



# Debugging.TV

Frame 0x0A

Presenter: Dmitry Vostokov

MEMORY DUMP ANALYSIS SERVICES

[DumpAnalysis.com](http://DumpAnalysis.com)

Including Crash and Hang Analysis Audit, Training and Seminars

Sponsors

**OPENTASK**

Iterative and Incremental Publishing

# Topics

- Platform independent memory dump analysis
- Platform independent MDA patterns
- Exception Thread pattern (on Mac OS X)
- WER vs. MER and WinDbg vs. GDB

# Exception Thread

```
void bar()
{
    int *p = NULL;

    *p = 1;
}

void foo()
{
    bar();
}

int main(int argc, const char * argv[])
{
    foo();
    return 0;
}
```

# Xcode

The screenshot shows the Xcode debugger interface. On the left, the 'By Thread' view shows a paused thread named 'Thread 1' (com.apple.main-thread) with four frames: '0 bar', '1 foo', '2 main', and '3 start'. The main window displays assembly code for 'Test`bar at main.c:12:'. The current instruction is highlighted: '0x101406eb0: movl \$1, (%rax)'. A tooltip on the right indicates a crash: 'Thread 1: EXC\_BAD\_ACCESS (code=1, address=0x0)'. The bottom toolbar shows navigation controls and the current thread/frame selection. The bottom status bar displays the memory address 'p = (int \*) 0x0000000000000000' and the format '(lldb)'. The 'All Output' panel is visible on the right with a 'Clear' button and three checkboxes.

By Thread    By Queue

Test  
Paused

Thread 1  
com.apple.main-thread

- 0 bar
- 1 foo
- 2 main
- 3 start

Test`bar at main.c:12:

```
0x101406ea0: pushq %rbp
0x101406ea1: movq %rsp, %rbp
0x101406ea4: movq $0, -8(%rbp)
0x101406eac: movq -8(%rbp), %rax
0x101406eb0: movl $1, (%rax)
0x101406eb6: popq %rbp
0x101406eb7: ret
|
```

Thread 1: EXC\_BAD\_ACCESS (code=1, address=0x0)

Test > Thread 1 > 0 bar

Auto    🔍    All Output    Clear    [ ] [ ] [ ]

▶ **p** = (int \*) 0x0000000000000000    (lldb)

# Terminal and GDB

```
Dmitrys-MacBook-Air:/ DumpAnalysis$ ulimit -c unlimited
```

```
Dmitrys-MacBook-Air:/ DumpAnalysis$ ulimit -a  
core file size          (blocks, -c) unlimited  
[...]
```

```
Dmitrys-MacBook-Air:/ DumpAnalysis$  
/Users/DumpAnalysis/Library/Developer/Xcode/DerivedData/Test-  
adsepwnmyotiyfiajcuqrwjzmy/Build/Products/Debug/Test  
Segmentation fault: 11 (core dumped)
```

```
Dmitrys-MacBook-Air:/ DumpAnalysis$ ls /cores  
core.5494
```

```
Dmitrys-MacBook-Air:/ DumpAnalysis$  
/Applications/Xcode.app/Contents/Developer/usr/bin/gdb  
/Users/DumpAnalysis/Library/Developer/Xcode/DerivedData/Test-  
adsepwnmyotiyfiajcuqrwjzmy/Build/Products/Debug/Test /cores/core.5494
```

# Debugger Output

```
GNU gdb 6.3.50-20050815 (Apple version gdb-1752) (Sat Jan 28 03:02:46 UTC 2012)
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 "x86_64-apple-darwin"...
Reading symbols for shared libraries .. done

Reading symbols for shared libraries . done
Reading symbols for shared libraries ..... done
#0  0x000000010eb07eb0 in bar ()
    at /Users/DumpAnalysis/Documents/MacOSX-Debugging/Test/Test/Test/main.c:15
15      *p = 1;
(gdb) where
#0  0x000000010eb07eb0 in bar ()
    at /Users/DumpAnalysis/Documents/MacOSX-Debugging/Test/Test/Test/main.c:15
#1  0x000000010eb07ec9 in foo ()
    at /Users/DumpAnalysis/Documents/MacOSX-Debugging/Test/Test/Test/main.c:20
#2  0x000000010eb07eeb in main (argc=1,
    argv=0x7fff6e706ad0)
    at /Users/DumpAnalysis/Documents/MacOSX-Debugging/Test/Test/Test/main.c:25
Current language:  auto; currently minimal
(gdb) q
```

# **!Ad Hardcore Technical Support Training**

April 2, 2012: [Introduction to Software Narratology](#) (free)

April 11-16, 2012: [Accelerated Windows Memory Dump Analysis](#)

April 20-23, 2012: [Advanced Windows Memory Dump Analysis](#)

April 27-30, 2012: [Accelerated Software Trace Analysis](#)

[Accelerated Mac OS X Core Dump Analysis](#)

Forthcoming: [Linux Core Dump Analysis](#)

[Deep Down C++](#)

## [Training Schedule](#)

Debugging.TV

Happy St. Patrick's Day!