0x4002ab78 "kickWatchDog") at snprintf.c:35
#5  0x4002943c in ?? ()
(gdb) f 0
#0  0x400daa54 in _IO_strn_overflow (fp=0x40a712b0, c=60) at vsnprintf.c:50
(gdb)
1 SD8362  2 phase2  3 SD83X3  4 SD83X3  5 Phase2  6 Profilemgr  7 SIMS  8 o_o  9 EC2  10 Top  11 SD73X3  [12] FB-IP8162  13 FB-8151  14 SD8362  [rd1-2 | 0.42 0.54 0.56 | 10/31 15:11
```

多虧了

神器 GDB

我們終於學會

Shared Library Debugging

某Daemon之死

至今仍是個謎(驚)

案例二、

Case 2

劫持 FDs

File Descriptors Hijacking

時間有限
下回揭曉

File Descriptor Hijacking / 劫持 FDs 之奇技淫巧

Reference

- 快快樂樂學 GNU Debugger (gdb) Part I + II (Jserv)
  - http://jserv.sayya.org/debugger/
- http://pyclewn.sourceforge.net/
- http://clewn.sourceforge.net/
- http://reverse.put.as/
- GDB的妙用 (vgod)
- [GDB Tricks] File Descriptor Hijacking / 劫持 FDs 之奇技淫巧