
目录

前言	1.1
LLDB概览	1.2
如何用LLDB调试程序	1.3
LLDB命令	1.4
命令概览	1.4.1
lldb的cheat sheet	1.4.1.1
lldb的help	1.4.1.2
常用命令	1.4.2
image	1.4.2.1
image lookup	1.4.2.1.1
举例	1.4.2.1.1.1
单命令	1.4.2.1.1.1.1
多命令对比	1.4.2.1.1.1.2
help语法	1.4.2.1.1.2
image list	1.4.2.1.2
举例	1.4.2.1.2.1
心得	1.4.2.1.2.2
help语法	1.4.2.1.2.3
image dump	1.4.2.1.3
register	1.4.2.2
expression	1.4.2.3
p	1.4.2.3.1
po	1.4.2.3.2
memory	1.4.2.4
memory read	1.4.2.4.1
help语法	1.4.2.4.1.1
disassemble	1.4.2.5
thread	1.4.2.6
frame	1.4.2.7
breakpoint	1.4.2.8
help语法	1.4.2.8.1
watchpoint	1.4.2.9
help语法	1.4.2.9.1
调试控制	1.4.2.10

run	1.4.2.10.1
continue	1.4.2.10.2
next	1.4.2.10.3
nexti	1.4.2.10.3.1
step	1.4.2.10.4
stepi	1.4.2.10.4.1
jump	1.4.2.10.5
finish	1.4.2.10.6
exit	1.4.2.10.7
LLDB心得	1.5
导出结果到文件	1.5.1
命令缩写	1.5.2
Xcode中lldb	1.5.3
支持自动补全	1.5.3.1
查看函数调用堆栈	1.5.3.2
iOS逆向	1.5.4
chisel	1.5.4.1
LLVM	1.5.5
附录	1.6
文档	1.6.1
参考资料	1.6.2

主流调试器：LLDB

- 最新版本： `v1.2.1`
- 更新时间： `20231221`

简介

介绍主流的调试器LLDB。先是LLDB概览；再详细介绍LLDB的命令，包括LLDB的命令概览和LLDB的各个命令；LLDB命令概览包括cheat sheet和help语法；LLDB常用命令包括image，以及image中的image lookup、image list、image dump等，且给出了详细的例子和help语法；然后是register、expression和其中的p和po，；然后介绍了其中的、memory和memory read及其中的help语法、disassemble、thread、frame、breakpoint、watchpoint、以及调试控制相关的命令，包括run、continue、next和nexti、step和stepi、jump、finish、exit等，且都给出help语法和用法举例；然后再整理出相关心得，包括命令的缩写、Xcode中的lld，包括自动补全和查看函数调用堆栈、iOS逆向、LLVM等等。最后给出相关的文档和资料。

源码+浏览+下载

本书的各种源码、在线浏览地址、多种格式文件下载如下：

HonKit源码

- [crifan/popular_debugger_lldb](#): 主流调试器：LLDB

如何使用此HonKit源码去生成发布为电子书

详见：[crifan/honkit_template: demo how to use crifan honkit template and demo](#)

在线浏览

- 主流调试器：LLDB [book.crifan.org](#)
- 主流调试器：LLDB [crifan.github.io](#)

离线下载阅读

- 主流调试器：LLDB PDF
- 主流调试器：LLDB ePub
- 主流调试器：LLDB Mobi

版权和用途说明

此电子书教程的全部内容，如无特别说明，均为本人原创。其中部分内容参考自网络，均已备注了出处。如发现有侵权，请通过邮箱联系我 `admin` 艾特 `crifan.com`，我会尽快删除。谢谢合作。

各种技术类教程，仅作为学习和研究使用。请勿用于任何非法用途。如有非法用途，均与本人无关。

鸣谢

感谢我的老婆陈雪的包容理解和悉心照料，才使得我 `crifan` 有更多精力去专注技术专研和整理归纳出这些电子书和技术教程，特此鸣谢。

其他

作者的其他电子书

本人 `crifan` 还写了其他 150+ 本电子书教程，感兴趣可移步至：

[crifan/crifan_ebook_readme: Crifan的电子书的使用说明](#)

关于作者

关于作者更多介绍，详见：

[关于CrifanLi李茂 – 在路上](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-12-21 22:15:37

LLDB概览

TODO:

- 【已解决】XCode和Ildb调试常见用法和调试心得
-

背景

- 主流常见 调试器 = debugger
 - GNU 的 [GDB](#)
 - (开源项目 LLVM 中的) [LLDB](#)
- iOS端
 - Apple的 Xcode 的内置调试器
 - 之前: [GDB](#)
 - 现在(Xcode 5+): [LLDB](#)
- Android端
 - Android内置的调试器
 - 之前: [GDB](#)
 - 现在: [LLDB](#)

LLDB

- [LLDB](#)
 - 名称: 常写成小写的 [lldb](#)
 - 是什么: 一个下一代的、高性能的 开源调试器
 - 说明
 - 和LLVM关系
 - 属于 (更大的, 开源的) [LLVM](#) 项目的一部分 =其中 一个模块
 - 所以LLDB也是开源的
 - 常搭配 [LLVM](#) 的其他模块一起使用
 - [expression parser](#) = 解释器 : [Clang](#)
 - [disassembler](#) = 反汇编器 : [LLVM disassembler](#)
 - 和Xcode关系
 - 是Xcode内置的调试器: 之前是GDB, 现在是LLDB
 - 特点
 - 支持调试语言
 - Xcode中的LLDB
 - 支持调试 [C](#)、[Objective-C](#)、[C++](#)
 - 支持运行平台: 桌面端 [macOS](#)、移动端 [iOS](#) (设备和模拟器)
 - 支持众多平台: [macOS](#)、[iOS](#)、[Linux](#)、[FreeBSD](#)、[NetBSD](#)、[Windows](#)

Features matrix [4]

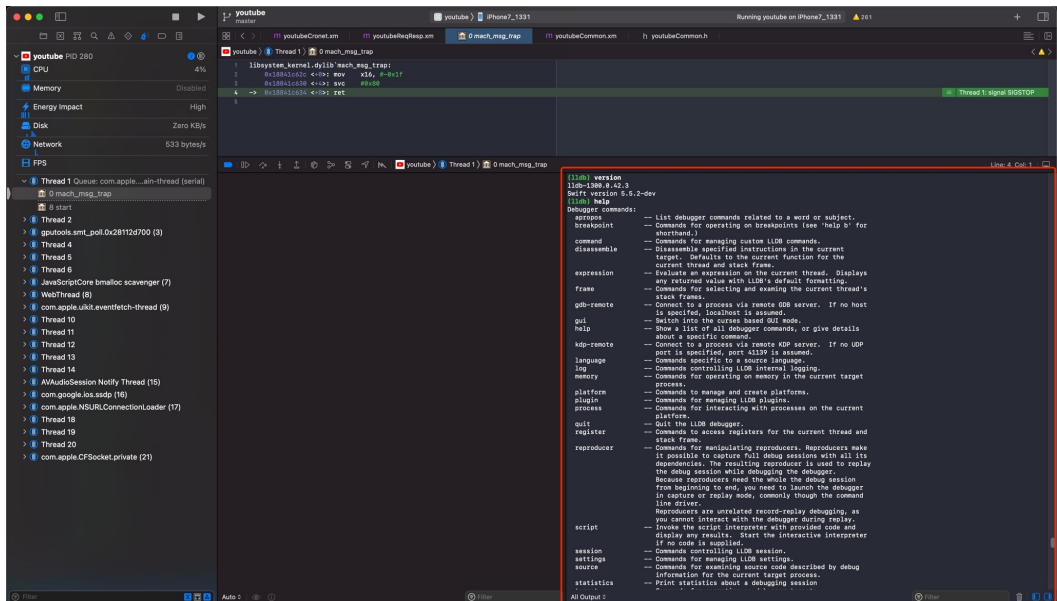
Feature	FreeBSD	Linux	macOS	Windows
Backtracing	✓	✓	✓	✓
Breakpoints	✓	✓	✓	✓
C++11:	✓	✓	✓	?
Command-line lldb tool	✓	✓	✓	✓
Core file debugging	✓	✓	✓	✓
Debugserver (remote debugging)	Not ported	Not ported	✓	Not ported
Disassembly	✓	✓	✓	✓
Expression evaluation	?	Works with some bugs	✓	Works with some bugs
JIT debugging	?	Symbolic debugging only	Untested	✗
Objective-C 2.0:	?	N/A	✓	N/A

- 支持 REPL、C++ 和 Python 插件
 - 注：REPL = Read-Eval-Print Loop = 交互式解释器
- 此处
 - 主要使用场景
 - iOS逆向时，用 LLDB 调试 objc 的相关内容

LLDB的位置和版本

Mac

- Mac中的lldb
 - 二进制
 - Mac自带的：`/usr/bin/lldb`
 - Xcode中的：`/Applications/Xcode.app/Contents/Developer/usr/bin/lldb`
 - 集成进XCode
 - 位置：内嵌在Xcode中的（一般是右下角的）调试区域的控制台



Mac自带的lldb

```
crifan@licrifandeMacBook-Pro ~$ which lldb
/usr/bin/lldb
crifan@licrifandeMacBook-Pro ~$ ll /usr/bin/lldb
-rwxr-xr-x  1 root  wheel   134K  1  1  2020 /usr/bin/lldb

crifan@licrifandeMacBook-Pro ~$ /usr/bin/lldb --version
lldb-1300.0.42.3
Swift version 5.5.2-dev
```

Xcode中的lldb

```
crifan@licrifandeMacBook-Pro ~$ ll /Applications/Xcode.app/Contents/Developer/usr/bin/lldb
-rwxr-xr-x  1 crifan  staff   828K  12 15  2021 /Applications/Xcode.app/Contents/Developer/usr/bin/lldb

crifan@licrifandeMacBook-Pro ~$ /Applications/Xcode.app/Contents/Developer/usr/bin/lldb --version
lldb-1300.0.42.3
Swift version 5.5.2-dev
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:34:42

如何用LLDB调试程序

关于如何用LLDB去调试具体的程序，详见：

- 用LLDB调试iOS程序
 - [lldb+debugserver · iOS逆向开发：动态调试](#)
- 用LLDB调试Android程序
 - [lldb调试安卓 · Android逆向：动态调试](#)

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-08-12 21:58:47

LLDB命令

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

LLDB命令概览

TODO:

【整理】lldb的语法和用法

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

lldb的cheat sheet

lldb的 cheat sheet =小抄=手册:

lldb cheat sheet

Execution Commands

```
start lld (prefix with xcrun on os x)
>lldb [program.app]
>lldb -- program.app arg1
load program
>file program.app
run program
>process launch [-- args]
>run [args]
set arguments
>settings set target.run-args 1
launch process in new terminal
>process launch --tty -- <args>
set env variables
>settings set target.env-vars DEBUG=1
remove env variables
>settings remove target.env-vars
DEBUG
show program arguments
>settings show target.run-args
set env variable and run
>process launch -v DEBUG=1
attach to process by PID
>process attach --pid 123
attach to process by name
>process attach --name a.out [--
waitfor]
attach to remote gdb on eorgadd
>gdb-remote eorgadd:8000
attach to gdb server on localhost
>gdb-remote 8000
attach to remote Darwin kernel in kdp mode
>kdp-remote eorgadd
source level single step
>thread step-in
>step
>s
source level setup over
>thread step-over
>nnext
>n
instruction level single step
>thread step-inst
```

```
set dynamic type printing as default
>settings set target.prefer-dynamic
run-target
```

```
calling a function with a breakpoint
>expr -i 0 --
function_with_a_breakpoint()
```

```
calling a function that crashes
>expr -u 0 -- function_which_crashes()
Examining Thread State
```

```
show backtrace (current thread)
>thread backtrace
>bt
```

```
show backtrace for all threads
>thread backtrace all
>bt all
```

```
backtrace the first 5 frames of current thread
>thread backtrace -c 5
>bt 5 (lldb-169 and later)
>bt -c 5 (lldb-168 and later)
```

```
select a different stack frame by index
>frame select 12
>fr s 12
>fr 12
```

```
show frame information
>frame info
select stack frame the called current frame
>up
>frame select --relative=1
```

```
select stack frame that is called by current
frame
>drown
>frame select --relative=-1
>fr s -r-1
```

```
select different frame using relative offset
>frame select --relative 2
>fr s -r2
>frame select --relative -3
>fr s -r-3
```

```
show general purpose registers
>register read
write 123 to register rax
>register write rax 123
```

```
skip 8 bytes using with program counter
>register write pc $pc+8
```

```
>si
instruction level single step over
>thread step-inst-over
>n1
step out of the currently selected frame
>thread step-out
>finish
```

```
Return from currently frame, with return
value
>thread return [RETURN_EXPRESSION]
```

```
Backtrace and disassemble every time you
stop
>target stop-hook add
>bt
>disassemble --pc
>DONE
```

```
run until line 12 or end of frame
>thread until 12
```

```
Breakpoint Commands
set breakpoint at all functions named main
>breakpoint set --name main
>br s -n main
>bt main
```

```
set breakpoint in file test.c line 12
>breakpoint set --file test.c --line
12
>br s -f test.c -l 12
>bt test.c:12
```

```
set breakpoint at all C++ methods with name
main
>breakpoint set --method main
>br s -M main
```

```
set breakpoint at ObjC function
>breakpoint set --name "[NSString
stringWithFormat:]"
>b "[NSString stringWithFormat:]
```

```
set breakpoint at all ObjC functions whose
selector is count
>breakpoint set --selector count
>br s -S count
```

```
set breakpoint by regular expression function
name
>breakpoint set --func-regex print.*
```

```
ensure that breakpoints by file and line work
(c/cpp/objc)
>settings set target.inline-
breakpoint-strategy always
```

```
show general purpose registers as signed
decimal
>register read --format i
>re r -f i
>register read/d
```

```
show all registers in all register threads
>register read --all
>re r -a
```

```
show registers rax, rsp, rbp
>register read rax rsp rbp
show register rax with binary format
>register read --format binary rax
```

```
read memory from 0xbffff3c0 and show 4 hex
uint32_t values
>memory read --size 4 --format x --
count 4 0xbffff3c0
>me r -s4 -fx -c4 0xbffff3c0
>x -s4 -fx -c4 0xbffff3c0
>memory read/4xw 0xbffff3c0
>x/4xw 0xbffff3c0
>memory read --gdb-format 4xw
0xbffff3c0
```

```
read memory starting at the expression
"argv[0]"
>memory read `argv[0]`
>memory read --size `sizeof(int)`
`argv[0]`
```

```
read 512 bytes from address 0xbffff3c0 and
save results to a local file
>memory read --outfile /tmp/mem.txt -
-count 512 0xbffff3c0
>me r -o/tmp/mem.txt -c512 0xbffff3c0
>x/512bx -o/tmp/mem.txt 0xbffff3c0
```

```
save binary memory data starting at 0x1000
and ending at 0x2000 to file
>memory read --outfile /tmp/mem.bin -
-binary 0x1000 0x2000
>me r -o /tmp/mem.bin -b 0x1000
0x2000
```

```
get information about specific heap
allocation (Mac OS X only)
>command script import
lldb.macosx.heap
>process launch --environment
MallocStackLogging=1 -- [ARGS]
>malloc_info --stack-history
0x10010d680
```

```
>br s -f foo.c -l 12
set a breakpoint by regular expression on
source file contents
>breakpoint set --source-pattern
regular-expression --file Sourcefile
>br s -p regular-expression -f file
```

```
set conditional breakpoint
>breakpoint set --name foo --
condition '(int)strcmp(y,"hello") ==
0'
>br s -n foo -c
(int)strcmp(y,"hello") == 0'
```

```
list breakpoints
>breakpoint list
>br 1
```

```
delete a breakpoint
>breakpoint delete 1
>br del 1
```

```
Watchpoint Commands
set watchpoint on variable when written to
>watchpoint set variable global_var
>wa s v global_var
```

```
set watchpoint on memory of pointer size
>watchpoint set expression --
0x123456
>wa s e -- 0x123456
```

```
set watchpoint on memory of custom size
>watchpoint set expression -x
byte_size -- 0x123456
>wa s e -x byte_size -- 0x123456
```

```
set a condition on a watchpoint
>watch set var global
>watchpoint modify -c '(global==5)'
```

```
list watchpoints
>watchpoint list
>watch 1
```

```
delete a watchpoint
>watchpoint delete 1
>watch del 1
```

```
Examining Variables
show arguments and local variables
>frame variable
>fr v
```

```
show local variables
>frame variable --no-args
>fr v -a
```

```
get information about specific heap
allocation and cast result to dynamic type
that can be deduced (Mac OS X only)
>command script import
lldb.macosx.heap
>malloc_info --type 0x10010d680
```

```
find all heap blocks that contain pointer
specified by an expression EXPR (Mac OS X
only)
>command script import
lldb.macosx.heap
>ptr_refs EXPR
```

```
find all heap blocks that contain a C string
anywhere in the block (Mac OS X only)
>command script import
lldb.macosx.heap
>cstr_refs CSTRING
```

```
disassemble current function for current
frame
>disassemble -frame
>di -f
```

```
disassemble any functions named main
>disassemble --name main
>di -n main
```

```
disassemble address range
>disassemble --start-address 0x1eb8 -
-end-address 0x1ec3
>di -s 0x1eb8 -e 0x1ec3
```

```
disassemble 20 instructions from start
address
>disassemble --start-address 0x1eb8 -
-count 20
>di -s 0x1eb8 -c 20
```

```
show mixed source and disassembly by the
current function
>disassemble --frame -mixed
>di -f -m
```

```
disassemble the current function for the
current frame and show the opcode bytes
>disassemble --frame -bytes
>di -f -b
```

```
disassemble the current source line for the
current frame
>disassemble --line
>di -l
```

```
show contents of variable bar
>frame variable bar
>fr v bar
>p bar
```

```
show contents of var bar formatted as hex
>frame variable --format x bar
>fr v -f x bar
```

```
show contents of global variable baz
>target variable baz
>ta v baz
```

```
show global/static variables in current file
>target variable
>ta v
```

```
show argc and argv every time you stop
>target stop-hook add --one-liner
"frame variable argc argv"
>ta st a -o "fr v argc argv"
>display argc
>display argv
```

```
display argc and argv when stopping in main
>target stop-hook add --name main --
one-liner "frame variable argc argv"
>ta st a -n main -o "fr v argc argv"
```

```
display "this" when in class MyClass
>target stop-hook add --classname
MyClass --one-liner "frame variable
*this"
>ta st a -c MyClass -o "fr v *this"
```

```
Evaluating Expressions
evaluate expression (print alias possible as
well)
>expr (int) printf ("Print nine:
%d.", 4 + 5)
>print (int) printf ("Print nine:
%d.", 4 + 5)
```

```
using a convenience variable
>expr unsigned int $foo = 5
```

```
print the ObjC description of an object
>expr -o -- [SomeClass
returnAnObject]
>po [SomeClass returnAnObject]
```

```
print dynamic type of expression result
>expr -d 1 -- [SomeClass
returnAnObject]
>expr -d 1 --
someCppObjectPtrOrReference
```

Executable and Shared Library Query Commands

```
list the main executable and all dependent
shared libraries
>image list
```

```
look up information for a raw address in the
executable or any shared libraries
>image lookup --address 0x1ec4
>im loo -a 0x1ec4
```

```
look up functions matching a regular
expression in a binary
>image lookup -r -n <FUNC_REGEX>
(debug symbols)
>image lookup -r -s <FUNC_REGEX>
(non-debug syms)
```

```
find full source line information
>image lookup -v --address 0x1ec4
(Look for entryLine)
```

```
look up information for an address in a.out
only
>image lookup --address 0x1ec4 a.out
>im loo -a 0x1ec4 a.out
```

```
look up information for a type Pointer by
name
>image lookup --type Point
>im loo -t Point
```

```
dump all sections from the main executable
and any shared libraries
>image dump sections
```

```
dump all sections in the a.out module
>image dump sections a.out
```

```
dump all symbols from the main executable
and any shared libraries
>image dump symtab
```

```
dump all symbols in a.out and liba.so
>image dump symtab a.out liba.so
```

```
Miscellaneous
echo text to the screen
>script print "Here is some text"
```

```
remap source file pathnames for the debug
session (e.g. if program was built on another
PC)
>settings set target.source-map
/buildbot/path /my/path
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

lldb的帮助

lldb的帮助: 显示所有命令的帮助信息

```
(lldb) help
Debugger commands:
  apropos          -- List debugger commands related to a word or subject.
  breakpoint       -- Commands for operating on breakpoints (see 'help b' for
                    shorthand.)
  command         -- Commands for managing custom LLDB commands.
  disassemble     -- Disassemble specified instructions in the current
                    target. Defaults to the current function for the
                    current thread and stack frame.
  expression      -- Evaluate an expression on the current thread. Displays
                    any returned value with LLDB's default formatting.
  frame           -- Commands for selecting and examining the current thread's
                    stack frames.
  gdb-remote      -- Connect to a process via remote GDB server. If no host
                    is specified, localhost is assumed.
  gui             -- Switch into the curses based GUI mode.
  help           -- Show a list of all debugger commands, or give details
                    about a specific command.
  kdp-remote      -- Connect to a process via remote KDP server. If no UDP
                    port is specified, port 41139 is assumed.
  language        -- Commands specific to a source language.
  log             -- Commands controlling LLDB internal logging.
  memory          -- Commands for operating on memory in the current target
                    process.
  platform        -- Commands to manage and create platforms.
  plugin          -- Commands for managing LLDB plugins.
  process         -- Commands for interacting with processes on the current
                    platform.
  quit            -- Quit the LLDB debugger.
  register        -- Commands to access registers for the current thread and
                    stack frame.
  reproducer      -- Commands for manipulating reproducers. Reproducers make
                    it possible to capture full debug sessions with all its
                    dependencies. The resulting reproducer is used to replay
                    the debug session while debugging the debugger.
                    Because reproducers need the whole the debug session
                    from beginning to end, you need to launch the debugger
                    in capture or replay mode, commonly though the command
                    line driver.
                    Reproducers are unrelated record-replay debugging, as
                    you cannot interact with the debugger during replay.
  script          -- Invoke the script interpreter with provided code and
                    display any results. Start the interactive interpreter
                    if no code is supplied.
  session         -- Commands controlling LLDB session.
  settings        -- Commands for managing LLDB settings.
  source          -- Commands for examining source code described by debug
                    information for the current target process.
  statistics      -- Print statistics about a debugging session
```

```

target      -- Commands for operating on debugger targets.
thread      -- Commands for operating on one or more threads in the
              current process.
trace       -- Commands for loading and using processor trace
              information.
type        -- Commands for operating on the type system.
version     -- Show the LLDB debugger version.
watchpoint  -- Commands for operating on watchpoints.
Current command abbreviations (type 'help command alias' for more info):
add-dsym    -- Add a debug symbol file to one of the target's current modules
              by specifying a path to a debug symbols file or by using the
              options to specify a module.
attach      -- Attach to process by ID or name.
b           -- Set a breakpoint using one of several shorthand formats.
bt          -- Show the current thread's call stack. Any numeric argument
              displays at most that many frames. The argument 'all' displays
              all threads. Use 'settings set frame-format' to customize the
              printing of individual frames and 'settings set thread-format'
              to customize the thread header.
c           -- Continue execution of all threads in the current process.
call        -- Evaluate an expression on the current thread. Displays any
              returned value with LLDB's default formatting.
continue    -- Continue execution of all threads in the current process.
detach      -- Detach from the current target process.
di          -- Disassemble specified instructions in the current target.
              Defaults to the current function for the current thread and
              stack frame.
dis         -- Disassemble specified instructions in the current target.
              Defaults to the current function for the current thread and
              stack frame.
display     -- Evaluate an expression at every stop (see 'help target
              stop-hook'.)
down        -- Select a newer stack frame. Defaults to moving one frame, a
              numeric argument can specify an arbitrary number.
env         -- Shorthand for viewing and setting environment variables.
exit        -- Quit the LLDB debugger.
f           -- Select the current stack frame by index from within the current
              thread (see 'thread backtrace'.)
file        -- Create a target using the argument as the main executable.
finish      -- Finish executing the current stack frame and stop after
              returning. Defaults to current thread unless specified.
history     -- Dump the history of commands in this session.
              Commands in the history list can be run again using "!<INDEX>".
              "!-<OFFSET>" will re-run the command that is <OFFSET> commands
              from the end of the list (counting the current command).
image      -- Commands for accessing information for one or more target
              modules.
j           -- Set the program counter to a new address.
jump        -- Set the program counter to a new address.
kill        -- Terminate the current target process.
l           -- List relevant source code using one of several shorthand formats.
list        -- List relevant source code using one of several shorthand formats.
n           -- Source level single step, stepping over calls. Defaults to
              current thread unless specified.
next        -- Source level single step, stepping over calls. Defaults to
              current thread unless specified.

```

```

nexti    -- Instruction level single step, stepping over calls. Defaults to
          current thread unless specified.
ni       -- Instruction level single step, stepping over calls. Defaults to
          current thread unless specified.
p        -- Evaluate an expression on the current thread. Displays any
          returned value with LLDB's default formatting.
parray   -- parray <COUNT> <EXPRESSION> -- lldb will evaluate EXPRESSION to
          get a typed-pointer-to-an-array in memory, and will display
          COUNT elements of that type from the array.
po       -- Evaluate an expression on the current thread. Displays any
          returned value with formatting controlled by the type's author.
poarray  -- poarray <COUNT> <EXPRESSION> -- lldb will evaluate EXPRESSION to
          get the address of an array of COUNT objects in memory, and will
          call po on them.
print    -- Evaluate an expression on the current thread. Displays any
          returned value with LLDB's default formatting.
q        -- Quit the LLDB debugger.
r        -- Launch the executable in the debugger.
rbreak   -- Sets a breakpoint or set of breakpoints in the executable.
re       -- Commands to access registers for the current thread and stack
          frame.
repl     -- Evaluate an expression on the current thread. Displays any
          returned value with LLDB's default formatting.
run      -- Launch the executable in the debugger.
s        -- Source level single step, stepping into calls. Defaults to
          current thread unless specified.
shell    -- Run a shell command on the host.
si       -- Instruction level single step, stepping into calls. Defaults to
          current thread unless specified.
sif      -- Step through the current block, stopping if you step directly
          into a function whose name matches the TargetFunctionName.
step     -- Source level single step, stepping into calls. Defaults to
          current thread unless specified.
stepi    -- Instruction level single step, stepping into calls. Defaults to
          current thread unless specified.
t        -- Change the currently selected thread.
tbreak   -- Set a one-shot breakpoint using one of several shorthand formats.
undisplay -- Stop displaying expression at every stop (specified by stop-hook
          index.)
up       -- Select an older stack frame. Defaults to moving one frame, a
          numeric argument can specify an arbitrary number.
v        -- Show variables for the current stack frame. Defaults to all
          arguments and local variables in scope. Names of argument,
          local, file static and file global variables can be specified.
          Children of aggregate variables can be specified such as
          'var- child.x'. The -> and [] operators in 'frame variable' do
          not invoke operator overloads if they exist, but directly access
          the specified element. If you want to trigger operator
          overloads use the expression command to print the variable
          instead.
          It is worth noting that except for overloaded operators, when
          printing local variables 'expr local_var' and 'frame var
          local_var' produce the same results. However, 'frame variable'
          is more efficient, since it uses debug information and memory
          reads directly, rather than parsing and evaluating an
          expression, which may even involve JITing and running code in

```

```

the target program.
var      -- Show variables for the current stack frame. Defaults to all
          arguments and local variables in scope. Names of argument,
          local, file static and file global variables can be specified.
          Children of aggregate variables can be specified such as
          'var- child.x'. The -> and [] operators in 'frame variable' do
          not invoke operator overloads if they exist, but directly access
          the specified element. If you want to trigger operator
          overloads use the expression command to print the variable
          instead.
          It is worth noting that except for overloaded operators, when
          printing local variables 'expr local_var' and 'frame var
          local_var' produce the same results. However, 'frame variable'
          is more efficient, since it uses debug information and memory
          reads directly, rather than parsing and evaluating an
          expression, which may even involve JITing and running code in
          the target program.
vo       -- Show variables for the current stack frame. Defaults to all
          arguments and local variables in scope. Names of argument,
          local, file static and file global variables can be specified.
          Children of aggregate variables can be specified such as
          'var- child.x'. The -> and [] operators in 'frame variable' do
          not invoke operator overloads if they exist, but directly access
          the specified element. If you want to trigger operator
          overloads use the expression command to print the variable
          instead.
          It is worth noting that except for overloaded operators, when
          printing local variables 'expr local_var' and 'frame var
          local_var' produce the same results. However, 'frame variable'
          is more efficient, since it uses debug information and memory
          reads directly, rather than parsing and evaluating an
          expression, which may even involve JITing and running code in
          the target program.
x        -- Read from the memory of the current target process.
For more information on any command, type 'help command-name'.

```

lldb的help的用法解释

单个命令=子命令

单个命令的语法，可以用：

```
help <command name>
```

举例：

- `help register`

```

(lldb) help register
Commands to access registers for the current thread and stack frame.

Syntax: register [read write] ...

```


The following subcommands are supported:

```
read -- Dump the contents of one or more register values from the
      current frame. If no register is specified, dumps them all.
write -- Modify a single register value.
```

For more help on any particular subcommand, type 'help <command> <subcommand>'.
└─

- help memory

```
(lldb) help memory
```

Commands for operating on memory in the current target process.

Syntax: memory <subcommand> [<subcommand-options>]

The following subcommands are supported:

```
find -- Find a value in the memory of the current target process.
history -- Print recorded stack traces for allocation/deallocation events
         associated with an address.
read -- Read from the memory of the current target process.
region -- Get information on the memory region containing an address in
         the current target process.
write -- Write to the memory of the current target process.
```

For more help on any particular subcommand, type 'help <command> <subcommand>'.
└─

单个命令的子命令=单个命令的参数

而命令的子命令的语法，也是前面加上help:

```
help <command> <subcommand>
```

举例:

- help memory read

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:23:26

常用命令

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

image

TODO:

- 【已解决】lldb命令使用心得: image

- image核心子命令
 - [image lookup](#)
 - [image list](#)
 - [image dump](#)

image命令语法

```
(lldb) help image
Commands for accessing information for one or more target modules.

Syntax: image

'image' is an abbreviation for 'target modules'
(lldb) help target modules
Commands for accessing information for one or more target modules.

Syntax: target modules <sub-command> ...

The following subcommands are supported:

    add          -- Add a new module to the current target's modules.
    dump        -- Commands for dumping information about one or more target
                  modules.
    list         -- List current executable and dependent shared library
                  images.
    load         -- Set the load addresses for one or more sections in a
                  target module.
    lookup       -- Look up information within executable and dependent
                  shared library images.
    search-paths -- Commands for managing module search paths for a target.
    show-unwind  -- Show synthesized unwind instructions for a function.

For more help on any particular subcommand, type 'help command <subcommand>'.

```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:03:28

image lookup

- image lookup 的典型写法举例

- -a

```
image lookup -a 0x114fb69c4

image lookup -v -a 0x112d57730
image lookup -va 0x0000000195d31f14
im loo -va 0xa916260182319ea0

image lookup -a 0x00000000009e8000+0x0000001008f83b8
```

- -n

```
image lookup -n statfs
image lookup -n ___lldb_unnamed_symbol12575$$akd
image lookup -n "+[UMUserManager sharedManager]"

image lookup -name getInt

image lookup -r -n objc_msgSend
image lookup -r -n accountsWithAccountType
image lookup -r -n "accountsWithAccountType:"
image lookup -rn initWithURL
image lookup -rn "AADeviceInfo"
image lookup -rn "setValue:forHTTPHeaderField:"
image lookup -rn "AADeviceInfo udid"
im loo -rn SetRequestPriority

image lookup -vn ___lldb_unnamed_symbol12575$$akd
image lookup -vn "-[AALoginAccountRequest urlRequest]"
image lookup -vn "-[AKAppleIDAuthenticationContextManager shouldContinueWithAut
henticationResults:error:forContextID:completion:]"
image lookup -vn "+[AADeviceInfo(Deprecated) udid]"
image lookup -vn "+[AADeviceInfo udid]"
image lookup -vn "-[AADeviceInfo udid]"

image lookup -vrn "AADeviceInfo udid"
```

- -s

```
image lookup -s statfs

image lookup -r -s objc_msgSend
image lookup -r -s statfs
image lookup -r -s handlePressGesture
image lookup -r -s _nextButtonSelected
image lookup -r -s "authenticationContext"
```

```
image lookup -vs SecTrustEvaluateFastAsync
image lookup -vs "AADeviceInfo"
image lookup -vs "__lldb_unnamed_sybo1972"

image lookup -v -s SSLHandshake
```

o -t

```
image lookup -type Person
image lookup -type NSObject
```

- image lookup 的主要参数

- o 根据不同类型去搜

- address=地址

- -a <address-expression> = --address <address-expression>

- 作用: Lookup an address in one or more target modules

- name=名称

- -n <function-or-symbol> = --name <function-or-symbol>

- 作用: Lookup a function or symbol by **name** in one or more target modules

- searches **debug** symbols

- symbol=符号

- -s <symbol> = --symbol <symbol>

- 作用: Lookup a symbol by name in the **symbol** tables in one or more target modules

- searches **non-debug** symbols

- type=类型

- -t <name> = --type <name>

- 作用: Lookup a type by name in the debug symbols in one or more target modules

- o 其他辅助参数

- 输出详细信息

- -v = --verbose

- 作用: Enable verbose lookup information

- 以正则表达式的方式去搜

- -r = --regex

- 作用: The <name> argument for name lookups are regular expressions.

以及:

- lldb支持命令的缩写

- o 常见的缩写

- image -> img 、 im

- lookup -> loo

- lldb支持字母合在一起

- o -v -a -> -va

- o -r -n -> -rn

所以就有多种常见写法:

- 基本参数的写法
 - `image lookup -a == im loo -a`
 - `image lookup -n == im loo -n`
 - `image lookup -s == im loo -s`
- 如果加上 `-v` , 则是:
 - `image lookup -v -a == im loo -v -a`
 - `== image lookup -va == im loo -va`
 - `image lookup -v -n == im loo -v -n`
 - `== image lookup -vn == im loo -vn`
 - `image lookup -v -s == im loo -v -s`
 - `== image lookup -vs == im loo -vs`
- 如果加上 `-r` , 则是:
 - `image lookup -r -n == im loo -r -n`
 - `== image lookup -rn == im loo -rn`
 - `image lookup -r -s == im loo -r -s`
 - `== image lookup -rs == im loo -rs`

image lookup的高级用法

限定在某个二进制Module内部查找

lldb 中 `image lookup` 的基本用法:

```
image lookup -a 0x1000
```

如果要限定在某个 二进制文件 = 库 = 模块 = Module 中的话, 语法是:

```
image lookup -a 0x1000 a.out
```

- 其中 `a.out` 是二进制文件名

举例

AppleAccount

```
im loo -rn didReceiveData AppleAccount
```

- 参数说明
 - `AppleAccount` 文件 (此处调试时的路径是)
 - `/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount`
- 完整输出

```
(lldb) im loo -rn didReceiveData AppleAccount
3 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
    Address: AppleAccount[0x0000000192f4cf34] (AppleAccount.__TEXT.__text + 1000)
)
    Summary: AppleAccount`-[AAURLSessionDelegate URLSession:dataTask:didReceiveData:]
    Address: AppleAccount[0x0000000192f4cff4] (AppleAccount.__TEXT.__text + 1192)
    Summary: AppleAccount`-[AAURLSession URLSession:dataTask:didReceiveData:]
    Address: AppleAccount[0x0000000192f83d88] (AppleAccount.__TEXT.__text + 225852)
)
    Summary: AppleAccount`-[AARquester connection:didReceiveData:]
```

AppleStoreCore

```
image lookup -rn initialize AppleStoreCore
```

- 完整输出

```
(lldb) image lookup -rn initialize AppleStoreCore
3 matches found in /Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-fbcdzphrbokcgxhejxslslydrdyaa/Build/Products/Debug-iphon eos/Jolly.app/Frameworks/AppleStoreCore.framework/AppleStoreCore:
    Address: AppleStoreCore[0x000000000049a6bc] (AppleStoreCore.__TEXT.__text + 4807496)
    Summary: AppleStoreCore`initializeAvailabilityCheck
    Address: AppleStoreCore[0x000000000049a6c4] (AppleStoreCore.__TEXT.__text + 4807504)
    Summary: AppleStoreCore`_initializeAvailabilityCheck
    Address: AppleStoreCore[0x00000000000a8010] (AppleStoreCore.__TEXT.__text + 668828)
    Summary: AppleStoreCore`static AppleStoreCore.User.initialize() -> ()
(lldb) image lookup -rs initialize AppleStoreCore
4 symbols match the regular expression 'initialize' in /Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-fbcdzphrbokcgxhejxslslydrdyaa/Build/Products/Debug-iphon eos/Jolly.app/Frameworks/AppleStoreCore.framework/AppleStoreCore:
    Address: AppleStoreCore[0x000000000049a6bc] (AppleStoreCore.__TEXT.__text + 4807496)
    Summary: AppleStoreCore`initializeAvailabilityCheck
    Address: AppleStoreCore[0x000000000049a6c4] (AppleStoreCore.__TEXT.__text + 4807504)
    Summary: AppleStoreCore`_initializeAvailabilityCheck
    Address: AppleStoreCore[0x00000000004d0430] (AppleStoreCore.__TEXT.__stubs + 11844)
    Summary: AppleStoreCore`symbol stub for: swift_deallocUninitializedObject
    Address: AppleStoreCore[0x00000000000a8010] (AppleStoreCore.__TEXT.__text + 668828)
    Summary: AppleStoreCore`static AppleStoreCore.User.initialize() -> ()
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:37:47

image lookup的举例

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:43:52

image lookup的单命令举例

```
image lookup -a
```

lldb官网示例

```
(lldb) image lookup --address 0x100000aa3
Address: a.out[0x0000000100000aa3] (a.out.__TEXT.__text + 131)
Summary: a.out`main + 67 at main.c:13

(lldb) image lookup -a 0x1000
Address: a.out[0x00000000000001000] (a.out.__PAGEZERO + 4096)

Address: libsystem_c.dylib[0x00000000000001000] (libsystem_c.dylib.__TEXT.__text +
928)
Summary: libsystem_c.dylib`mcount + 9

Address: libsystem_dnssd.dylib[0x00000000000001000] (libsystem_dnssd.dylib.__TEXT.
__text + 456)
Summary: libsystem_dnssd.dylib`ConvertHeaderBytes + 38

Address: libsystem_kernel.dylib[0x00000000000001000] (libsystem_kernel.dylib.__TEX
T.__text + 1116)
Summary: libsystem_kernel.dylib`clock_get_time + 102

(lldb) image lookup -a 0x1000 a.out
Address: a.out[0x00000000000001000] (a.out.__PAGEZERO + 4096)

(lldb) image lookup --address 0x100123aa3
Address: a.out[0x0000000100000aa3] (a.out.__TEXT.__text + 131)
Summary: a.out`main + 67 at main.c:13
```

AwemeCore相关

```
(lldb) image lookup -a 0x114fb69c4
Address: AwemeCore[0x000000001009a9c4] (AwemeCore.__BD_TEXT.__text + 175598020)
Summary: AwemeCore`___lldb_unnamed_symbol1674948$$AwemeCore

(lldb) image lookup -a 0x10aed1514
Address: AwemeCore[0x0000000005fb5514] (AwemeCore.__BD_TEXT.__text + 6886676)
Summary: AwemeCore`___lldb_unnamed_symbol149294$$AwemeCore

(lldb) image lookup -a 0x10ad33da0
Address: AwemeCore[0x0000000005e17da0] (AwemeCore.__BD_TEXT.__text + 5193120)
Summary: AwemeCore`___lldb_unnamed_symbol136558$$AwemeCore

(lldb) image lookup -a 0x10a7125c4
Address: AwemeCore[0x0000000005a3a5c4] (AwemeCore.__BD_TEXT.__text + 1140164)
Summary: AwemeCore`___lldb_unnamed_symbol16381$$AwemeCore

(lldb) image lookup -a 0x11427c178
```

```
Address: AwemeCore[0x00000000fa14178] (AwemeCore.__BD_TEXT.__text + 168755576)
Summary: AwemeCore`___lldb_unnamed_symbol1588524$$AwemeCore

(lldb) image lookup -a 0x1128fc41c
Address: AwemeCore[0x00000000fa1441c] (AwemeCore.__BD_TEXT.__text + 168756252)
Summary: AwemeCore`___lldb_unnamed_symbol1588526$$AwemeCore
```

Apple Store相关

AppleAccountUI

```
(lldb) image lookup -a 0x1af6922fc
Address: AppleAccountUI[0x00000001ae0222fc] (AppleAccountUI.__TEXT.__text + 475404)
)
Summary: AppleAccountUI`-[AAUISignInViewController _attemptAuthenticationWithCont
ext:].cold.1
```

StoreKitUI

```
(lldb) image lookup -a 0x00000001b0b30578
Address: StoreKitUI[0x00000001ae9f0578] (StoreKitUI.__TEXT.__objc_methlist + 2432)

Summary: StoreKitUI`_OBJC_$_CLASS_METHODS_SKUIImageCollectionViewCell + 16
```

YouTube相关

YouTube

```
(lldb) image lookup -a 0x0000000105163494
Address: YouTube[0x00000001003db494] (YouTube.__TEXT.__text + 4013856)
Summary: YouTube`___lldb_unnamed_symbol122084$$YouTube + 164
```

Module_Framework

```
(lldb) image lookup -a 0x00000001063d9850
Address: Module_Framework[0x000000000194d850] (Module_Framework.__TEXT.__text + 2
6515536)
Summary: Module_Framework`___lldb_unnamed_symbol15681$$Module_Framework + 376
(lldb) image lookup -a 0x00000001063d8d34
Address: Module_Framework[0x000000000194cd34] (Module_Framework.__TEXT.__text + 2
6512692)
Summary: Module_Framework`___lldb_unnamed_symbol15676$$Module_Framework + 80
(lldb) image lookup -a 0x000000010888211c
Address: Module_Framework[0x0000000003df611c] (Module_Framework.__TEXT.__text + 6
4954652)
Summary: Module_Framework`___lldb_unnamed_symbol170908$$Module_Framework + 52
(lldb) image lookup -a 0x0000000108884064
Address: Module_Framework[0x0000000003df8064] (Module_Framework.__TEXT.__text + 6
```

```

4962660)
  Summary: Module_Framework`___lldb_unnamed_symbol1171000$$Module_Framework + 28
(1ldb) image lookup -a 0x0000000108388e4c
  Address: Module_Framework[0x00000000038fce4c] (Module_Framework.__TEXT.__text + 5
9739724)
  Summary: Module_Framework`___lldb_unnamed_symbol110205$$Module_Framework + 40
(1ldb) image lookup -a 0x00000001063d999c
  Address: Module_Framework[0x000000000194d99c] (Module_Framework.__TEXT.__text + 2
6515868)
  Summary: Module_Framework`-[HAMCronetDataLoadTask startWithDelegate:delegateQueue:
]_block_block + 36

(1ldb) image lookup -a 0x1062d4298
  Address: Module_Framework[0x0000000003e84298] (Module_Framework.__TEXT.__stubs +
29856)
  Summary: Module_Framework`symbol stub for: objc_msgSend

(1ldb) image lookup -a 0x103a97ed4
  Address: Module_Framework[0x0000000000f8fed4] (Module_Framework.__TEXT.__text + 1
6301780)
  Summary: Module_Framework`-[MLHAMQueuePlayerSegmentList updatePeriodCurrentTimeFo
rSegment:]_block

(1ldb) image lookup -a 0x108806a08
  Address: Module_Framework[0x000000000198aa08] (Module_Framework.__TEXT.__text + 2
6765832)
  Summary: Module_Framework`-[HAMPlayerInternal pause]

(1ldb) im loo -a 0x00000001091694a4
  Address: Module_Framework[0x0000000003df94a4] (Module_Framework.__TEXT.__text + 6
4967844)
  Summary: Module_Framework`___lldb_unnamed_symbol1171165$$Module_Framework

```

-a 找不到的例子

另外，通过 `-a` 也有找不到的

比如：

进入：跳板汇编代码后

```

-> 0x112925950: adrp    x16, -70002
    0x112925954: ldr     x16, [x16, #0xb30]
    0x112925958: br     x16
    0x11292595c: adrp    x16, -70002
    0x112925960: ldr     x16, [x16, #0xb38]
    0x112925964: br     x16
    0x112925968: adrp    x16, -70002
    0x11292596c: ldr     x16, [x16, #0xb40]

```

去查找： `0x112925950`

```
(1ldb) image lookup -a 0x112925950
```

```
Address: AwemeCore[0x000000001147d950] (AwemeCore.__BD_TEXT.__stubs + 28440)
Summary:
```

就是：找不到的。

其他找不到的例子：

```
(lldb) image lookup -a 0x114af2380
Address: AwemeCore[0x000000001147a380] (AwemeCore.__BD_TEXT.__stubs + 14664)
Summary:
```

-a 找出是 data 的例子

```
(lldb) image lookup -a 0x00000001f2468c6a
Address: AppleAccountUI[0x00000001f0328c6a] (AppleAccountUI.__DATA_DIRTY.__objc_data + 242)
Summary: (void *)0x73e800000001f246
```

先 `_shortMethodDescription` 再用 `-a` 查找函数地址，可以找到Class函数

对于 `_shortMethodDescription` 查看到的函数地址：

```
+ (id) awe_userRecommendImageNamed:(id) arg1; (0x114d06fcc)
+ (id) awe_userRecommendImageNamed:(id) arg1 compatibleWithTraitCollection:(id) arg2; (0x114d06fdc)
```

用 `-a` 去查找对应地址，可以找到，对应的 `Class` 级别的函数：

```
(lldb) image lookup -a 0x114d06fcc
Address: AwemeCore[0x00000000ffffefcc] (AwemeCore.__BD_TEXT.__text + 174960588)
Summary: AwemeCore`+[UIImage(AWEUserRecommend) awe_userRecommendImageNamed:]
(lldb) image lookup -a 0x114d06fdc
Address: AwemeCore[0x00000000ffffefdc] (AwemeCore.__BD_TEXT.__text + 174960604)
Summary: AwemeCore`+[UIImage(AWEUserRecommend) awe_userRecommendImageNamed:compatibleWithTraitCollection:]
```

其他

```
(lldb) image lookup -a 0x1021a4ab0
Address: libsubstrate.dylib[0x0000000000014ab0] (libsubstrate.dylib.__TEXT.__stubs + 516)
Summary: libsubstrate.dylib`symbol stub for: _dyld_get_all_image_infos

(lldb) image lookup -a 0x0000000105c636fc
Address: substitute-inserter.dylib[0x0000000000076fc] (substitute-inserter.dylib.__TEXT.__text + 11384)
Summary: substitute-inserter.dylib`___lldb_unnamed_symbol6$$substitute-inserter.d
```

```
ylib + 3536
```

```
image lookup -n == image lookup -name
```

getInt

```
(lldb) image lookup -name getInt
1 match found in /Users/liu_david/Library/Developer/Xcode/DerivedData/TestWeak-egnzdbnd
wsikvcheqmcxvkqnwbw/Build/Products/Debug/TestWeak:
    Address: TestWeak[0x0000000100001a70] (TestWeak.__TEXT.__text + 48)
    Summary: TestWeak`-[Person getInt] at main.m:29
1 match found in /usr/lib/libicucore.A.dylib:
    Address: libicucore.A.dylib[0x000000000003b046] (libicucore.A.dylib.__TEXT.__text + 239078)
    Summary: libicucore.A.dylib`icu::ResourceBundle::getInt(UErrorCode) const
1 match found in /System/Library/Frameworks/Security.framework/Versions/A/Security:
    Address: Security[0x00000000012260e] (Security.__TEXT.__text + 1184206)
    Summary: Security`Security::Context::getInt(unsigned int, int) const
1 match found in /System/Library/Frameworks/SceneKit.framework/Versions/A/SceneKit:
    Address: SceneKit[0x00000000024f396] (SceneKit.__TEXT.__text + 2414070)
    Summary: SceneKit`getInt(std::__1::basic_string char, std::__1::char_traits char
>, std::__1::allocator char>> const, std::__1::basic_string char, std::__1::char_traits char,
std::__1::allocator char>>, int, bool, bool)
1 match found in /usr/lib/libTelephonyUtilDynamic.dylib:
    Address: libTelephonyUtilDynamic.dylib[0x000000000012e10] (libTelephonyUtilDynamic.dylib.__TEXT.__text + 69904)
    Summary: libTelephonyUtilDynamic.dylib`ctu::cf::map_adapter::getInt(__CFString
const*, int) const
1 match found in /System/Library/PrivateFrameworks/CorePrediction.framework/Versions/A/
CorePrediction:
    Address: CorePrediction[0x0000000000475c4] (CorePrediction.__TEXT.__text + 286
980)
    Summary: CorePrediction`-[CPMLEvaluationResult getInt]
```

transformOtherModelToSuit:

```
(lldb) image lookup -n transformOtherModelToSuit:
1 match found in /Users/crifan/Library/Developer/Xcode/DerivedData/DiDi-cwpbvvyvqmeijmc
jnnethzuthsy/Build/Products/Debug-iphonesimulator/DiDi.app/DiDi:
    Address: DiDi[0x0000000100293d60] (DiDi.__TEXT.__text + 2693664)
    Summary: DiDi`+[FW_BetFunction transformOtherModelToSuit:] at FW_BetFunction.m:
107
```

__lldb_unnamed_symbol2575\$\$akd

```
(lldb) image lookup -n __lldb_unnamed_symbol2575$$akd
1 match found in /System/Library/PrivateFrameworks/AuthKit.framework/akd:
    Address: akd[0x00000001000a0460] (akd.__TEXT.__text + 639120)
    Summary: akd`__lldb_unnamed_symbol2575$$akd
```

"+[UMUserManager sharedManager]"

```
(lldb) image lookup -n "+[UMUserManager sharedManager]"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/UserManagement.framework/UserManagement:
    Address: UserManagement[0x000000019ffdb3e4] (UserManagement.__TEXT.__text + 1896
)
    Summary: UserManagement`+[UMUserManager sharedManager]
```

`image lookup -r -n == image lookup -rn == im loo -rn`

objc_msgSend

```
(lldb) image lookup -r -n objc_msgSend
7 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.7 (17H35)
/Symbols/usr/lib/libobjc.A.dylib:
    Address: libobjc.A.dylib[0x00000001801b7e60] (libobjc.A.dylib.__TEXT.__text + 2
816)
    Summary: libobjc.A.dylib`objc_msgSend_uncached      Address: libobjc.A.dylib[
0x00000001801b7b20] (libobjc.A.dylib.__TEXT.__text + 1984)
    Summary: libobjc.A.dylib`objc_msgSend      Address: libobjc.A.dylib[0x0000000
1801b7ce0] (libobjc.A.dylib.__TEXT.__text + 2432)
    Summary: libobjc.A.dylib`objc_msgSendSuper      Address: libobjc.A.dylib[0x00
000001801b7d60] (libobjc.A.dylib.__TEXT.__text + 2560)
    Summary: libobjc.A.dylib`objc_msgSendSuper2      Address: libobjc.A.dylib[0x0
000001801b8080] (libobjc.A.dylib.__TEXT.__text + 3360)
    Summary: libobjc.A.dylib`objc_msgSendSuper2_debug      Address: libobjc.A.dyl
ib[0x00000001801b8060] (libobjc.A.dylib.__TEXT.__text + 3328)
    Summary: libobjc.A.dylib`objc_msgSend_debug      Address: libobjc.A.dylib[0x0
000001801b8040] (libobjc.A.dylib.__TEXT.__text + 3296)
    Summary: libobjc.A.dylib`objc_msgSend_noarg
2 matches found in /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjoxbki
bsdwekuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/libcycrypt.dylib:
    Address: libcycrypt.dylib[0x000000000069cb8] (libcycrypt.dylib.__TEXT.__text +
416952)
    Summary: libcycrypt.dylib`$objc_msgSend(OpaqueJSContext const*, OpaqueJSValue*,
OpaqueJSValue*, unsigned long, OpaqueJSValue const* const*, OpaqueJSValue const**)
    Address: libcycrypt.dylib[0x00000000006d318] (libcycrypt.dylib.__TEXT.__text + 430
872)
    Summary: libcycrypt.dylib`$objc_msgSend(OpaqueJSContext const*, OpaqueJSValue*,
OpaqueJSValue*, unsigned long, OpaqueJSValue const* const*)
```

accountsWithAccountType

```
(lldb) image lookup -r -n accountsWithAccountType
```

```

44 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/Frameworks/Accounts.framework/Accounts:
    Address: Accounts[0x0000000181af486c] (Accounts.__TEXT.__text + 26736)
    Summary: Accounts`-[ACAccountStore accountsWithAccountTypeIdentifiers:preloaded
Properties:error:] Address: Accounts[0x0000000181af5fbc] (Accounts.__TEXT.__text
+ 32704)
    Summary: Accounts`__79-[ACAccountStore accountsWithAccountTypeIdentifiers:prelo
adedProperties:error:]_block_invoke Address: Accounts[0x0000000181af6e08] (Accou
nts.__TEXT.__text + 36364)
    Summary: Accounts`__79-[ACAccountStore accountsWithAccountTypeIdentifiers:prelo
adedProperties:error:]_block_invoke_2 Address: Accounts[0x0000000181af9328] (Acc
ounts.__TEXT.__text + 45868)
    Summary: Accounts`__56-[ACAccountStore accountsWithAccountType:options:error:]_
block_invoke_2 Address: Accounts[0x0000000181afb2cc] (Accounts.__TEXT.__text + 5
3968)
    Summary: Accounts`-[ACAccountStore accountsWithAccountType:] Address: Ac
counts[0x0000000181afb944] (Accounts.__TEXT.__text + 55
...

```

找到了太多的结果。

"accountsWithAccountType:"

所以故意用 "accountsWithAccountType:" 缩小范围:

```

(lldb) image lookup -r -n "accountsWithAccountType:"
27 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/Frameworks/Accounts.framework/Accounts:
    Address: Accounts[0x0000000181af9328] (Accounts.__TEXT.__text + 45868)
    Summary: Accounts`__56-[ACAccountStore accountsWithAccountType:options:error:]_
block_invoke_2 Address: Accounts[0x0000000181afb2cc] (Accounts.__TEXT.__text + 5
3968)
    Summary: Accounts`-[ACAccountStore accountsWithAccountType:] Address: Ac
counts[0x0000000181afc3bc] (Accounts.__TEXT.__text + 58304)
    Summary: Accounts`-[ACAccountStore accountsWithAccountType:options:completion:]
Address: Accounts[0x0000000181afcb30] (Accounts.__TEXT.__text + 60212)
    Summary: Accounts`__42-[ACAccountStore accountsWithAccountType:]_block_invoke
Address: Accounts[0x0000000181afcbb0] (Accounts.__TEXT.__text + 60340)
    Summary: Accounts`-[ACAccountStore accountsWithAccountType:options:error:]
Address: Accounts[0x0000000181afd1bc] (Accounts.__TEXT.__text + 61888)
    Summary: Accounts`__56-[ACAccountStore accountsWithAccountType:options:error:]_
block_invoke Address: Accounts[0x0000000181afd474] (Accounts.__TEXT.__text + 625
84)
    Summary: Accounts`__61-[ACAccountStore accountsWithAccountType:options:completi
on:]_block_invoke_2 Address: Accounts[0x0000000181b001ac] (Accounts.__TEXT.__tex
t + 74160)
    Summary: Accounts`__61-[ACAccountStore accountsWithAccountType:options:completi
on:]_block_invoke Address: Accounts[0x0000000181b00a28] (Accounts.__TEXT.__text
+ 76332)
    Summary: Accounts`__42-[ACAccountStore accountsWithAccountType:]_block_invoke_2
Address: Accounts[0x0000000181b064f0] (Accounts.__TEXT.__text + 99572)
    Summary: Accounts`__42-[ACAccountStore accountsWithAccountType:]_block_invoke.1
70 Address: Accounts[0x0000000181b0653c] (Accounts.__TEXT.__text + 99648)

```



```

Summary: Accounts`__56-[ACAccountStore accountsWithAccountType:options:error:]_
block_invoke.179      Address: Accounts[0x0000000181b06598] (Accounts.__TEXT.__text +
99740)
Summary: Accounts`-[ACAccountStore accountsWithAccountType:completion:]
Address: Accounts[0x0000000181b068bc] (Accounts.__TEXT.__text + 100544)
Summary: Accounts`__53-[ACAccountStore accountsWithAccountType:completion:]_blo
ck_invoke      Address: Accounts[0x0000000181b069b8] (Accounts.__TEXT.__text + 100796)

Summary: Accounts`__53-[ACAccountStore accountsWithAccountType:completion:]_blo
ck_invoke_2      Address: Accounts[0x0000000181b06b70] (Accounts.__TEXT.__text + 1012
36)
Summary: Accounts`__53-[ACAccountStore accountsWithAccountType:completion:]_blo
ck_invoke.182      Address: Accounts[0x0000000181b06c10] (Accounts.__TEXT.__text + 10
1396)
Summary: Accounts`__53-[ACAccountStore accountsWithAccountType:completion:]_blo
ck_invoke_2.183      Address: Accounts[0x0000000181b06da4] (Accounts.__TEXT.__text +
101800)
Summary: Accounts`__53-[ACAccountStore accountsWithAccountType:completion:]_blo
ck_invoke.186      Address: Accounts[0x0000000181b06e08] (Accounts.__TEXT.__text + 10
1900)
Summary: Accounts`__61-[ACAccountStore accountsWithAccountType:options:completi
on:]_block_invoke.187      Address: Accounts[0x0000000181b06eb4] (Accounts.__TEXT.__t
ext + 102072)
Summary: Accounts`__61-[ACAccountStore accountsWithAccountType:options:completi
on:]_block_invoke_2.188      Address: Accounts[0x0000000181b07048] (Accounts.__TEXT._
_text + 102476)
Summary: Accounts`__61-[ACAccountStore accountsWithAccountType:options:completi
on:]_block_invoke.191      Address: Accounts[0x0000000181b4b90c] (Accounts.__TEXT.__t
ext + 383248)
Summary: Accounts`-[ACAccountStore accountsWithAccountType:].cold.1      Addr
ess: Accounts[0x0000000181b4b968] (Accounts.__TEXT.__text + 383340)
Summary: Accounts`__42-[ACAccountStore accountsWithAccountType:]_block_invoke_2
.cold.1      Address: Accounts[0x0000000181b4b9c4] (Accounts.__TEXT.__text + 383432)
Summary: Accounts`__42-[ACAccountStore accountsWithAccountType:]_block_invoke.1
70.cold.1      Address: Accounts[0x0000000181b4b9f8] (Accounts.__TEXT.__text + 383484)

Summary: Accounts`__56-[ACAccountStore accountsWithAccountType:options:error:]_
block_invoke_2.cold.1      Address: Accounts[0x0000000181b4ba60] (Accounts.__TEXT.__t
ext + 383588)
Summary: Accounts`-[ACAccountStore accountsWithAccountType:completion:].cold.1
Address: Accounts[0x0000000181b4babc] (Accounts.__TEXT.__text + 383680)
Summary: Accounts`__53-[ACAccountStore accountsWithAccountType:completion:]_blo
ck_invoke_2.183.cold.1      Address: Accounts[0x0000000181b4bb18] (Accounts.__TEXT._
text + 383772)
Summary: Accounts`__61-[ACAccountStore accountsWithAccountType:options:completi
on:]_block_invoke_2.188.cold.1
4 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/PrivateFrameworks/ContactsFoundation.framework/ContactsF
oundation:
Address: ContactsFoundation[0x000000018cd42fb4] (ContactsFoundation.__TEXT.__te
xt + 455316)
Summary: ContactsFoundation`-[__CNACAccountStoreBasedProvider accountsWithAccoun
tType:]      Address: ContactsFoundation[0x000000018cd43704] (ContactsFoundation.__TE
XT.__text + 457188)
Summary: ContactsFoundation`-[__CNACAccountStaticProvider accountsWithAccountTyp
e:]      Address: ContactsFoundation[0x000000018cd43830] (ContactsFoundation.__TEXT._

```

```

_text + 457488)
    Summary: ContactsFoundation`__54-[CNACAccountStaticProvider accountsWithAccount
tType:]_block_invoke      Address: ContactsFoundation[0x000000018cd43e08] (ContactsFo
undation.__TEXT.__text + 458984)
    Summary: ContactsFoundation`-[CNACAccountProvider accountsWithAccountType:]
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
    Address: AppleAccount[0x0000000192fa0604] (AppleAccount.__TEXT.__text + 342712)
    Summary: AppleAccount`-[ACAccountStore(AppleID) accountsWithAccountType:appleID:
]
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/AppleMediaServices.framework/AppleMediaS
ervices:
    Address: AppleMediaServices[0x000000018452d3ac] (AppleMediaServices.__TEXT.__te
xt + 67172)
    Summary: AppleMediaServices`-[ACAccountStore(AppleMediaServices_Project) _ams_a
ccountsWithAccountType:options:error:]
2 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/PrivateFrameworks/CalendarFoundation.framework/CalendarF
oundation:
    Address: CalendarFoundation[0x0000000199fc36d4] (CalendarFoundation.__TEXT.__te
xt + 44404)
    Summary: CalendarFoundation`+[CalAccountsProvider _accountsWithAccountType:inSt
ore:error:]      Address: CalendarFoundation[0x0000000199fc3954] (CalendarFoundation.
__TEXT.__text + 45044)
    Summary: CalendarFoundation`__62+[CalAccountsProvider _accountsWithAccountType:
inStore:error:]_block_invoke
13 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/PrivateFrameworks/AccountsDaemon.framework/AccountsDaemo
n:
    Address: AccountsDaemon[0x00000001a57a1f4c] (AccountsDaemon.__TEXT.__text + 172
68)
    Summary: AccountsDaemon`-[ACDAccountStoreFilter accountsWithAccountType:options
:completion:]      Address: AccountsDaemon[0x00000001a57a287c] (AccountsDaemon.__TEXT
.__text + 19620)
    Summary: AccountsDaemon`__68-[ACDAccountStoreFilter accountsWithAccountType:opt
ions:completion:]_block_invoke      Address: AccountsDaemon[0x00000001a57a3e88] (Acco
untsDaemon.__TEXT.__text + 25264)
    Summary: AccountsDaemon`-[ACDAccountStore accountsWithAccountType:options:compl
etion:]      Address: AccountsDaemon[0x00000001a57a48a0] (AccountsDaemon.__TEXT.__tex
t + 27848)
    Summary: AccountsDaemon`-[ACDAccountStoreFilter accountsWithAccountType:handler:
]      Address: AccountsDaemon[0x00000001a57a5504] (AccountsDaemon.__TEXT.__text + 31
020)
    Summary: AccountsDaemon`__58-[ACDAccountStore _accountsWithAccountType:options:
error:]_block_invoke      Address: AccountsDaemon[0x00000001a57a5b18] (AccountsDaemon
.__TEXT.__text + 32576)
    Summary: AccountsDaemon`-[ACDAccountStore _accountsWithAccountType:options:erro
r:]      Address: AccountsDaemon[0x00000001a57a7988] (AccountsDaemon.__TEXT.__text +
40368)
    Summary: AccountsDaemon`__57-[ACDAccountStoreFilter accountsWithAccountType:han
dler:]_block_invoke      Address: AccountsDaemon[0x00000001a57a8294] (AccountsDaemon.
__TEXT.__text + 42684)
    Summary: AccountsDaemon`-[ACDAccountStore accountsWithAccountType:handler:]
    Address: AccountsDaemon[0x00000001a5824660] (AccountsDaemon.__TEXT.__text + 551560)
    Summary: AccountsDaemon`-[ACDAccountStore accountsWithAccountType:options:compl

```

```

etion:]._cold.1      Address: AccountsDaemon[0x00000001a58246c4] (AccountsDaemon.__TEXT.__text + 551660)
    Summary: AccountsDaemon`-[ACDAccountStore accountsWithAccountType:options:completion:]._cold.2      Address: AccountsDaemon[0x00000001a5824728] (AccountsDaemon.__TEXT.__text + 551760)
    Summary: AccountsDaemon`__58-[ACDAccountStore _accountsWithAccountType:options:
error:]._block_invoke.cold.1      Address: AccountsDaemon[0x00000001a5829e98] (AccountsDaemon.__TEXT.__text + 574144)
    Summary: AccountsDaemon`-[ACDAccountStoreFilter accountsWithAccountType:handler:]._cold.1      Address: AccountsDaemon[0x00000001a5829f5c] (AccountsDaemon.__TEXT.__text + 574340)
    Summary: AccountsDaemon`-[ACDAccountStoreFilter accountsWithAccountType:options:completion:]._cold.1

```

"accountTypeWithIdentifier:handler:"

```

(lldb) image lookup -r -n "accountTypeWithIdentifier:handler:"
4 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/AccountsDaemon.framework/AccountsDaemon:
    Address: AccountsDaemon[0x00000001a57a009c] (AccountsDaemon.__TEXT.__text + 9412)
)
    Summary: AccountsDaemon`-[ACDAccountStoreFilter accountTypeWithIdentifier:handler:]._cold.1      Address: AccountsDaemon[0x00000001a57a33a0] (AccountsDaemon.__TEXT.__text + 22472)
    Summary: AccountsDaemon`-[ACDAccountStore accountTypeWithIdentifier:handler:]._cold.1      Address: AccountsDaemon[0x00000001a57a7604] (AccountsDaemon.__TEXT.__text + 39468)
)
    Summary: AccountsDaemon`__53-[ACDAccountStore accountTypeWithIdentifier:handler:]._block_invoke      Address: AccountsDaemon[0x00000001a5824208] (AccountsDaemon.__TEXT.__text + 550448)
    Summary: AccountsDaemon`__53-[ACDAccountStore accountTypeWithIdentifier:handler:]._block_invoke.cold.1

```

initWithURL

```

(lldb) image lookup -rn initWithURL
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Accounts.framework/Accounts:
    Address: Accounts[0x0000000181b406b0] (Accounts.__TEXT.__text + 337588)
    Summary: Accounts`-[ACProtobufURL(Helpers) initWithURL:]._cold.1
16 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Foundation.framework/Foundation:
    Address: Foundation[0x0000000181bb3af0] (Foundation.__TEXT.__text + 295120)
    Summary: Foundation`-[NSConcreteFileHandle initWithURL:flags:createMode:error:]._cold.1
    Address:
...

```

SetRequestPriority

```
(lldb) im loo -rn SetRequestPriority
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/Frameworks/CFNetwork.framework/CFNetwork:
    Address: CFNetwork[0x0000000180b87c34] (CFNetwork.__TEXT.__text + 150228)
    Summary: CFNetwork`CFURLRequestSetRequestPriority
```

didReceiveData

```
(lldb) im loo -rn didReceiveData
...
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/ProtocolBuffer.framework/ProtocolBuffer:
    Address: ProtocolBuffer[0x000000019a32da74] (ProtocolBuffer.__TEXT.__text + 792
88)
    Summary: ProtocolBuffer`-[PBSessionRequester URLSession:dataTask:didReceiveData:
]
2 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/usr/lib/libTelephonyUtilDynamic.dylib:
...
2 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/Frameworks/Network.framework/Network:
    Address: Network[0x0000000198ecb61c] (Network.__TEXT.__text + 1297828)
    Summary: Network`__90-[NWURLSessionDelegateWrapper dataTask:didReceiveData:comp
lete:metrics:completionHandler:]_block_invoke      Address: Network[0x0000000198ecb744
] (Network.__TEXT.__text + 1298124)
    Summary: Network`__90-[NWURLSessionDelegateWrapper dataTask:didReceiveData:comp
lete:metrics:completionHandler:]_block_invoke_2
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/UIKitCore.framework/UIKitCore:
...
    Address: AppleMediaServices[0x00000001845330e4] (AppleMediaServices.__TEXT.__te
xt + 91036)
    Summary: AppleMediaServices`-[AMURLSession URLSession:dataTask:didReceiveData:]
    Address: AppleMediaServices[0x000000018491c054] (AppleMediaServices.__TEXT.__te
xt + 4190988)
    Summary: AppleMediaServices`-[AMCURLSessionDelegate URLSession:dataTask:didRec
eiveData:]
3 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit:
    Address: AuthKit[0x0000000192ea6fd8] (AuthKit.__TEXT.__text + 99748)
    Summary: AuthKit`-[AKURLSession URLSession:dataTask:didReceiveData:]      Add
ress: AuthKit[0x0000000192f0d750] (AuthKit.__TEXT.__text + 519452)
    Summary: AuthKit`-[AKURLSession URLSession:dataTask:didReceiveData:].cold.1
    Address: AuthKit[0x0000000192f0d784] (AuthKit.__TEXT.__text + 519504)
    Summary: AuthKit`-[AKURLSession URLSession:dataTask:didReceiveData:].cold.2
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/AppleIDAuthSupport.framework/AppleIDAuth
Support:
    Address: AppleIDAuthSupport[0x00000001b675a090] (AppleIDAuthSupport.__TEXT.__te
xt + 6184)
    Summary: AppleIDAuthSupport`-[AIASession URLSession:dataTask:didReceiveData:]
```

```

3 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
  Address: AppleAccount[0x00000000192f4cf34] (AppleAccount.__TEXT.__text + 1000)
  Summary: AppleAccount`-[__AAURLSessionDelegate URLSession:dataTask:didReceiveData:]
  Address: AppleAccount[0x00000000192f4cff4] (AppleAccount.__TEXT.__text + 1192)

  Summary: AppleAccount`-[AAURLSession URLSession:dataTask:didReceiveData:]
  Address: AppleAccount[0x00000000192f83d88] (AppleAccount.__TEXT.__text + 225852)
  Summary: AppleAccount`-[AARequester connection:didReceiveData:]
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/Frameworks/LinkPresentation.framework/LinkPresentation:
...
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/GameCenterUI.framework/GameCenterUI:
  Address: GameCenterUI[0x0000000019b06c5d4] (GameCenterUI.__TEXT.__text + 1058120)

  Summary: GameCenterUI`-[GKMatchmakerViewController match:didReceiveData:fromRem
otePlayer:]

```

"_proxiedAppBundleID"

```

(lldb) image lookup -r -n "_proxiedAppBundleID"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) a
rm64e/Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit:
  Address: AuthKit[0x00000000192effbd8] (AuthKit.__TEXT.__text + 463268)
  Summary: AuthKit`-[AKAppleIDAuthenticationContext _proxiedAppBundleID]

```

"setValue:forHTTPHeaderField:"

```

(lldb) image lookup -rn "setValue:forHTTPHeaderField:"
9 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)
/Symbols/System/Library/PrivateFrameworks/StoreServices.framework/StoreServices:
  Address: StoreServices[0x000000001921ae43c] (StoreServices.__TEXT.__text + 139100)
)
  Summary: StoreServices`-[SSURLBagContext setValue:forHTTPHeaderField:]
  Address: StoreServices[0x00000000192214c3c] (StoreServices.__TEXT.__text + 558940)
  Summary: StoreServices`-[SSMutableAuthenticationContext setValue:forHTTPHeaderF
ield:]
  Address: StoreServices[0x0000000019224fe9c] (StoreServices.__TEXT.__text +
801212)
  Summary: StoreServices`-[SSMutableURLRequestProperties setValue:forHTTPHeaderFi
eld:]
  Address: StoreServices[0x0000000019224ff54] (StoreServices.__TEXT.__text +
801396)
  Summary: StoreServices`__61-[SSMutableURLRequestProperties setValue:forHTTPHead
erField:]_block_invoke
  Address: StoreServices[0x0000000019226d5e0] (StoreServices
.__TEXT.__text + 921856)
  Summary: StoreServices`-[SSVPlaybackLeaseRequest setValue:forHTTPHeaderField:]
  Address: StoreServices[0x000000001922e1f04] (StoreServices.__TEXT.__text + 1399332)
)
  Summary: StoreServices`-[SSVAccountLessPlaybackOperation setValue:forHTTPHeader
Field:]
  Address: StoreServices[0x000000001922e1fc8] (StoreServices.__TEXT.__text
+ 1399528)

```

```

Summary: StoreServices`__63-[SSVAccountLessPlaybackOperation setValue:forHTTPHeaderField:]_block_invoke
Address: StoreServices[0x00000001922f7034] (StoreServices.__TEXT.__text + 1485652)
Summary: StoreServices`-[SSVPlatformRequestOperation setValue:forHTTPHeaderField:]
Address: StoreServices[0x00000001922f70f8] (StoreServices.__TEXT.__text + 1485848)
Summary: StoreServices`__59-[SSVPlatformRequestOperation setValue:forHTTPHeaderField:]_block_invoke
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/iTunesCloud.framework/iTunesCloud:
Address: iTunesCloud[0x0000000196363a5c] (iTunesCloud.__TEXT.__text + 760432)
Summary: iTunesCloud`-[ICMediaAssetDownloadRequest setValue:forHTTPHeaderField:]
2 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/iTunesStoreUI.framework/iTunesStoreUI:
Address: iTunesStoreUI[0x00000001aeb1b414] (iTunesStoreUI.__TEXT.__text + 388832)
)
Summary: iTunesStoreUI`-[SUScriptXMLHTTPStoreRequest setValue:forHTTPHeaderField:]
Address: iTunesStoreUI[0x00000001aeb53644] (iTunesStoreUI.__TEXT.__text + 618768)
Summary: iTunesStoreUI`-[SUScriptXMLHTTPRequest setValue:forHTTPHeaderField:]
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/Osprey.framework/Osprey:
Address: Osprey[0x00000001bb1f2e2c] (Osprey.__TEXT.__text + 67092)
Summary: Osprey`-[OspreyMutableRequest setValue:forHTTPHeaderField:]

```

"libobjc"

```

(lldb) image lookup -rn "libobjc"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/Symbolication.framework/Symbolication:
Address: Symbolication[0x000000019825ff50] (Symbolication.__TEXT.__text + 358032)
)
Summary: Symbolication`-[VMUObjectIdentifier libobjcSymbolOwner]

```

"AADeviceInfo udid"

```

(lldb) image lookup -rn "AADeviceInfo udid"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
Address: AppleAccount[0x0000000191e0d970] (AppleAccount.__TEXT.__text + 173384)
Summary: AppleAccount`-[AADeviceInfo udid]

```

"AADeviceInfo"

```

(lldb) image lookup -rn "AADeviceInfo"
46 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
Address: AppleAccount[0x0000000191de77d4] (AppleAccount.__TEXT.__text + 17324)

```

```
Summary: AppleAccount`+[AADeviceInfo isInternalBuild] Address: AppleAccount[0x0000000191e0d764] (AppleAccount.__TEXT.__text + 172860)
Summary: AppleAccount`+[AADeviceInfo currentInfo] Address: AppleAccount[0x0000000191e0d7a4] (AppleAccount.__TEXT.__text + 172924)
Summary: AppleAccount`__27+[AADeviceInfo currentInfo]_block_invoke Address: AppleAccount[0x0000000191e0d7d0] (AppleAccount.__TEXT.__text + 172968)
Summary: AppleAccount`+[AADeviceInfo locationServicesRestricted] Address: AppleAccount[0x0000000191e0d830] (AppleAccount.__TEXT.__text + 173064)
Summary: AppleAccount`-[AADeviceInfo deviceInfoDictionary] Address: AppleAccount[0x0000000191e0d950] (AppleAccount.__TEXT.__text + 173352)
Summary: AppleAccount`-[AADeviceInfo osVersion] Address: AppleAccount[0x0000000191e0d970] (AppleAccount.__TEXT.__text + 173384)
Summary: AppleAccount`-[AADeviceInfo uuid] Address: AppleAccount[0x0000000191e0d990] (AppleAccount.__TEXT.__text + 173416)
Summary: AppleAccount`-[AADeviceInfo serialNumber] Address: AppleAccount[0x0000000191e0d9b0] (AppleAccount.__TEXT.__text + 173448)
Summary: AppleAccount`-[AADeviceInfo wifiMacAddress] Address: AppleAccount[0x0000000191e0d9d0] (AppleAccount.__TEXT.__text + 173480)
Summary: AppleAccount`-[AADeviceInfo bluetoothMacAddress] Address: AppleAccount[0x0000000191e0d9f0] (AppleAccount.__TEXT.__text + 173512)
Summary: AppleAccount`-[AADeviceInfo productVersion] Address: AppleAccount[0x0000000191e0da10] (AppleAccount.__TEXT.__text + 173544)
Summary: AppleAccount`-[AADeviceInfo productType] Address: AppleAccount[0x0000000191e0da30] (AppleAccount.__TEXT.__text + 173576)
Summary: AppleAccount`-[AADeviceInfo deviceName] Address: AppleAccount[0x0000000191e0da50] (AppleAccount.__TEXT.__text + 173608)
Summary: AppleAccount`-[AADeviceInfo deviceColor] Address: AppleAccount[0x0000000191e0da70] (AppleAccount.__TEXT.__text + 173640)
Summary: AppleAccount`-[AADeviceInfo deviceEnclosureColor] Address: AppleAccount[0x0000000191e0da90] (AppleAccount.__TEXT.__text + 173672)
Summary: AppleAccount`-[AADeviceInfo deviceCoverGlassColor] Address: AppleAccount[0x0000000191e0dab0] (AppleAccount.__TEXT.__text + 173704)
Summary: AppleAccount`-[AADeviceInfo deviceHousingColor] Address: AppleAccount[0x0000000191e0dad0] (AppleAccount.__TEXT.__text + 173736)
Summary: AppleAccount`-[AADeviceInfo deviceBackingColor] Address: AppleAccount[0x0000000191e0daf0] (AppleAccount.__TEXT.__text + 173768)
Summary: AppleAccount`-[AADeviceInfo hasCellularCapability] Address: AppleAccount[0x0000000191e0db24] (AppleAccount.__TEXT.__text + 173820)
Summary: AppleAccount`-[AADeviceInfo mobileEquipmentIdentifier] Address: AppleAccount[0x0000000191e0db44] (AppleAccount.__TEXT.__text + 173852)
Summary: AppleAccount`-[AADeviceInfo internationalMobileEquipmentIdentity] Address: AppleAccount[0x0000000191e0db64] (AppleAccount.__TEXT.__text + 173884)
Summary: AppleAccount`-[AADeviceInfo storageCapacity] Address: AppleAccount[0x0000000191e0dbc0] (AppleAccount.__TEXT.__text + 173976)
Summary: AppleAccount`-[AADeviceInfo osName] Address: AppleAccount[0x0000000191e0dc30] (AppleAccount.__TEXT.__text + 174088)
Summary: AppleAccount`-[AADeviceInfo buildVersion] Address: AppleAccount[0x0000000191e0dca0] (AppleAccount.__TEXT.__text + 174200)
Summary: AppleAccount`-[AADeviceInfo regionCode] Address: AppleAccount[0x0000000191e0dca4] (AppleAccount.__TEXT.__text + 174204)
Summary: AppleAccount`-[AADeviceInfo apnsToken] Address: AppleAccount[0x0000000191e0dd08] (AppleAccount.__TEXT.__text + 174304)
Summary: AppleAccount`-[AADeviceInfo deviceClass] Address: AppleAccount[0x0000000191e0dd28] (AppleAccount.__TEXT.__text + 174336)
Summary: AppleAccount`-[AADeviceInfo modelName] Address: AppleAccount[0x0000000191e0dd48] (AppleAccount.__TEXT.__text + 174368)
```

```

Summary: AppleAccount`-[AADeviceInfo chipIdentifier] Address: AppleAccount[0x000000191e0dd68] (AppleAccount.__TEXT.__text + 174400)
Summary: AppleAccount`-[AADeviceInfo uniqueChipIdentifier] Address: AppleAccount[0x00000000191e0dd88] (AppleAccount.__TEXT.__text + 174432)
Summary: AppleAccount`-[AADeviceInfo appleIDClientIdentifier] Address: AppleAccount[0x00000000191e0e07c] (AppleAccount.__TEXT.__text + 175188)
Summary: AppleAccount`-[AADeviceInfo clientInfoHeader] Address: AppleAccount[0x00000000191e0e360] (AppleAccount.__TEXT.__text + 175928)
Summary: AppleAccount`-[AADeviceInfo userAgentHeader] Address: AppleAccount[0x00000000191e0e428] (AppleAccount.__TEXT.__text + 176128)
Summary: AppleAccount`__31+[AADeviceInfo isInternalBuild]_block_invoke Address: AppleAccount[0x00000000191e0e44c] (AppleAccount.__TEXT.__text + 176164)
Summary: AppleAccount`+[AADeviceInfo isMultiUserMode] Address: AppleAccount[0x00000000191e0e48c] (AppleAccount.__TEXT.__text + 176228)
Summary: AppleAccount`__31+[AADeviceInfo isMultiUserMode]_block_invoke Address: AppleAccount[0x00000000191e0e50c] (AppleAccount.__TEXT.__text + 176356)
Summary: AppleAccount`+[AADeviceInfo hasiCloudSignOutRestriction] Address: AppleAccount[0x00000000191e0e514] (AppleAccount.__TEXT.__text + 176364)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) infoDictionary] Address: AppleAccount[0x00000000191e0e558] (AppleAccount.__TEXT.__text + 176432)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) uidid] Address: AppleAccount[0x00000000191e0e59c] (AppleAccount.__TEXT.__text + 176500)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) osVersion] Address: AppleAccount[0x00000000191e0e5e0] (AppleAccount.__TEXT.__text + 176568)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) serialNumber] Address: AppleAccount[0x00000000191e0e624] (AppleAccount.__TEXT.__text + 176636)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) apnsToken] Address: AppleAccount[0x00000000191e0e668] (AppleAccount.__TEXT.__text + 176704)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) appleIDClientIdentifier] Address: AppleAccount[0x00000000191e0e6ac] (AppleAccount.__TEXT.__text + 176772)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) clientInfoHeader] Address: AppleAccount[0x00000000191e0e6f0] (AppleAccount.__TEXT.__text + 176840)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) userAgentHeader] Address: AppleAccount[0x00000000191e0e734] (AppleAccount.__TEXT.__text + 176908)
Summary: AppleAccount`+[AADeviceInfo(Deprecated) productVersion]

```

"authenticateWithContext:"

```

(lldb) image lookup -rn "authenticateWithContext:"
9 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346) /Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit:
Address: AuthKit[0x00000000191d3c2f8] (AuthKit.__TEXT.__text + 42196)
Summary: AuthKit`-[AKAppleIDAuthenticationController authenticateWithContext:completion:] Address: AuthKit[0x00000000191d3c864] (AuthKit.__TEXT.__text + 43584)
Summary: AuthKit`__72-[AKAppleIDAuthenticationController authenticateWithContext:completion:]_block_invoke Address: AuthKit[0x00000000191d3cab4] (AuthKit.__TEXT.__text + 44176)
Summary: AuthKit`__72-[AKAppleIDAuthenticationController authenticateWithContext:completion:]_block_invoke.148 Address: AuthKit[0x00000000191d3cb8c] (AuthKit.__TEXT.__text + 44392)
Summary: AuthKit`__72-[AKAppleIDAuthenticationController authenticateWithContext:completion:]_block_invoke.158 Address: AuthKit[0x00000000191d83278] (AuthKit.__TEXT.__text + 332884)

```



```

Summary: AuthKit`-[AKAuthHandlerImpl reauthenticateWithContext:completion:] Address: AuthKit[0x0000000191d83314] (AuthKit.__TEXT.__text + 333040)
Summary: AuthKit`__58-[AKAuthHandlerImpl reauthenticateWithContext:completion:]_block_invoke Address: AuthKit[0x0000000191d83330] (AuthKit.__TEXT.__text + 333068)
Summary: AuthKit`-[AKAuthHandlerImpl reauthenticateWithContext:completionWithResults:] Address: AuthKit[0x0000000191da41b4] (AuthKit.__TEXT.__text + 467856)
Summary: AuthKit`-[AKAppleIDAuthenticationController authenticateWithContext:completion:].cold.1 Address: AuthKit[0x0000000191da41e0] (AuthKit.__TEXT.__text + 467900)

Summary: AuthKit`-[AKAppleIDAuthenticationController authenticateWithContext:completion:].cold.2
3 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346) /Symbols/System/Library/PrivateFrameworks/AuthKitUI.framework/AuthKitUI:
Address: AuthKitUI[0x00000001c4e5a714] (AuthKitUI.__TEXT.__text + 95384)
Summary: AuthKitUI`-[AKBaseSignInViewController _authenticateWithContext:] Address: AuthKitUI[0x00000001c4e5a814] (AuthKitUI.__TEXT.__text + 95640)
Summary: AuthKitUI`__55-[AKBaseSignInViewController _authenticateWithContext:]_block_invoke Address: AuthKitUI[0x00000001c4e5a8dc] (AuthKitUI.__TEXT.__text + 95840)
Summary: AuthKitUI`__55-[AKBaseSignInViewController _authenticateWithContext:]_block_invoke_2
3 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346) /Symbols/System/Library/PrivateFrameworks/iTunesStore.framework/iTunesStore:
Address: iTunesStore[0x00000001b9403104] (iTunesStore.__TEXT.__text + 251036)
Summary: iTunesStore`-[ISStoreURLOperation _authenticateWithContext:error:] Address: iTunesStore[0x00000001b94038e4] (iTunesStore.__TEXT.__text + 253052)
Summary: iTunesStore`__54-[ISStoreURLOperation _authenticateWithContext:error:]_block_invoke Address: iTunesStore[0x00000001b94041ac] (iTunesStore.__TEXT.__text + 255300)
Summary: iTunesStore`__54-[ISStoreURLOperation _authenticateWithContext:error:]_block_invoke.680

```

image lookup -s

stats

```

(lldb) image lookup -s stats
1 symbols match 'stats' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/dyld:
Address: dyld[0x0000000000004c324] (dyld.__TEXT.__text + 308004)
Summary: dyld`stats64
1 symbols match 'stats' in /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCore.framework/AwemeCore:
...

```

image lookup -r -s

objc_msgSend

```

(lldb) image lookup -r -s objc_msgSend

```

```
3 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneros/Aweme.app/Frameworks/AwemeCore.framework/AwemeCore:
    Address: AwemeCore[0x000000001147d8fc] (AwemeCore.__BD_TEXT.__stubs + 28356)
    Summary: AwemeCore`symbol stub for: objc_msgSend Address: AwemeCore[0x000000001147d908] (AwemeCore.__BD_TEXT.__stubs + 28368)
    Summary: AwemeCore`symbol stub for: objc_msgSendSuper Address: AwemeCore[0x000000001147d914] (AwemeCore.__BD_TEXT.__stubs + 28380)
    Summary: AwemeCore`symbol stub for: objc_msgSendSuper2
3 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/System/Library/Frameworks/Foundation.framework/Foundation:
    Address: Foundation[0x000000001809b9850] (Foundation.__TEXT.__stubs + 11736)
    Summary: Foundation`symbol stub for: objc_msgSend Address: Foundation[0x000000001809b985c] (Foundation.__TEXT.__stubs + 11748)
    Summary: Foundation`symbol stub for: objc_msgSendSuper Address: Foundation[0x000000001809b9868] (Foundation.__TEXT.__stubs + 11760)
    Summary: Foundation`symbol stub for: objc_msgSendSuper2
7 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/usr/lib/libobjc.A.dylib:
    Address: libobjc.A.dylib[0x000000001801b7e60] (libobjc.A.dylib.__TEXT.__text + 2816)
    Summary: libobjc.A.dylib`_objc_msgSend_uncached Address: libobjc.A.dylib[0x000000001801b7b20] (libobjc.A.dylib.__TEXT.__text + 1984)
    Summary: libobjc.A.dylib`objc_msgSend Address: libobjc.A.dylib[0x000000001801b7ce0] (libobjc.A.dylib.__TEXT.__text + 2432)
    Summary: libobjc.A.dylib`objc_msgSendSuper Address: libobjc.A.dylib[0x000000001801b7d60] (libobjc.A.dylib.__TEXT.__text + 2560)
    Summary: libobjc.A.dylib`objc_msgSendSuper2 Address: libobjc.A.dylib[0x000000001801b8080] (libobjc.A.dylib.__TEXT.__text + 3360)
    Summary: libobjc.A.dylib`objc_msgSendSuper2_debug Address: libobjc.A.dylib[0x000000001801b8060] (libobjc.A.dylib.__TEXT.__text + 3328)
    Summary: libobjc.A.dylib`objc_msgSend_debug Address: libobjc.A.dylib[0x000000001801b8040] (libobjc.A.dylib.__TEXT.__text + 3296)
    Summary: libobjc.A.dylib`objc_msgSend_noarg
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/usr/lib/swift/libswiftCore.dylib:
    Address: libswiftCore.dylib[0x0000000018dc77d54] (libswiftCore.dylib.__TEXT.__stubs + 1188)
    Summary: libswiftCore.dylib`symbol stub for: objc_msgSend Address: libswiftCore.dylib[0x0000000018dc77d60] (libswiftCore.dylib.__TEXT.__stubs + 1200)
    Summary: libswiftCore.dylib`symbol stub for: objc_msgSendSuper2
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneros/Aweme.app/Frameworks/libAwemeDylib.dylib:
    Address: libAwemeDylib.dylib[0x0000000000002ced4] (libAwemeDylib.dylib.__TEXT.__stubs + 1296)
    Summary: libAwemeDylib.dylib`symbol stub for: objc_msgSend Address: libAwemeDylib.dylib[0x0000000000002cee0] (libAwemeDylib.dylib.__TEXT.__stubs + 1308)
    Summary: libAwemeDylib.dylib`symbol stub for: objc_msgSendSuper2
. . .
. . .
. . .
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneros/Aweme.app/Frameworks/VolcEngineRTC.framework/VolcEngineRTC:
```

```
Address: VolcEngineRTC[0x000000000098b35c] (VolcEngineRTC.__BD_TEXT.__stubs + 6
360)
Summary: VolcEngineRTC`symbol stub for: objc_msgSend Address: VolcEngine
RTC[0x000000000098b368] (VolcEngineRTC.__BD_TEXT.__stubs + 6372)
Summary: VolcEngineRTC`symbol stub for: objc_msgSendSuper2
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneos/Aw
eme.app/Frameworks/byteaudio.framework/byteaudio:
Address: byteaudio[0x00000000001a353c] (byteaudio.__TEXT.__stubs + 2700)
Summary: byteaudio symbol stub for: objc_msgSend Address: byteaudio[0x00
000000001a3548] (byteaudio.__TEXT.__stubs + 2712)
Summary: byteaudio`symbol stub for: objc_msgSendSuper2
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/System/Library/Frameworks/ARKit.framework/ARKit:
Address: ARKit[0x0000000019c696678] (ARKit.__TEXT.__stubs + 7488)
Summary: ARKit`symbol stub for: objc_msgSend Address: ARKit[0x0000000019c
696684] (ARKit.__TEXT.__stubs + 7500)
Summary: ARKit`symbol stub for: objc_msgSendSuper2
* * *
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/System/Library/PrivateFrameworks/QuickLo
okSupport.framework/QuickLookSupport:
Address: QuickLookSupport[0x000000001b2c55850] (QuickLookSupport.__TEXT.__stubs
+ 1824)
Summary: QuickLookSupport`symbol stub for: objc_msgSend Address: QuickLo
okSupport[0x000000001b2c5585c] (QuickLookSupport.__TEXT.__stubs + 1836)
Summary: QuickLookSupport`symbol stub for: objc_msgSendSuper2
4 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneos/Aw
eme.app/Frameworks/libcycrypt.dylib:
Address: libcycrypt.dylib[0x000000000069cb8] (libcycrypt.dylib.__TEXT.__text +
416952)
Summary: libcycrypt.dylib`$objc_msgSend(OpaqueJSContext const*, OpaqueJSValue*,
OpaqueJSValue*, unsigned long, OpaqueJSValue const* const*, OpaqueJSValue const**)
Address: libcycrypt.dylib[0x00000000006d318] (libcycrypt.dylib.__TEXT.__text + 430
872)
Summary: libcycrypt.dylib`$objc_msgSend(OpaqueJSContext const*, OpaqueJSValue*,
OpaqueJSValue*, unsigned long, OpaqueJSValue const* const*) Address: libcycrypt
.dylib[0x0000000000788a0] (libcycrypt.dylib.__TEXT.__stubs + 2208)
Summary: libcycrypt.dylib`symbol stub for: objc_msgSend Address: libcyacr
ipt.dylib[0x0000000000788ac] (libcycrypt.dylib.__TEXT.__stubs + 2220)
Summary: libcycrypt.dylib`symbol stub for: objc_msgSendSuper2
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneos/Aw
eme.app/Frameworks/RevealServer.framework/RevealServer:
Address: RevealServer[0x000000000056734] (RevealServer.__TEXT.__stubs + 2484)
Summary: RevealServer`symbol stub for: objc_msgSend Address: RevealServer
[0x000000000056740] (RevealServer.__TEXT.__stubs + 2496)
Summary: RevealServer`symbol stub for: objc_msgSendSuper2
1 symbols match the regular expression 'objc_msgSend' in /Library/MobileSubstrate/Dynam
icLibraries/MuJiaBaiHuoTweak.dylib:
Address: MuJiaBaiHuoTweak.dylib[0x000000000015d54] (MuJiaBaiHuoTweak.dylib.__T
EXT.__stubs + 480)
Summary: MuJiaBaiHuoTweak.dylib`symbol stub for: objc_msgSend
2 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
```

```
r/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/System/Library/PrivateFrameworks/AppSSOC
ore.framework/AppSSOCore:
  Address: AppSSOCore[0x00000001b909b7f8] (AppSSOCore.__TEXT.__stubs + 576)
  Summary: AppSSOCore`symbol stub for: objc_msgSend      Address: AppSSOCore[0x
00000001b909b804] (AppSSOCore.__TEXT.__stubs + 588)
  Summary: AppSSOCore`symbol stub for: objc_msgSendSuper2
1 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/usr/lib/libusrtcp.dylib:
  Address: libusrtcp.dylib[0x00000001857ffb18] (libusrtcp.dylib.__TEXT.__stubs +
1944)
  Summary: libusrtcp.dylib`symbol stub for: objc_msgSendSuper2
1 symbols match the regular expression 'objc_msgSend' in /Users/crifan/Library/Develope
r/Xcode/iOS DeviceSupport/13.7 (17H35)/Symbols/usr/lib/libboringsssl.dylib:
  Address: libboringsssl.dylib[0x00000001853f8710] (libboringsssl.dylib.__TEXT.__st
ubs + 2676)
  Summary: libboringsssl.dylib`symbol stub for: objc_msgSendSuper2
```

handlePressGesture

```
(lldb) image lookup -r -s handlePressGesture
5 symbols match the regular expression 'handlePressGesture' in /Users/crifan/Library/De
veloper/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/
UIKitCore.framework/UIKitCore:
  Address: UIKitCore[0x0000000182e04cac] (UIKitCore.__TEXT.__text + 7638380)
  Summary: UIKitCore`-[UIButtonBarButtonVisualProviderIOS _handlePressGesture:]
  Address: UIKitCore[0x0000000183060acc] (UIKitCore.__TEXT.__text + 10111884)
  Summary: UIKitCore`-[UIPressClickInteractionDriver _handlePressGesture:]
  Address: UIKitCore[0x0000000183060b70] (UIKitCore.__TEXT.__text + 10112048)
  Summary: UIKitCore`__54-[UIPressClickInteractionDriver _handlePressGesture:]_b
lock_invoke
  Address: UIKitCore[0x0000000183192f80] (UIKitCore.__TEXT.__text + 11366464)
  Summary: UIKitCore`-[UIPointerInteractionHoverDriver _handlePressGesture:]
  Address: UIKitCore[0x000000018332fc64] (UIKitCore.__TEXT.__text + 13057316)
  Summary: UIKitCore`-[UIBandSelectionInteraction _handlePressGesture:]
2 symbols match the regular expression 'handlePressGesture' in /Users/crifan/Library/De
veloper/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/
CameraEditKit.framework/CameraEditKit:
  Address: CameraEditKit[0x00000001c57d46b4] (CameraEditKit.__TEXT.__text + 54112)
  Summary: CameraEditKit`-[CEKWheelScrubberView _handlePressGesture:]
  Address: CameraEditKit[0x00000001c57e8320] (CameraEditKit.__TEXT.__text + 135116)
  Summary: CameraEditKit`-[CEKLightingControl _handlePressGesture:]
```

_nextButtonSelected

```
(lldb) image lookup -r -s _nextButtonSelected
3 symbols match the regular expression '_nextButtonSelected' in /Users/crifan/Library/D
eveloper/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks
/AppleAccountUI.framework/AppleAccountUI:
  Address: AppleAccountUI[0x00000001b0f8b978] (AppleAccountUI.__TEXT.__text + 695
92)
  Summary: AppleAccountUI`-[AAUISignInViewController _nextButtonSelected:]
```

...

_performAuthenticationForAccount

```
(lldb) image lookup -r -s _performAuthenticationForAccount
3 symbols match the regular expression '_performAuthenticationForAccount' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccountUI.framework/AppleAccountUI:
    Address: AppleAccountUI[0x00000001b0fcc8d8] (AppleAccountUI.__TEXT.__text + 335672)
    Summary: AppleAccountUI`-[AAUISignInController _performAuthenticationForAccount:serviceType:inViewController:completion:] Address: AppleAccountUI[0x00000001b0fccda4] (AppleAccountUI.__TEXT.__text + 336900)
    Summary: AppleAccountUI`__97-[AAUISignInController _performAuthenticationForAccount:serviceType:inViewController:completion:]_block_invoke Address: AppleAccountUI[0x00000001b0fcc8c] (AppleAccountUI.__TEXT.__text + 337132)
    Summary: AppleAccountUI`__97-[AAUISignInController _performAuthenticationForAccount:serviceType:inViewController:completion:]_block_invoke_2
```

authenticationContext

```
(lldb) image lookup -r -s "authenticationContext"
1 symbols match the regular expression 'authenticationContext' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/AppleAccountUI.framework/AppleAccountUI:
    Address: AppleAccountUI[0x00000001adfbc228] (AppleAccountUI.__TEXT.__text + 65592)
    Summary: AppleAccountUI`-[AAUISignInViewController authenticationContext]
...
2 symbols match the regular expression 'authenticationContext' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/AppleMediaServices.framework/AppleMediaServices:
    Address: AppleMediaServices[0x00000001848bc000] (AppleMediaServices.__TEXT.__text + 3797688)
    Summary: AppleMediaServices`+[AMSPaymentSheetTask _authenticationContextForRequest:] Address: AppleMediaServices[0x00000001848c0d24] (AppleMediaServices.__TEXT.__text + 3817436)
    Summary: AppleMediaServices`-[AMSPaymentSheetTask authenticationContext]
2 symbols match the regular expression 'authenticationContext' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit:
    Address: AuthKit[0x0000000192ee8440] (AuthKit.__TEXT.__text + 367116)
    Summary: AuthKit`-[AKAuthHandlerImpl buildReauthenticationContextFromContext:error:] Address: AuthKit[0x0000000192f14e50] (AuthKit.__TEXT.__text + 549916)
    Summary: AuthKit`-[AKAuthHandlerImpl buildReauthenticationContextFromContext:error:].cold.1
...
12 symbols match the regular expression 'authenticationContext' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateFrameworks/iTunesStore.framework/iTunesStore:
    Address: iTunesStore[0x00000001b6c89c58] (iTunesStore.__TEXT.__text + 110356)
    Summary: iTunesStore`-[ISURLOperation authenticationContext] Address: iT
```

```
unesStore[0x00000001b6ca3ee4] (iTunesStore.__TEXT.__text + 217504)
    Summary: iTunesStore`-[ISDataProvider authenticationContext] Address: iT
unesStore[0x00000001b6cbb05c] (iTunesStore.__TEXT.__text + 312088)
    Summary: iTunesStore`-[ISStoreAuthenticateOperation authenticationContext]
Address: iTunesStore[0x00000001b6cc1e38] (iTunesStore.__TEXT.__text + 340212)
    Summary: iTunesStore`-[ISQRCodeDialog initWithDialogDictionary:authenticationCo
ntext:] Address: iTunesStore[0x00000001b6cc4494] (iTunesStore.__TEXT.__text + 35
0032)
    Summary: iTunesStore`-[ISDialog initWithDialogDictionary:authenticationContext:]
Address: iTunesStore[0x00000001b6cc762c] (iTunesStore.__TEXT.__text + 362728)
    Summary: iTunesStore`-[ISDialog authenticationContext] Address: iTunesSt
ore[0x00000001b6cce2e8] (iTunesStore.__TEXT.__text + 390564)
    Summary: iTunesStore`-[ISServerAuthenticationOperation authenticationContext]
Address: iTunesStore[0x00000001d861aab4] (iTunesStore.__DATA.__objc_ivar + 780)
    Summary: iTunesStore`ISStoreAuthenticateOperation._authenticationContext
Address: iTunesStore[0x00000001d861ab5c] (iTunesStore.__DATA.__objc_ivar + 948)
    Summary: iTunesStore`ISDialog._authenticationContext Address: iTunesStore
[0x00000001d861ac54] (iTunesStore.__DATA.__objc_ivar + 1196)
    Summary: iTunesStore`ISServerAuthenticationOperation._authenticationContext
Address: iTunesStore[0x00000001d861a8a4] (iTunesStore.__DATA.__objc_ivar + 252)
    Summary: iTunesStore`ISDataProvider._authenticationContext Address: iTun
esStore[0x00000001d861a870] (iTunesStore.__DATA.__objc_ivar + 200)
    Summary: iTunesStore`ISURLOperation._authenticationContext
15 symbols match the regular expression 'authenticationContext' in /Users/crifan/Librar
y/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/PrivateF
rameworks/iTunesStoreUI.framework/iTunesStoreUI:
    Address: iTunesStoreUI[0x00000001ab93300c] (iTunesStoreUI.__TEXT.__text + 40536)

    Summary: iTunesStoreUI`-[SUScriptInterface authenticationContext] Address
s: iTunesStoreUI[0x00000001ab95d8a8] (iTunesStoreUI.__TEXT.__text + 214772)
    Summary: iTunesStoreUI`-[SUStorePageViewController authenticationContext]
Address: iTunesStoreUI[0x00000001ab974b14] (iTunesStoreUI.__TEXT.__text + 309600)
    Summary: iTunesStoreUI`-[SUWebViewController authenticationContext] Address
ess: iTunesStoreUI[0x00000001ab98df6c] (iTunesStoreUI.__TEXT.__text + 413112)
    Summary: iTunesStoreUI`-[SUScriptXMLHTTPStoreRequest authenticationContext]
Address: iTunesStoreUI[0x00000001aba03ce4] (iTunesStoreUI.__TEXT.__text + 895792)
    Summary: iTunesStoreUI`-[SUWebViewManager authenticationContext] Address
: iTunesStoreUI[0x00000001aba3fef8] (iTunesStoreUI.__TEXT.__text + 1142084)
    Summary: iTunesStoreUI`-[SUXMLHTTPStoreRequestOperation authenticationContext]
Address: iTunesStoreUI[0x00000001aba54544] (iTunesStoreUI.__TEXT.__text + 1225616
)
    Summary: iTunesStoreUI`-[SUAddiTunesPassOperation authenticationContext]
Address: iTunesStoreUI[0x00000001d81062f8] (iTunesStoreUI.__DATA.__objc_ivar + 1088)
    Summary: iTunesStoreUI`SUWebViewController._authenticationContext Address
s: iTunesStoreUI[0x00000001d8106514] (iTunesStoreUI.__DATA.__objc_ivar + 1628)
    Summary: iTunesStoreUI`SUScriptXMLHTTPStoreRequest._authenticationContext
Address: iTunesStoreUI[0x00000001d81065c8] (iTunesStoreUI.__DATA.__objc_ivar + 1808)
    Summary: iTunesStoreUI`SUScriptAppleIdAuthenticationOperation._authenticationCo
ntext Address: iTunesStoreUI[0x00000001d81069f4] (iTunesStoreUI.__DATA.__objc_iv
ar + 2876)
    Summary: iTunesStoreUI`SUWebViewManager._authenticationContext Address:
iTunesStoreUI[0x00000001d8106f64] (iTunesStoreUI.__DATA.__objc_ivar + 4268)
    Summary: iTunesStoreUI`SUXMLHTTPStoreRequestOperation._authenticationContext
Address: iTunesStoreUI[0x00000001d81070f4] (iTunesStoreUI.__DATA.__objc_ivar + 4668
)
    Summary: iTunesStoreUI`SUAddiTunesPassOperation._authenticationContext A
```

```

address: iTunesStoreUI[0x00000001d8106698] (iTunesStoreUI.__DATA.__objc_ivar + 2016)
  Summary: iTunesStoreUI`SUScriptAuthenticationOperation.__authenticationContext
  Address: iTunesStoreUI[0x00000001d8105f28] (iTunesStoreUI.__DATA.__objc_ivar + 112)
)
  Summary: iTunesStoreUI`SUScriptInterface.__authenticationContext

```

```
image lookup -va == im loo -va
```

Apple Store相关

akd

```

(lldb) image lookup -va 0x10461c710
Address: akd[0x00000001000bc710] (akd.__TEXT.__stubs + 2028)
Summary: akd`symbol stub for: voucher_mach_msg_set
Module: file = "/System/Library/PrivateFrameworks/AuthKit.framework/akd", arch =
"arm64"
Symbol: id = {0x0000023b}, range = [0x000000010461c710-0x000000010461c71c), name =
"voucher_mach_msg_set"

```

AppleAccount

```

(lldb) image lookup -va 0x0000000195d31f14
Address: AppleAccount[0x0000000191e59f14] (AppleAccount.__TEXT.__text + 486124)
Summary: AppleAccount`-[AALoginAccountRequest urlRequest] + 300
Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19
A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount", ar
ch = "arm64"
Symbol: id = {0x00000b24}, range = [0x0000000195d31de8-0x0000000195d320f0), name =
"-[AALoginAccountRequest urlRequest]"

(lldb) image lookup -va 0x0000000195cbcd20
Address: AppleAccount[0x0000000191de4d20] (AppleAccount.__TEXT.__text + 6392)
Summary: AppleAccount`__51-[AAResponse performRequestWithSession:withHandler:]_blo
ck_invoke + 100
Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19
A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount", ar
ch = "arm64"
Symbol: id = {0x0000001c}, range = [0x0000000195cbccbc-0x0000000195cbcf0), mang
led="__51-[AAResponse performRequestWithSession:withHandler:]_block_invoke"

```

AppleAccountUI

```

(lldb) image lookup -va 0x1b1750550
Address: AppleAccountUI[0x00000001b0f8c550] (AppleAccountUI.__TEXT.__text + 72624)

Summary: AppleAccountUI`__62-[AAUISignInViewController _attemptAuthenticationWith
Context:]_block_invoke

```

```

Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19
A346)/Symbols/System/Library/PrivateFrameworks/AppleAccountUI.framework/AppleAccountUI"
, arch = "arm64"
Symbol: id = {0x0000017c}, range = [0x00000001b1750550-0x00000001b1750624), mang
led="__62-[AAUISignInViewController _attemptAuthenticationWithContext:]_block_invoke"

```

AuthKit

```

(lldb) image lookup -va 0x0000000194509de8
Address: AuthKit[0x0000000192e99de8] (AuthKit.__TEXT.__text + 46004)
Summary: AuthKit`-[AKAppleIDAuthenticationController authenticateWithContext:comp
letion:] + 724
Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19
B74) arm64e/Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit", arch =
"arm64e"
Symbol: id = {0x00000111}, range = [0x0000000194509b14-0x000000019450a0e0), name=
"-[AKAppleIDAuthenticationController authenticateWithContext:completion:]"

```

AwemeCore

```

(lldb) image lookup -v -a 0x112d57730
Address: AwemeCore[0x0000000010047730] (AwemeCore.__BD_TEXT.__text + 175257392)
Summary: AwemeCore`___lldb_unnamed_symbol1653310$$AwemeCore
Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjo
xbkibsdwekuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCore.frame
work/AwemeCore", arch = "arm64"
Symbol: id = {0x001e5c5c}, range = [0x0000000112d57730-0x0000000112d57a28), name=
"___lldb_unnamed_symbol1653310$$AwemeCore"

```

```

(lldb) image lookup -v -a 0x116d51950
Address: AwemeCore[0x000000001147d950] (AwemeCore.__BD_TEXT.__stubs + 28440)
Summary:
Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjo
xbkibsdwekuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCore.frame
work/AwemeCore", arch = "arm64"

```

CFNetwork

```

(lldb) im loo -va 0xa916260182319ea0
Address: CFNetwork[0x00000000180ddea0] (CFNetwork.__TEXT.__text + 2665792)
Summary: CFNetwork`___lldb_unnamed_symbol12486$$CFNetwork
Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19
B74) arm64e/Symbols/System/Library/Frameworks/CFNetwork.framework/CFNetwork", arch = "a
rm64e"
Symbol: id = {0x00003a5a}, range = [0x00000000182319ea0-0x00000000182319f94), name=
"___lldb_unnamed_symbol12486$$CFNetwork"

```


image lookup -vs

SecTrustEvaluateFastAsync

```
(lldb) image lookup -vs SecTrustEvaluateFastAsync
1 symbols match 'SecTrustEvaluateFastAsync' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security:
    Address: Security[0x0000000189226304] (Security.__TEXT.__text + 944572)
    Summary: Security`SecTrustEvaluateFastAsync
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security", arch = "arm64e"
    Symbol: id = {0x000020a1}, range = [0x000000018b40a304-0x000000018b40a3b0), name="SecTrustEvaluateFastAsync"
```

AADeviceInfo

```
(lldb) image lookup -vs "AADeviceInfo"
2 symbols match 'AADeviceInfo' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
    Address: AppleAccount[0x00000001f032e1d8] (AppleAccount.__DATA_DIRTY.__objc_data + 800)
    Summary: (void *)0x00000001f4206200: AADeviceInfo
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount", arch = "arm64"
    Symbol: id = {0x00001876}, range = [0x00000001f42061d8-0x00000001f4206200), name="AADeviceInfo", mangled="OBJC_CLASS_$_AADeviceInfo"

    Address: AppleAccount[0x00000001f032e200] (AppleAccount.__DATA_DIRTY.__objc_data + 840)
    Summary: (void *)0x00000001f3fff1e0: NSObject
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount", arch = "arm64"
    Symbol: id = {0x0000195f}, range = [0x00000001f4206200-0x00000001f4206228), name="AADeviceInfo", mangled="OBJC_METACLASS_$_AADeviceInfo"
```

__lldb_unnamed_symbol972

```
(lldb) image lookup -vs "__lldb_unnamed_symbol972"
1 symbols match '__lldb_unnamed_symbol972' in /cores/dyld:
    Address: dyld[0x0000000000002d558] (dyld.__TEXT.__text + 181592)
    Summary: dyld dyld3::OverflowSafeArray<dyld4::PrebuiltObjC::ObjCOptimizerImage::ObjCObject, 4294967295ul>::push_back(dyld4::PrebuiltObjC::ObjCOptimizerImage::ObjCObject const&)
    Module: file = "/cores/dyld", arch = "arm64"
    Symbol: id = {0x000003cc}, range = [0x0000000102d71558-0x0000000102d7165c), na
```

```
me="dyld3::OverflowSafeArray<dyld4::PrebuiltObjC::ObjCOptimizerImage::ObjCObject, 42949
67295ul>::push_back(dyld4::PrebuiltObjC::ObjCOptimizerImage::ObjCObject const&)", mangl
ed="_ZN5dyld317overflowSafeArrayIN5dyld412PrebuiltObjC18ObjCOptimizerImage10ObjCObjectE
Lm4294967295EE9push_backERKS4_"
```

...

```
1 symbols match '___lldb_unnamed_symbol1972' in /Users/crifan/Library/Developer/Xcode/iO
S DeviceSupport/15.0 (19A346)/Symbols/usr/lib/libMobileGestalt.dylib:
  Address: libMobileGestalt.dylib[0x00000001908f6c40] (libMobileGestalt.dylib.__T
EXT.__text + 90176)
```

```
  Summary: libMobileGestalt.dylib`___lldb_unnamed_symbol1972
```

```
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/usr/lib/libMobileGestalt.dylib", arch = "arm64"
```

```
  Symbol: id = {0x000003cc}, range = [0x0000000194db6c40-0x0000000194db76e4], na
me="___lldb_unnamed_symbol1972"
```

...

image lookup -vn

___lldb_unnamed_symbol2575\$\$akd

```
(lldb) image lookup -vn ___lldb_unnamed_symbol2575$$akd
1 match found in /System/Library/PrivateFrameworks/AuthKit.framework/akd:
  Address: akd[0x00000001000a0460] (akd.__TEXT.__text + 639120)
  Summary: akd`___lldb_unnamed_symbol2575$$akd
  Module: file = "/System/Library/PrivateFrameworks/AuthKit.framework/akd", arch
= "arm64"
  Symbol: id = {0x00000c53}, range = [0x00000001021f8460-0x00000001021fa928], na
me="___lldb_unnamed_symbol2575$$akd"
```

"-[AALoginAccountRequest urlRequest]"

```
(lldb) image lookup -vn "-[AALoginAccountRequest urlRequest]"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/
Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
  Address: AppleAccount[0x0000000191e59de8] (AppleAccount.__TEXT.__text + 485824)
  Summary: AppleAccount`-[AALoginAccountRequest urlRequest]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount",
arch = "arm64"
  Symbol: id = {0x00000b24}, range = [0x0000000195d31de8-0x0000000195d320f0], na
me="-[AALoginAccountRequest urlRequest]"
```

"-[AKAppleIDAuthenticationContextManager shouldContinueWithAuthenticationResults:error:forContextID:compl etion:]"

```
(lldb) image lookup -vn "-[AKAppleIDAuthenticationContextManager shouldContinueWithAuth
enticationResults:error:forContextID:completion:]"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/
Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit:
  Address: AuthKit[0x0000000191d4b66c] (AuthKit.__TEXT.__text + 104520)
  Summary: AuthKit`-[AKAppleIDAuthenticationContextManager shouldContinueWithAuth
enticationResults:error:forContextID:completion:]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/System/Library/PrivateFrameworks/AuthKit.framework/AuthKit", arch = "ar
m64"
  Symbol: id = {0x00000025a}, range = [0x0000000019250f66c-0x0000000019250fd18], na
me="-[AKAppleIDAuthenticationContextManager shouldContinueWithAuthenticationResults:err
or:forContextID:completion:]"
```

"+[AADeviceInfo(Deprecated) udid]"

```
(lldb) image lookup -vn "+[AADeviceInfo(Deprecated) udid]"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/
Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
  Address: AppleAccount[0x0000000191e0e558] (AppleAccount.__TEXT.__text + 176432)
  Summary: AppleAccount`+[AADeviceInfo(Deprecated) udid]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount",
arch = "arm64"
  Symbol: id = {0x000000473}, range = [0x00000000195ce6558-0x00000000195ce659c], na
me="+[AADeviceInfo(Deprecated) udid]"
```

"-[AADeviceInfo udid]"

```
(lldb) image lookup -vn "-[AADeviceInfo udid]"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/
Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
  Address: AppleAccount[0x0000000191e0d970] (AppleAccount.__TEXT.__text + 173384)
  Summary: AppleAccount`-[AADeviceInfo udid]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount",
arch = "arm64"
  Symbol: id = {0x000000453}, range = [0x00000000197599970-0x00000000197599990], na
me="-[AADeviceInfo udid]"
```

```
(lldb) image lookup -vn "+[AADeviceInfo udid]"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/
Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
  Address: AppleAccount[0x0000000191e0e558] (AppleAccount.__TEXT.__text + 176432)
  Summary: AppleAccount`+[AADeviceInfo(Deprecated) udid]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount",
arch = "arm64"
  Symbol: id = {0x000000473}, range = [0x0000000019759a558-0x0000000019759a59c], na
me="+[AADeviceInfo(Deprecated) udid]"
```

image lookup -vrn

AppleAccount

```
(lldb) image lookup -vrn "AADeviceInfo udid"
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (19A346)/
Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount:
  Address: AppleAccount[0x0000000191e0d970] (AppleAccount.__TEXT.__text + 173384)
  Summary: AppleAccount`-[AADeviceInfo udid]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.0 (
19A346)/Symbols/System/Library/PrivateFrameworks/AppleAccount.framework/AppleAccount",
arch = "arm64"
  Symbol: id = {0x000000453}, range = [0x0000000195ce5970-0x0000000195ce5990), na
me="-[AADeviceInfo udid]"
```

image lookup -type == im loo -t

ZJJPerson

```
(lldb) image lookup -t ZJJPerson
Best match found in
/Users/zhaojing/Library/Developer/CoreSimulator/Devices/ED430020-74C1-4FF3-A595-492C4A714F41/data/Containers/
Bundle/Application/256675C0-DE3A-400B-9431-079A890A6255/ZJJAllModule_Example.app/ZJJAllModule_Example:
id = {0x7fffffff004bcbe}, name = "ZJJPerson", byte-size = 24, decl = ZJJPerson.h:13, compiler_type =
"@interface ZJJPerson : NSObject{
  NSString * _name;
  NSInteger _age;
}
@property(nonatomic, readwrite, getter = name, setter = setName:) NSString *name;
@property(nonatomic, assign, readwrite, getter = age, setter = setAge:) NSInteger age;
@end"
(lldb) @稀土掘金技术社区
```

Person

```
(lldb) image lookup -type Person
Best match found in /Users/liu_david/Library/Developer/Xcode/DerivedData/TestWeak-egnzd
bndwslikvcheqmcxvkqnbw/Build/Products/Debug/TestWeak:
id = {0x10000002b}, name = "Person", byte-size = 24, decl = main.m:12, compiler_type =
"@interface Person : NSObject{
  BOOL _isMan;
  short _age;
  NSString * _name;
}
@property(nonatomic, copy, readwrite, getter = name, setter = setName:) NSString *name;
@property(nonatomic, assign, readwrite, getter = age, setter = setAge:) short age;
@property(nonatomic, assign, readwrite, getter = isMan, setter = setIsMan:) BOOL isMan;
@end"

(lldb) image lookup -type NSObject
Best match found in /Users/liu_david/Library/Developer/Xcode/DerivedData/TestWeak-egnzd
bndwslikvcheqmcxvkqnbw/Build/Products/Debug/TestWeak:
id = {0x7fffffff000002e8}, name = "NSObject", byte-size = 8, decl = NSObject.h:53, comp
iler_type = "@interface NSObject{
```

```
Class isa;
}
@end"

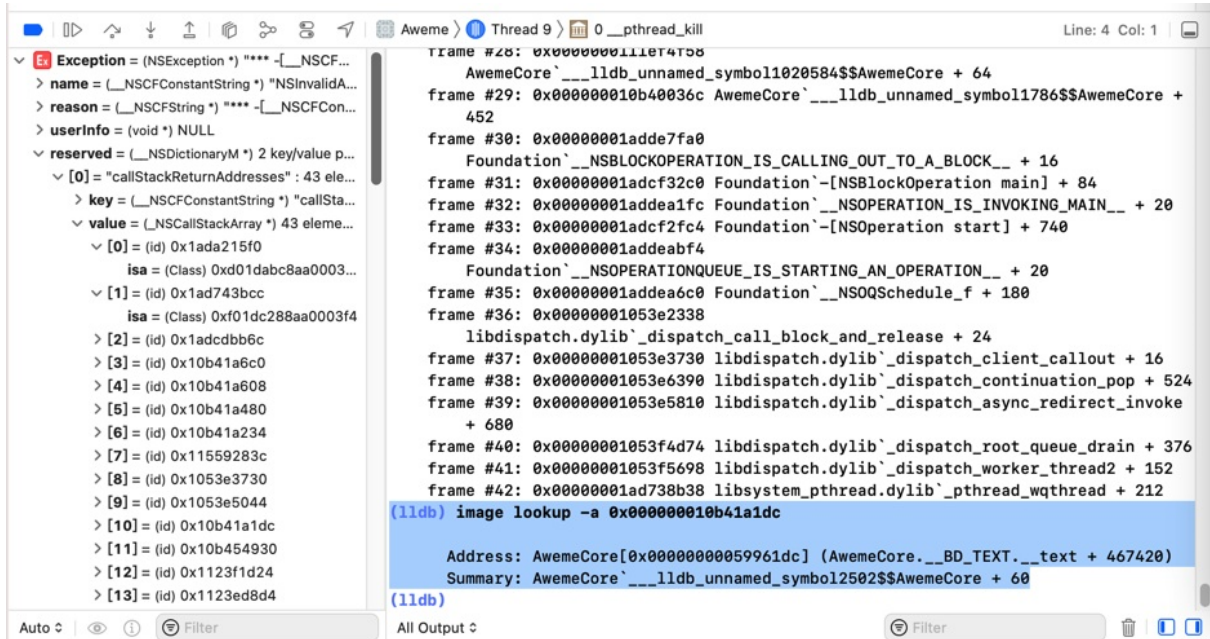
(lldb) im loo -t Person
Best match found in /Users/liu_david/Library/Developer/Xcode/DerivedData/TestWeak-egnzd
bndwsikvcheqmcxvkqnwbw/Build/Products/Debug/TestWeak:
id = {0x10000002b}, name = "Person", byte-size = 24, decl = main.m:12, compiler_type =
"@interface Person : NSObject{
    BOOL _isMan;
    short _age;
    NSString * _name;
}
@property(nonatomic, copy, readwrite, getter = name, setter = setName:) NSString *name;
@property(nonatomic, assign, readwrite, getter = age, setter = setAge:) short age;
@property(nonatomic, assign, readwrite, getter = isMan, setter = setIsMan:) BOOL isMan;
@end"
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:44:27

image lookup的多命令对比举例

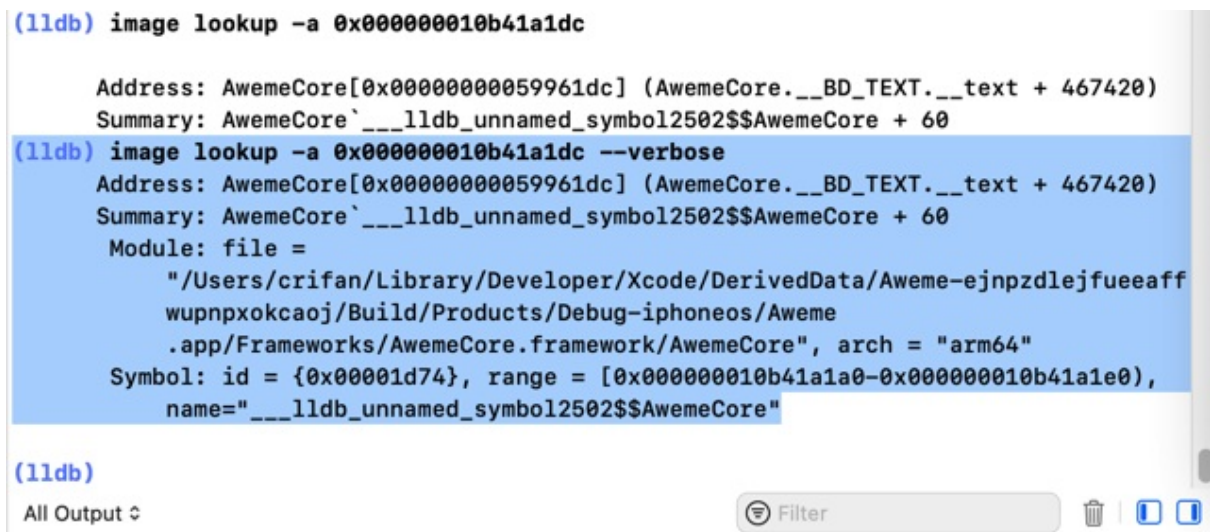
`image lookup -a` vs `image lookup -v -a == image lookup -va`

AwemeCore 0x000000010b41a1dc



```
(lldb) image lookup -a 0x000000010b41a1dc
Address: AwemeCore[0x0000000059961dc] (AwemeCore.__BD_TEXT.__text + 467420)
Summary: AwemeCore`___lldb_unnamed_symbol12502$$AwemeCore + 60
```

加上verbose, 可以输出更加详细的信息:



```
(lldb) image lookup -a 0x000000010b41a1dc --verbose
Address: AwemeCore[0x0000000059961dc] (AwemeCore.__BD_TEXT.__text + 467420)
Summary: AwemeCore`___lldb_unnamed_symbol12502$$AwemeCore + 60
```

```

Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-ejnpz
dlejfueeaffwupnpxokcaoj/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCor
e.framework/AwemeCore", arch = "arm64"
Symbol: id = {0x00001d74}, range = [0x000000010b41a1a0-0x000000010b41a1e0), n
ame="___lldb_unnamed_symbol12502$$AwemeCore"

```

AwemeCore 0x10f04f810

```

(lldb) image lookup -a 0x10f04f810
Address: AwemeCore[0x00000000c587810] (AwemeCore.__BD_TEXT.__text + 113653776)
Summary: AwemeCore`___lldb_unnamed_symbol1023498$$AwemeCore + 32

(lldb) image lookup -v -a 0x10f04f810
Address: AwemeCore[0x00000000c587810] (AwemeCore.__BD_TEXT.__text + 113653776)
Summary: AwemeCore`___lldb_unnamed_symbol1023498$$AwemeCore + 32
Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcidjo
xbkibsdwekuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCore.frame
work/AwemeCore", arch = "arm64"
Symbol: id = {0x0014c028}, range = [0x000000010f04f7f0-0x000000010f04f844), name=
"___lldb_unnamed_symbol1023498$$AwemeCore"

```

Module_Framework 0x0000000107dd18c4

```

(lldb) image lookup -a 0x0000000107dd18c4
Address: Module_Framework[0x00000000198d8c4] (Module_Framework.__TEXT.__text + 2
6777796)
Summary: Module_Framework`-[HAMPlayerInternal setStatus:]_block + 72

(lldb) image lookup -v -a 0x0000000107dd18c4
Address: Module_Framework[0x00000000198d8c4] (Module_Framework.__TEXT.__text + 2
6777796)
Summary: Module_Framework`-[HAMPlayerInternal setStatus:]_block + 72
Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/youtube-haehnt
tylwejqpfqywoxzkydyfa/Build/Products/Debug-iphonios/youtube.app/Frameworks/Module_Fram
ework.framework/Module_Framework", arch = "arm64"
Symbol: id = {0x0003eee7}, range = [0x0000000107dd187c-0x0000000107dd1904), name=
"-[HAMPlayerInternal setStatus:]_block"

```

akd 0x104B0C460

```

(lldb) image lookup -a 0x104B0C460
Address: akd[0x00000001000a0460] (akd.__TEXT.__text + 639120)
Summary: akd`___lldb_unnamed_symbol12575$$akd
(lldb) image lookup -va 0x104B0C460
Address: akd[0x00000001000a0460] (akd.__TEXT.__text + 639120)
Summary: akd`___lldb_unnamed_symbol12575$$akd
Module: file = "/System/Library/PrivateFrameworks/AuthKit.framework/akd", arch =
"arm64"
Symbol: id = {0x00000c53}, range = [0x0000000104b0c460-0x0000000104b0e928), name=

```

```
"__lldb_unnamed_symbol12575$$akd"
```

AppleStoreCore 0x000000010373d398

```
(lldb) image lookup -a 0x000000010373d398
Address: AppleStoreCore[0x000000000000a9398] (AppleStoreCore.__TEXT.__text + 673828)
)
Summary: AppleStoreCore`__lldb_unnamed_symbol13027$$AppleStoreCore + 872

(lldb) image lookup -va 0x000000010373d398
Address: AppleStoreCore[0x000000000000a9398] (AppleStoreCore.__TEXT.__text + 673828)
)
Summary: AppleStoreCore`__lldb_unnamed_symbol13027$$AppleStoreCore + 872
Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-fbcdzphr
bokcgxhejx1slydryaa/Build/Products/Debug-iphonios/Jolly.app/Frameworks/AppleStoreCore.
framework/AppleStoreCore", arch = "arm64"
Symbol: id = {0x000040fa}, range = [0x000000010373d030-0x000000010373d640), name=
"__lldb_unnamed_symbol13027$$AppleStoreCore"
```

image lookup -vs vs **image lookup -vn**

Security SSLHandshake

```
(lldb) image lookup -vs SSLHandshake
1 symbols match 'SSLHandshake' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security:
Address: Security[0x00000001891501c0] (Security.__TEXT.__text + 67704)
Summary: Security`SSLHandshake
Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security", arch = "arm64e"
Symbol: id = {0x00001da4}, range = [0x000000018acc01c0-0x000000018acc0320), name="SSLHandshake"

(lldb) image lookup -vn SSLHandshake
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security:
Address: Security[0x00000001891501c0] (Security.__TEXT.__text + 67704)
Summary: Security`SSLHandshake
Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security", arch = "arm64e"
Symbol: id = {0x00001da4}, range = [0x000000018acc01c0-0x000000018acc0320), name="SSLHandshake"
```

Security SecTrustEvaluateFastAsync

```
(lldb) image lookup -vs SecTrustEvaluateFastAsync
```



```

1 symbols match 'SecTrustEvaluateFastAsync' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security:
    Address: Security[0x0000000189226304] (Security.__TEXT.__text + 944572)
    Summary: Security`SecTrustEvaluateFastAsync
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security", arch = "arm64e"
    Symbol: id = {0x000020a1}, range = [0x000000018b40a304-0x000000018b40a3b0), name="SecTrustEvaluateFastAsync"

(lldb) image lookup -vn SecTrustEvaluateFastAsync
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security:
    Address: Security[0x0000000189226304] (Security.__TEXT.__text + 944572)
    Summary: Security`SecTrustEvaluateFastAsync
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/Security.framework/Security", arch = "arm64e"
    Symbol: id = {0x000020a1}, range = [0x000000018b40a304-0x000000018b40a3b0), name="SecTrustEvaluateFastAsync"

```

initialize

- 用 `symbol` 找: 只找到1个

```

(lldb) image lookup -vs initialize
1 symbols match 'initialize' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/WatchConnectivity.framework/WatchConnectivity:
    Address: WatchConnectivity[0x00000001bf42bdb8] (WatchConnectivity.__TEXT.__text + 123420)
    Summary: WatchConnectivity`initialize
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/System/Library/Frameworks/WatchConnectivity.framework/WatchConnectivity", arch = "arm64e"
    Symbol: id = {0x0000021c}, range = [0x00000001c0507db8-0x00000001c0507e9c), name="initialize"

```

- 用 `n = name` 找: 找到非常多

此处只列出部分内容:

```

(lldb) image lookup -vn initialize
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/usr/lib/dyld:
    Address: dyld[0x000000000000afd0] (dyld.__TEXT.__text + 40912)
    Summary: dyld`dyld4::RuntimeState::initialize()
    Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74) arm64e/Symbols/usr/lib/dyld", arch = "arm64e"
    Symbol: id = {0x000000c7}, range = [0x00000001032b6fd0-0x00000001032b70bc), name="dyld4::RuntimeState::initialize()", mangled="_ZN5dyld412RuntimeState10initializeEv"

```

```

2 matches found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (19B74)
arm64e/Symbols/System/Library/Frameworks/AVKit.framework/AVKit:
  Address: AVKit[0x000000019a386f48] (AVKit.__TEXT.__text + 330556)
  Summary: AVKit`+[AVChapter initialize]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (
19B74) arm64e/Symbols/System/Library/Frameworks/AVKit.framework/AVKit", arch = "arm64e"
  Symbol: id = {0x00000063c}, range = [0x000000019b462f48-0x000000019b462fb8], na
me="+[AVChapter initialize]"
  Address: AVKit[0x000000019a41a474] (AVKit.__TEXT.__text + 933992)
  Summary: AVKit`+[AVPlayerController initialize]
  Module: file = "/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/15.1 (
19B74) arm64e/Symbols/System/Library/Frameworks/AVKit.framework/AVKit", arch = "arm64e"
  Symbol: id = {0x00001316}, range = [0x000000019b4f6474-0x000000019b4f6500], na
me="+[AVPlayerController initialize]"
...
1 match found in /Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-fbcdzphrbokcgx
hejxlslydrdyaa/Build/Products/Debug-iphonios/Jolly.app/Frameworks/RevealServer.framewor
k/RevealServer:
  Address: RevealServer[0x00000000004a9dc] (RevealServer.__TEXT.__text + 277116)
  Summary: RevealServer`+[IBAHTTPConnection initialize]
  Module: file = "/Users/crifan/Library/Developer/Xcode/DerivedData/Jolly-fbcdzp
hrbokcgxhejxlslydrdyaa/Build/Products/Debug-iphonios/Jolly.app/Frameworks/RevealServer.
framework/RevealServer", arch = "arm64e"
  Symbol: id = {0x00001189}, range = [0x00000001048b69dc-0x00000001048b6a30], na
me="+[IBAHTTPConnection initialize]"
...

```

image lookup -s vs image lookup -n vs image lookup -r -s

stats

```

(lldb) image lookup -s stats
1 symbols match 'stats' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.
3.1 (17D50)/Symbols/usr/lib/dyld:
  Address: dyld[0x000000000004c324] (dyld.__TEXT.__text + 308004)
  Summary: dyld stats64
1 symbols match 'stats' in /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fsw
cidjoxbkibsdwেকuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCore.
framework/AwemeCore:
...
1 symbols match 'stats' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.
3.1 (17D50)/Symbols/usr/lib/system/libsystem_kernel.dylib:
  Address: libsystem_kernel.dylib[0x000000018028c8c4] (libsystem_kernel.dylib.__T
EXT.__text + 160068)
  Summary: libsystem_kernel.dylib`stats

(lldb) image lookup -n stats
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)
/Symbols/usr/lib/dyld:
  Address: dyld[0x000000000004c324] (dyld.__TEXT.__text + 308004)

```

```
Summary: dyld`statfs64
1 match found in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)
/Symbols/usr/lib/system/libsystem_kernel.dylib:
Address: libsystem_kernel.dylib[0x000000018028c8c4] (libsystem_kernel.dylib.__T
EXT.__text + 160068)
Summary: libsystem_kernel.dylib`statfs

(lldb) image lookup -r -s statfs
4 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/dyld:
Address: dyld[0x000000000004c0a4] (dyld.__TEXT.__text + 307364)
Summary: dyld`fstatfs64 Address: dyld[0x000000000004c0a4] (dyld.__TEXT.__
_text + 307364)
Summary: dyld`fstatfs64 Address: dyld[0x000000000004c324] (dyld.__TEXT.__
_text + 308004)
Summary: dyld`statfs64 Address: dyld[0x000000000004c324] (dyld.__TEXT.__
text + 308004)
Summary: dyld`statfs64
2 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneos/Aweme.ap
p/Frameworks/AwemeCore.framework/AwemeCore:
...
2 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/Frameworks/Foundation.framewo
rk/Foundation:
Address: Foundation[0x00000001809a3160] (Foundation.__TEXT.__stubs + 10188)
Summary: Foundation`symbol stub for: fstatfs Address: Foundation[0x000000
001809a3f34] (Foundation.__TEXT.__stubs + 13728)
Summary: Foundation`symbol stub for: statfs
2 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/DerivedData/Aweme-fswcidjoxbkibsdwekuzlsfcdqls/Build/Products/Debug-iphoneos/Aweme.ap
p/Frameworks/libAwemeDylib.dylib:
Address: libAwemeDylib.dylib[0x000000000003896c] (libAwemeDylib.dylib.__TEXT.__
stubs + 1104)
Summary: libAwemeDylib.dylib`symbol stub for: fstatfs Address: libAwemeD
ylib.dylib[0x0000000000038ca8] (libAwemeDylib.dylib.__TEXT.__stubs + 1932)
Summary: libAwemeDylib.dylib`symbol stub for: statfs
2 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/libsqlite3.dylib:
Address: libsqlite3.dylib[0x0000000182217478] (libsqlite3.dylib.__TEXT.__stubs
+ 744)
Summary: libsqlite3.dylib`symbol stub for: fstatfs Address: libsqlite3.d
ylib[0x0000000182217730] (libsqlite3.dylib.__TEXT.__stubs + 1440)
Summary: libsqlite3.dylib`symbol stub for: statfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/Frameworks/IOKit.framework/Ve
rsions/A/IOKit:
Address: IOKit[0x0000000181458f9c] (IOKit.__TEXT.__stubs + 5088)
Summary: IOKit`symbol stub for: statfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcod
e/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/Frameworks/ImageIO.framework/
ImageIO:
Address: ImageIO[0x0000000180e541ac] (ImageIO.__TEXT.__stubs + 8340)
Summary: ImageIO`symbol stub for: statfs
```

```
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/Frameworks/SystemConfiguration.framework/SystemConfiguration:
    Address: SystemConfiguration[0x000000018073646c] (SystemConfiguration.__TEXT.__stubs + 3000)
    Summary: SystemConfiguration`symbol stub for: fstatfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/system/libcopyfile.dylib:
    Address: libcopyfile.dylib[0x00000001a05aa260] (libcopyfile.dylib.__TEXT.__stubs + 624)
    Summary: libcopyfile.dylib`symbol stub for: fstatfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/system/libremovefile.dylib:
    Address: libremovefile.dylib[0x00000001b5fc0b74] (libremovefile.dylib.__TEXT.__stubs + 144)
    Summary: libremovefile.dylib`symbol stub for: fstatfs
3 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/system/libsystem_c.dylib:
    Address: libsystem_c.dylib[0x00000001800883a4] (libsystem_c.dylib.__TEXT.__text + 4268)
    Summary: libsystem_c.dylib`cvt_statfs_to_statvfs      Address: libsystem_c.dylib[0x00000001800fa510] (libsystem_c.dylib.__TEXT.__stubs + 1272)
    Summary: libsystem_c.dylib`symbol stub for: fstatfs      Address: libsystem_c.dylib[0x00000001800fab10] (libsystem_c.dylib.__TEXT.__stubs + 2808)
    Summary: libsystem_c.dylib`symbol stub for: statfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/system/libsystem_info.dylib:
    Address: libsystem_info.dylib[0x00000001803634ac] (libsystem_info.dylib.__TEXT.__stubs + 1836)
    Summary: libsystem_info.dylib`symbol stub for: statfs
4 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/system/libsystem_kernel.dylib:
    Address: libsystem_kernel.dylib[0x000000018028b3fc] (libsystem_kernel.dylib.__TEXT.__text + 154748)
    Summary: libsystem_kernel.dylib`fstatfs      Address: libsystem_kernel.dylib[0x000000018028b3fc] (libsystem_kernel.dylib.__TEXT.__text + 154748)
    Summary: libsystem_kernel.dylib`fstatfs      Address: libsystem_kernel.dylib[0x000000018028c8c4] (libsystem_kernel.dylib.__TEXT.__text + 160068)
    Summary: libsystem_kernel.dylib`statfs      Address: libsystem_kernel.dylib[0x000000018028c8c4] (libsystem_kernel.dylib.__TEXT.__text + 160068)
    Summary: libsystem_kernel.dylib`statfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/libarchive.2.dylib:
    Address: libarchive.2.dylib[0x00000001a027e28c] (libarchive.2.dylib.__TEXT.__stubs + 1212)
    Summary: libarchive.2.dylib`symbol stub for: fstatfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/usr/lib/libMobileGestalt.dylib:
    Address: libMobileGestalt.dylib[0x000000018149fd18] (libMobileGestalt.dylib.__TEXT.__stubs + 2280)
    Summary: libMobileGestalt.dylib`symbol stub for: statfs
...
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/PrivateFrameworks/MobileKeyBag.framework/MobileKeyBag:
    Address: MobileKeyBag[0x00000001822554d4] (MobileKeyBag.__TEXT.__stubs + 2460)
```

```
Summary: MobileKeyBag`symbol stub for: statfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/PrivateFrameworks/APFS.framework/APFS:
  Address: APFS[0x00000001aa73f114] (APFS.__TEXT.__stubs + 948)
  Summary: APFS`symbol stub for: fstatfs
1 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/PrivateFrameworks/ktrace.framework/ktrace:
  Address: ktrace[0x00000001b3d0555c] (ktrace.__TEXT.__stubs + 3552)
  Summary: ktrace`symbol stub for: statfs
...
2 symbols match the regular expression 'statfs' in /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.3.1 (17D50)/Symbols/System/Library/PrivateFrameworks/CoreServicesInternal.framework/CoreServicesInternal:
  Address: CoreServicesInternal[0x000000018178fd50] (CoreServicesInternal.__TEXT.__text + 116648)
  Summary: CoreServicesInternal`GetStatfsByFSID(fsId, statfs*, int)
  Address: CoreServicesInternal[0x0000000181798ca8] (CoreServicesInternal.__TEXT.__stubs + 3264)

  Summary: CoreServicesInternal`symbol stub for: statfs
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:44:21

image lookup的帮助

```
(lldb) help image lookup
```

```
Look up information within executable and dependent shared library images.
```

```
Syntax: target modules lookup <cmd-options> [<filename> [ filename <...>]]
```

```
Command Options Usage:
```

```
target modules lookup [-Av] -a <address-expression> [-o <offset>] [<filename> [ filename <...>]]
```

```
target modules lookup [-Arv] -s <symbol> [<filename> [ filename <...>]]
```

```
target modules lookup [-Aiv] -f <filename> [-l <linenum>] [<filename> [ filename <...>]]
```

```
target modules lookup [-Airv] -F <function-name> [<filename> [ filename <...>]]
```

```
target modules lookup [-Airv] -n <function-or-symbol> [ filename [<filename> <...>]]
```

```
target modules lookup [-Av] -t <name> [ filename [<filename> <...>]]
```

```
-A ( --all )
```

```
Print all matches, not just the best match, if a best match is available.
```

```
-F <function-name> ( --function <function-name> )
```

```
Lookup a function by name in the debug symbols in one or more target modules.
```

```
-a <address-expression> ( --address <address-expression> )
```

```
Lookup an address in one or more target modules.
```

```
-f <filename> ( --file <filename> )
```

```
Lookup a file by fullpath or basename in one or more target modules.
```

```
-i ( --no-inlines )
```

```
Ignore inline entries (must be used in conjunction with --file or --function).
```

```
-l <linenum> ( --line <linenum> )
```

```
Lookup a line number in a file (must be used in conjunction with --file).
```

```
-n <function-or-symbol> ( --name <function-or-symbol> )
```

```
Lookup a function or symbol by name in one or more target modules.
```

```
-o <offset> ( --offset <offset> )
```

```
When looking up an address subtract <offset> from any addresses before doing the lookup.
```

```
-r ( --regex )
```

```
The <name> argument for name lookups are regular expressions.
```

```
-s <symbol> ( --symbol <symbol> )
```

```
Lookup a symbol by name in the symbol tables in one or more target modules.
```

```
-t <name> ( --type <name> )  
  Lookup a type by name in the debug symbols in one or more target  
  modules.
```

```
-v ( --verbose )  
  Enable verbose lookup information.
```

This **command** takes options and free-form arguments. If your arguments resemble option specifiers (i.e., they start with a - or --), you must use ' -- ' between the end of the **command** options and the beginning of the arguments.

'image' is an abbreviation for 'target modules'

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 20:08:46

image list

- 输出加载的镜像image列表

```
image list -o -f
```

- 输出加载的镜像image列表，只输出特定二进制

```
image list -o -f | grep moduleName
```

- 举例

```
image list -o -f | grep AwemeCore  
image list -o -f | grep Module_Framework
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 21:47:59

image list举例

Aweme

```
(lldb) image list -o -f
[ 0] 0x0000000004874000 /private/var/containers/Bundle/Application/9AB25481-0AD3-435C-A02E-68F9623535BB/Aweme.app/Aweme (0x0000000104874000)
[ 1] 0x0000000104b78000 /Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.4.1 (17E262)/Symbols/usr/lib/dyld
```

- 另外某次的类似例子的截图

◦

-
- 说明
 - 第一个是app本身的二进制
 - `/private/var/containers/Bundle/Application/9AB25481-0AD3-435C-A02E-68F9623535BB/Aweme.app/Aweme`
 - 此处 ALSR 的基地址是: `0x0000000004874000`
 - 第二个是 `dyld`
 - `/Users/crifan/Library/Developer/Xcode/iOS DeviceSupport/13.4.1(17E262)/Symbols/usr/lib/dyld`

AwemeCore

```
(lldb) image list -o -f | grep AwemeCore
[ 0] 0x0000000100adc000 /Users/crifan/Library/Developer/Xcode/DerivedData/Aweme-fswcid
joxbk1bsdwekuzlsfcdqls/Build/Products/Debug-iphonios/Aweme.app/Frameworks/AwemeCore.fra
mework/AwemeCore
```

Module_Framework

- 带 `grep` 的

```
(lldb) image list -o -f | grep Module_Framework
```

```
[ 0] 0x0000000104238000 /Users/crifan/Library/Developer/Xcode/DerivedData/youtube-dvlfmmtvybrcdraorwznbwepoae/Build/Products/Debug-iphones/youtube.app/Frameworks/Module_Framework.framework/Module_Framework
```

- 带 `grep` 输出后，继续计算函数地址和查找相关函数

```
(lldb) image list -o -f | grep Module_Framework
[ 0] 0x0000000102b50000 /Users/crifan/Library/Developer/Xcode/DerivedData/youtube-dvlfmmtvybrcdraorwznbwepoae/Build/Products/Debug-iphones/youtube.app/Frameworks/Module_Framework.framework/Module_Framework
(lldb) p/x 0x0000000102b50000 + 0x10470B8
(long) $0 = 0x0000000103b970b8
(lldb) im loc -a 0x0000000103b970b8
Address: Module_Framework[0x00000000010470b8] (Module_Framework.__TEXT.__text + 17051832)
Summary: Module_Framework`___lldb_unnamed_symbol12565$$Module_Framework
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 21:45:26



crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-10-25 21:45:50

image list的help语法

```
(lldb) help image list
```

List current executable and dependent shared library images.

Syntax: target modules list <cmd-options> [<shlib-name> [<shlib-name> [...]]]

Command Options Usage:

```
target modules list [-ghou] [-a <address-expression>] [-A[<width>]] [-b[<width>]] [-d[<width>]] [-f[<width>]] [-m[<width>]] [-p[<none>]] [-r[<width>]] [-s[<width>]] [-S[<width>]] [-t[<width>]] [-<shlib-name> [ <shlib-name> [...] ] ]
```

-A [<width>] (**--arch** [<width>])

Display the architecture when listing images.

-S [<width>] (**--symfile-unique** [<width>])

Display the symbol **file** with optional width only **if** it is different from the executable object file.

-a <address-expression> (**--address** <address-expression>)

Display the image at this address.

-b [<width>] (**--basename**=[<width>])

Display the **basename** with optional width **for** the image object file.

-d [<width>] (**--directory**=[<width>])

Display the directory with optional width **for** the image object file.

-f [<width>] (**--fullpath**=[<width>])

Display the fullpath to the image object file.

-g (**--global**)

Display the modules from the global module list, not just the current target.

-h (**--header**)

Display the image base address as a load address **if** debugging, a **file** address otherwise.

-m [<width>] (**--mod-time**=[<width>])

Display the modification **time** with optional width of the module.

-o (**--offset**)

Display the image load address offset from the base **file** address (the **slide** amount).

-p [<none>] (**--pointer**=[<none>])

Display the module pointer.

-r [<width>] (**--ref-count**=[<width>])

Display the reference count **if** the module is still **in** the shared module cache.

```
-s [width>] ( --symfile=[width] )  
    Display the fullpath to the image symbol file with optional width.  
  
-t [width>] ( --triple=[width] )  
    Display the triple when listing images.  
  
-u ( --uuid )  
    Display the UUID when listing images.
```

This `command` takes options and free-form arguments. If your arguments resemble option specifiers (i.e., they start with a - or --), you must use ' -- ' between the end of the `command` options and the beginning of the arguments.

'image' is an abbreviation for 'target modules'

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 21:46:50

image dump

- Dump all sections from the main executable and any shared libraries.

```
image dump sections
```

- Dump all sections in the a.out module

```
image dump sections a.out
```

- Dump all symbols from the main executable and any shared libraries

```
image dump symtab
```

- Dump all symbols in a.out and liba.so

```
image dump symtab a.out liba.so
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:01:51

register

TODO:

- 【记录】lldb命令使用心得: register
- 【记录】lldb命令使用心得: register read

register举例

```
(lldb) reg r x0 x1 x2 x3
x0 = 0x000000029e100ea0
x1 = 0x0000000000000000
x2 = 0x0000000292566f00
x3 = 0x0000000289d922e0
```

```
(lldb) reg r x8
x8 = 0x0000000107e2fef8 Module_Framework`vtable for video_streaming::OnesieReque
stProto + 16
```

register语法

```
(lldb) help register
Commands to access registers for the current thread and stack frame.
```

Syntax: register [read|write] ...

The following subcommands are supported:

```
read -- Dump the contents of one or more register values from the
       current frame. If no register is specified, dumps them all.
write -- Modify a single register value.
```

For more help on any particular subcommand, type 'help <command> <subcommand>'.

register read语法

```
(lldb) help register read
Dump the contents of one or more register values from the current frame. If no
register is specified, dumps them all.
```

Syntax: register read <cmd-options> [<register-name> [<register-name> [...]]]

Command Options Usage:

```
register read [-A] [-f <format>] [-G <gdb-format>] [-s <index>] [<register-name> [<re
gister-name> [...]]]
```

```
register read [-Aa] [-f <format>] [-G gdb-format] [ <register-name> [ <register-name>
[...]]]
```

-A (**--alternate**)

Display register names using the alternate register name if there is one.

-G <gdb-format> (**--gdb-format** <gdb-format>)

Specify a format using a GDB format specifier string.

-a (**--all**)

Show all register sets.

-f <format> (**--format** <format>)

Specify a format to be used for display.

-s <index> (**--set** <index>)

Specify which register sets to dump by index.

This **command** takes options and free-form arguments. If your arguments resemble option specifiers (i.e., they start with a - or --), you must use ' -- ' between the end of the **command** options and the beginning of the arguments.

expression

TODO:

- 【记录】lldb命令使用心得: expression
- 【记录】lldb命令使用心得: p和po
 - 【整理】Xcode中lldb命令对比: po和p

p和po

- `p == expression --`
- `po == expression -0 --`

expression语法

```
(lldb) help expression
Evaluate an expression on the current thread. Displays any returned value with
LLDB's default formatting. Expects 'raw' input (see 'help raw-input'.)

Syntax: expression <cmd-options> -- <expr>

Command Options Usage:
  expression [-AFLORTgp] [-f <format>] [-G <gdb-format>] [-a <boolean>] [-j <boolean>]
[-X <source-language>] [-v[<description-verbosity>]] [-i <boolean>] [-l <source-languag
e>] [-t <unsigned-integer>] [-u <boolean>] [-d <none>] [-S <boolean>] [-D <count>] [-P
<count>] [-Y[<count>]] [-V <boolean>] [-Z <count>] -- <expr>
  expression [-AFLORTgp] [-a <boolean>] [-j <boolean>] [-X <source-language>] [-i <bool
ean>] [-l <source-language>] [-t <unsigned-integer>] [-u <boolean>] [-d <none>] [-S <bo
olean>] [-D <count>] [-P <count>] [-Y[<count>]] [-V <boolean>] [-Z <count>] -- <expr>
  expression [-r] -- <expr>
  expression <expr>

-A ( --show-all-children )
  Ignore the upper bound on the number of children to show.

-D <count> ( --depth <count> )
  Set the max recurse depth when dumping aggregate types (default is
  infinity).

-F ( --flat )
  Display results in a flat format that uses expression paths for
  each variable or member.

-G <gdb-format> ( --gdb-format <gdb-format> )
  Specify a format using a GDB format specifier string.

-L ( --location )
  Show variable location information.
```

```
-O ( --object-description )
    Display using a language-specific description API, if possible.

-P <count> ( --ptr-depth <count> )
    The number of pointers to be traversed when dumping values (default
    is zero).

-R ( --raw-output )
    Don't use formatting options.

-S <boolean> ( --synthetic-type <boolean> )
    Show the object obeying its synthetic provider, if available.

-T ( --show-types )
    Show variable types when dumping values.

-V <boolean> ( --validate <boolean> )
    Show results of type validators.

-X <source-language> ( --apply-fixits <source-language> )
    If true, simple fix-it hints will be automatically applied to the
    expression.

-Y[ <count> ] ( --no-summary-depth=[ <count> ] )
    Set the depth at which omitting summary information stops (default
    is 1).

-Z <count> ( --element-count <count> )
    Treat the result of the expression as if its type is an array of
    this many values.

-a <boolean> ( --all-threads <boolean> )
    Should we run all threads if the execution doesn't complete on one
    thread.

-d <none> ( --dynamic-type <none> )
    Show the object as its full dynamic type, not its static type, if
    available.
    Values: no-dynamic-values | run-target | no-run-target

-f <format> ( --format <format> )
    Specify a format to be used for display.

-g ( --debug )
    When specified, debug the JIT code by setting a breakpoint on the
    first instruction and forcing breakpoints to not be ignored (-i0)
    and no unwinding to happen on error (-u0).

-i <boolean> ( --ignore-breakpoints <boolean> )
    Ignore breakpoint hits while running expressions

-j <boolean> ( --allow-jit <boolean> )
    Controls whether the expression can fall back to being JITted if
    it's not supported by the interpreter (defaults to true).

-l <source-language> ( --language <source-language> )
```

Specifies the Language to use when parsing the expression. If not `set` the `target.language` setting is used.

`-p (--top-level)`

Interpret the expression as a complete translation unit, without injecting it into the `local` context. Allows declaration of persistent, top-level entities without a `$` prefix.

`-r (--repl)`

Drop into Swift REPL

`-t <unsigned-integer> (--timeout <unsigned-integer>)`

Timeout value (in microseconds) for running the expression.

`-u <boolean> (--unwind-on-error <boolean>)`

Clean up program state if the expression causes a crash, or raises a signal. Note, unlike `gdb` hitting a breakpoint is controlled by another option (`-i`).

`-v[<description-verbosity>] (--description-verbosity [<description-verbosity>])`

How verbose should the output of this expression be, if the object description is asked for.

Values: `compact` | `full`

Single and multi-line expressions:

The expression provided on the `command` line must be a complete expression with no newlines. To evaluate a multi-line expression, hit a `return` after an empty expression, and `lldb` will enter the multi-line expression editor. Hit `return` on an empty line to end the multi-line expression.

Timeouts:

If the expression can be evaluated statically (without running code) then it will be. Otherwise, by default the expression will run on the current thread with a short timeout: currently .25 seconds. If it doesn't return in that time, the evaluation will be interrupted and resumed with all threads running. You can use the `-a` option to disable retrying on all threads. You can use the `-t` option to set a shorter timeout.

User defined variables:

You can define your own variables for convenience or to be used in subsequent expressions. You define them the same way you would define variables in C. If the first character of your user defined variable is a `$`, then the variable's value will be available in future expressions, otherwise it will just be available in the current expression.

Continuing evaluation after a breakpoint:

If the `"-i false"` option is used, and execution is interrupted by a breakpoint hit, once you are done with your investigation, you can either remove the expression execution frames from the stack with `"thread return -x"` or if you are still interested in the expression result you can issue the `"continue"` command and the expression evaluation will complete and the

expression result will be available using the `"thread.completed-expression"` key in the thread format.

Examples:

```
expr my_struct- a = my_array[3]
expr -f bin -- (index * 8) + 5
expr unsigned int $foo = 5
expr char c[] = \"foo\"; c[0]
```

Important Note: Because this `command` takes 'raw' input, if you use any `command` options you must use ' -- ' between the end of the `command` options and the beginning of the raw input.

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

p

- `p == expression --`

常见p的缩写的语法

- 英文

```
p //  
p/x //x hex hexadecimal  
p/d //d decimal signed decimal  
p/u //u unsigned decimal  
p/o //o octal  
p/t //t two binary  
p/a //a address  
p/c //c char character  
p/f //f float  
p/s //s string  
p/r //r raw
```

- 中文

```
p // < 默认打印十进制  
p/x // < 以十六进制打印整数  
p/d // < 以带符号的十进制打印整数  
p/u // < 以无符号的十进制打印整数  
p/o // < 以八进制打印整数  
p/t // < 以二进制打印整数  
p/a // < 以十六进制打印地址  
p/c // < 打印字符常量  
p/f // < 打印浮点数  
p/s // < 打印字符串  
p/r // < 格式化打印
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

po

TODO:

- 【整理】iOS逆向心得：当iPhone锁屏时Xcode中lldb的po会卡死
- 【整理】iOS逆向心得：po异常时NSString的字符串无法像char*一样打印出来
- 【整理】iOS逆向调试心得：po不是对象实例但可以看到是哪个类

-
- `po == expression -0 --`

po举例

```
(lldb) po $x1
8203662366

(lldb) po (SEL)$x1
"stringByAppendingString:"

(lldb) po $x2
<nil>

(lldb) po $x3
{(
    executing
)}

(lldb) po $arg3
<nil>
```

```
(lldb) reg r x1 x2 x3
x1 = 0x000000010a328d0a
x2 = 0x0000000281f79160
x3 = 0x000000016b4b8ea8
(lldb) po 0x000000010a328d0a
4466052362

(lldb) po 0x0000000281f79160
<AWELazyRegisterHandler: 0x281f79160>

(lldb) po 0x000000016b4b8ea8
6095081128
```


memory

TODO:

- 【记录】lldb命令使用心得: memory
- 【记录】lldb命令使用心得: memory read
- 【已解决】iOS逆向: Xcode中lldb从内存中导出二进制数据

memory举例

查看内存中的数据

```
(lldb) x/16gx 0x0000000109117438
0x109117438: 0x0000000000000000 0x0000000000000000
0x109117448: 0x00000001052b6a88 0x00000001052b6ad8
0x109117458: 0x000000010521b03c 0x00000001052b5758
0x109117468: 0x00000001052b5850 0x00000001052b5980
0x109117478: 0x00000001052b5ab0 0x00000001052b6638
0x109117488: 0x00000001052b6674 0x000000010521b40c
0x109117498: 0x00000001052b65fc 0x00000001052b6a10
0x1091174a8: 0x00000001052b6a4c 0x00000001052b5a74
```

memory语法

```
(lldb) help memory
Commands for operating on memory in the current target process.

Syntax: memory <subcommand> [
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:27:12

memory read

心得

(把内存中某段数据) 导出到文件

```
memory read --binary --force --outfile /Users/crifan/dev/tmp/lldb_mem_dump/akd_func2540_arm64e.bin 0x10485d98c 0x1048600dc
```

- 参数解释

- `--outfile /Users/crifan/dev/tmp/lldb_mem_dump/akd_func2540_arm64e.bin`
 - 注意：此处输出文件路径是PC端（此处Mac端），不是移动端（iPhone端）
- `--binary`
 - 以二进制格式导出
 - 否则默认以txt文本格式导出
- `--force`
 - 强制导出大量数据
 - 注：
 - 默认最多只能导出 `1K = 1024` 个字节
 - 不加此参数，会出现警告

```
error: Normally, 'memory read' will not read over 1024 bytes of data.
error: Please use --force to override this restriction just once.
error: or set target.max-memory-read-size if you will often need a large
r limit.
```

- 如果经常导出（超过 1K 的）大量数据，则再去设置：

```
set target.max-memory-read-size
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:25:20

memory read的help语法

```
(lldb) help memory read
```

Read from the memory of the current target process.

Syntax: memory read <cmd-options> <address-expression> [<address-expression>]

Command Options Usage:

```
memory read [-drd] [-f <format>] [-c <count>] [-G <gdb-format>] [-s <byte-size>] [-l <number-per-line>] [-o <filename>] <address-expression> [ <address-expression> ]
```

```
memory read [-dbrd] [-f <format>] [-c <count>] [-s <byte-size>] [-o <filename>] <address-expression> [ <address-expression> ]
```

```
memory read [-AFLORTdrd] -t <name> [-f <format>] [-c <count>] [-G <gdb-format>] [-E <count>] [-o <filename>] [-d <none>] [-S <boolean>] [-D <count>] [-P <count>] [-Y <count>] [-V <boolean>] [-Z <count>] <address-expression> [ <address-expression> ]
```

```
memory read -t <name> [-x <source-language>] <address-expression> [ <address-expression> ]
```

-A (--show-all-children)

Ignore the upper bound on the number of children to show.

-D <count> (--depth <count>)

Set the max recurse depth when dumping aggregate types (default is infinity).

-E <count> (--offset <count>)

How many elements of the specified type to skip before starting to display data.

-F (--flat)

Display results in a flat format that uses expression paths for each variable or member.

-G <gdb-format> (--gdb-format <gdb-format>)

Specify a format using a GDB format specifier string.

-L (--location)

Show variable location information.

-O (--object-description)

Display using a language-specific description API, if possible.

-P <count> (--ptr-depth <count>)

The number of pointers to be traversed when dumping values (default is zero).

-R (--raw-output)

Don't use formatting options.

-S <boolean> (--synthetic-type <boolean>)

Show the object obeying its synthetic provider, if available.

-T (--show-types)

```
    Show variable types when dumping values.

-V <boolean> ( --validate <boolean> )
    Show results of type validators.

-Y[<count>] ( --no-summary-depth=[<count>] )
    Set the depth at which omitting summary information stops (default
    is 1).

-Z <count> ( --element-count <count> )
    Treat the result of the expression as if its type is an array of
    this many values.

-b ( --binary )
    If true, memory will be saved as binary. If false, the memory is
    saved save as an ASCII dump that uses the format, size, count and
    number per line settings.

-c <count> ( --count <count> )
    The number of total items to display.

-d <none> ( --dynamic-type <none> )
    Show the object as its full dynamic type, not its static type, if
    available.
    Values: no-dynamic-values | run-target | no-run-target

-f <format> ( --format <format> )
    Specify a format to be used for display.

-l <number-per-line> ( --num-per-line <number-per-line> )
    The number of items per line to display.

-o <filename> ( --outfile <filename> )
    Specify a path for capturing command output.

-r ( --force )
    Necessary if reading over target.max-memory-read-size bytes.

-s <byte-size> ( --size <byte-size> )
    The size in bytes to use when displaying with the selected format.

-t <name> ( --type <name> )
    The name of a type to view memory as.

-x <source-language> ( --language <source-language> )
    The language of the type to view memory as.

-d ( --append-outfile )
    Append to the file specified with '--outfile <path>'.
```

This command takes options and free-form arguments. If your arguments resemble option specifiers (i.e., they start with a - or --), you must use ' -- ' between the end of the `command` options and the beginning of the arguments.

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:24:21

disassemble

TODO:

【记录】lldb命令使用心得: disassemble

- `disassemble == dis == di`

```
di      -- Disassemble specified instructions in the current target.
        Defaults to the current function for the current thread and
        stack frame.
dis     -- Disassemble specified instructions in the current target.
        Defaults to the current function for the current thread and
        stack frame.
```

disassemble举例

官网的例子

[GDB to LLDB command map — The LLDB Debugger \(llvm.org\)](#)

官网有相关例子:

- Disassemble the current function for the current frame

```
(lldb) disassemble --frame
(lldb) di -f
```

- Disassemble any functions named main

```
(lldb) disassemble --name main
(lldb) di -n main
```

- Disassemble an address range

```
(lldb) disassemble --start-address 0x1eb8 --end-address 0x1ec3
(lldb) di -s 0x1eb8 -e 0x1ec3
```

- Disassemble 20 instructions from a given address

```
(lldb) disassemble --start-address 0x1eb8 --count 20
(lldb) di -s 0x1eb8 -c 20
```

- Show mixed source and disassembly for the current function for the current frame

```
(lldb) disassemble --frame --mixed
(lldb) di -f -m
```


- Disassemble the current function for the current frame and show the opcode bytes

```
(lldb) disassemble --frame --bytes
(lldb) di -f -b
```

- Disassemble the current source line for the current frame

```
(lldb) disassemble --line
(lldb) di -l
```

某次测试的一些命令

- 某次测试的一些命令
 - 如果当前已处于某个函数，则：

```
dis -f == dis = dis -n FunctionName
```

- 其他常见用法：

```
disassemble == dis = di
dis -n "-[AAUISignInViewController _nextButtonSelected:]"
dis -f
dis --start-address 0x1b30cb978 --end-address 0x1b30cb990
dis --start-address 0x1b30cb978 --count 10
dis --frame --mixed
dis --frame --bytes
dis --line
```

当前frame = 当前函数的全部汇编代码

```
dis -f
```

- 效果

```
(lldb) dis -f
AppleAccountUI`-[AAUISignInViewController _nextButtonSelected:]
-> 0x1b30cb978 <+0>: stp    x20, x19, [sp, #-0x20]!
    0x1b30cb97c <+4>: stp    x29, x30, [sp, #0x10]
    0x1b30cb980 <+8>: add    x29, sp, #0x10                ; =0x10
    0x1b30cb984 <+12>: mov    x19, x0
    0x1b30cb988 <+16>: adrp   x8, -84407
    0x1b30cb98c <+20>: add    x1, x8, #0xc7a                ; =0xc7a
    0x1b30cb990 <+24>: bl     0x1b2c70578
    0x1b30cb994 <+28>: adrp   x8, -84407
    0x1b30cb998 <+32>: add    x1, x8, #0xc91                ; =0xc91
    0x1b30cb99c <+36>: mov    x0, x19
    0x1b30cb9a0 <+40>: ldp    x29, x30, [sp, #0x10]
    0x1b30cb9a4 <+44>: ldp    x20, x19, [sp], #0x20
    0x1b30cb9a8 <+48>: b     0x1b2c70578
```

指定函数

```
disassemble -n "-[AAUSignInViewController _nextButtonSelected:]"
```

- 效果

```
(lldb) disassemble -n "-[AAUSignInViewController _nextButtonSelected:]"
AppleAccountUI`-[AAUSignInViewController _nextButtonSelected:]
-> 0x1b30cb978 <+0>: stp    x20, x19, [sp, #-0x20]!
    0x1b30cb97c <+4>: stp    x29, x30, [sp, #0x10]
    0x1b30cb980 <+8>: add    x29, sp, #0x10           ; =0x10
    0x1b30cb984 <+12>: mov    x19, x0
    0x1b30cb988 <+16>: adrp   x8, -84407
    0x1b30cb98c <+20>: add    x1, x8, #0xc7a           ; =0xc7a
    0x1b30cb990 <+24>: bl     0x1b2c70578
    0x1b30cb994 <+28>: adrp   x8, -84407
    0x1b30cb998 <+32>: add    x1, x8, #0xc91           ; =0xc91
    0x1b30cb99c <+36>: mov    x0, x19
    0x1b30cb9a0 <+40>: ldp    x29, x30, [sp, #0x10]
    0x1b30cb9a4 <+44>: ldp    x20, x19, [sp], #0x20
    0x1b30cb9a8 <+48>: b      0x1b2c70578
```

◦

指定地址范围

```
disassemble --start-address 0x1b30cb978 --end-address 0x1b30cb990
```

- 参数说明

- 包括: `start address =0x1b30cb978`
- 不包括: `end address =0x1b30cb990`

- 输出效果

```
(lldb) disassemble --start-address 0x1b30cb978 --end-address 0x1b30cb990
AppleAccountUI`-[AAUSignInViewController _nextButtonSelected:]
-> 0x1b30cb978 <+0>: stp    x20, x19, [sp, #-0x20]!
    0x1b30cb97c <+4>: stp    x29, x30, [sp, #0x10]
    0x1b30cb980 <+8>: add    x29, sp, #0x10           ; =0x10
```

```

0x1b30cb984 <+12> : mov    x19, x0
0x1b30cb988 <+16> : adrp  x8, -84407
0x1b30cb98c <+20> : add   x1, x8, #0xc7a           ; =0xc7a

```

额外加上行数：

```
disassemble --start-address 0x1b30cb978 --count 10
```

- 输出效果

```

(lldb) disassemble --start-address 0x1b30cb978 --count 10
AppleAccountUI`-[AAUISignInViewController _nextButtonSelected:]:
-> 0x1b30cb978 <+0>: stp    x20, x19, [sp, #-0x20]!
    0x1b30cb97c <+4>: stp    x29, x30, [sp, #0x10]
    0x1b30cb980 <+8>: add   x29, sp, #0x10           ; =0x10
    0x1b30cb984 <+12>: mov   x19, x0
    0x1b30cb988 <+16>: adrp  x8, -84407
    0x1b30cb98c <+20>: add   x1, x8, #0xc7a           ; =0xc7a
    0x1b30cb990 <+24>: bl    0x1b2c70578
    0x1b30cb994 <+28>: adrp  x8, -84407
    0x1b30cb998 <+32>: add   x1, x8, #0xc91           ; =0xc91
    0x1b30cb99c <+36>: mov   x0, x19

```

显示opcode

```
disassemble --frame --bytes
```

- 说明

- 和IDA中设置显示opcode的效果类似：解析后的arm指令前面，显示出对应的二进制数据opcode

- 输出效果

```

(lldb) disassemble --frame --bytes
AppleAccountUI`-[AAUISignInViewController _nextButtonSelected:]:
-> 0x1b30cb978 <+0>: 0xa9be4ff4 stp    x20, x19, [sp, #-0x20]!
    0x1b30cb97c <+4>: 0xa9017bfd stp    x29, x30, [sp, #0x10]
    0x1b30cb980 <+8>: 0x910043fd add   x29, sp, #0x10           ; =0x10
    0x1b30cb984 <+12>: 0xaa0003f3 mov   x19, x0
    0x1b30cb988 <+16>: 0xb0f5b248 adrp  x8, -84407
    0x1b30cb98c <+20>: 0x9131e901 add   x1, x8, #0xc7a           ; =0xc7a
    0x1b30cb990 <+24>: 0x97ee92fa bl    0x1b2c70578
    0x1b30cb994 <+28>: 0xb0f5b248 adrp  x8, -84407
    0x1b30cb998 <+32>: 0x91324501 add   x1, x8, #0xc91           ; =0xc91
    0x1b30cb99c <+36>: 0xaa1303e0 mov   x0, x19
    0x1b30cb9a0 <+40>: 0xa9417bfd ldp   x29, x30, [sp, #0x10]
    0x1b30cb9a4 <+44>: 0xa8c24ff4 ldp   x20, x19, [sp], #0x20
    0x1b30cb9a8 <+48>: 0x17ee92f4 b     0x1b2c70578

```

其他例子

显示某个地址的反汇编

```
disassemble -s 0x00000001091694a4
```

- 输出效果

```
(lldb) disassemble -s 0x00000001091694a4
Module_Framework`___lldb_unnamed_symbol171165$$Module_Framework:
    0x1091694a4 <+0>: ret

Module_Framework`___lldb_unnamed_symbol171166$$Module_Framework:
    0x1091694a8 <+0>: b      0x1091a45ac      ; ___lldb_unnamed_symbol1747
29$$Module_Framework

Module_Framework`___lldb_unnamed_symbol171167$$Module_Framework:
    0x1091694ac <+0>: ldr   x2, [x0, #0x10]
    0x1091694b0 <+4>: br    x2

Module_Framework`___lldb_unnamed_symbol171168$$Module_Framework:
    0x1091694b4 <+0>: ret

Module_Framework`___lldb_unnamed_symbol171169$$Module_Framework:
    0x1091694b8 <+0>: b      0x1091a45ac      ; ___lldb_unnamed_symbol1747
29$$Module_Framework

Module_Framework`___lldb_unnamed_symbol171170$$Module_Framework:
    0x1091694bc <+0>: ldr   x2, [x0, #0x10]
    0x1091694c0 <+4>: br    x2
```

指定显示的行数

- 显示从 函数最开始 算起的 20行代码

```
dis -c 20
```

- 显示从 当前PC位置 算起的 20行代码

```
dis --pc -c 20
```

输出效果:

```
(lldb) c
Process 869 resuming
Process 869 stopped
* thread #3, queue = 'com.apple.akd.anisette', stop reason = breakpoint 2.1
  frame #0: 0x0000000102e704d4 akd`___lldb_unnamed_symbol2575$$akd + 116
akd`___lldb_unnamed_symbol2575$$akd:
-> 0x102e704d4 <+116>: ldrsw  x9, [x25, w9, sxtw #2]
    0x102e704d8 <+120>: nop
    0x102e704dc <+124>: ldr   x10, #0x58c7c
    0x102e704e0 <+128>: add   x9, x9, x10
Target 0: (akd) stopped.
(lldb) dis -c 20
akd`___lldb_unnamed_symbol2575$$akd:
    0x102e70460 +0 : sub    sp, sp, #0xf0      ; =0xf0
```

```

0x102e70464 +4 : stp    x28, x27, [sp, #0x90]
0x102e70468 +8 : stp    x26, x25, [sp, #0xa0]
0x102e7046c +12 : stp   x24, x23, [sp, #0xb0]
0x102e70470 +16 : stp   x22, x21, [sp, #0xc0]
0x102e70474 +20 : stp   x20, x19, [sp, #0xd0]
0x102e70478 +24 : stp   x29, x30, [sp, #0xe0]
0x102e7047c +28 : add    x29, sp, #0xe0           ; =0xe0
0x102e70480 +32 : nop
0x102e70484 +36 : ldr    x8, #0x54354           ; (void *)0x00000001f13db058: _
_stack_chk_guard
0x102e70488 +40 : ldr    x8, [x8]
0x102e7048c +44 : stur   x8, [x29, #-0x58]
0x102e70490 +48 : mov    w26, #0x5a87
0x102e70494 +52 : movk   w26, #0x6c24, lsl #16
0x102e70498 +56 : add    x8, x0, #0x6           ; =0x6
0x102e7049c +60 : cmp    x8, #0x5             ; =0x5
0x102e704a0 +64 : ccmn   x0, #0x8, #0x4, hs
0x102e704a4 +68 : mov    w8, #0x1
0x102e704a8 +72 : csel   w8, wzr, w8, eq
0x102e704ac +76 : mov    w11, #0xa5a2
(lldb) dis --pc -c 20
akd`___lldb_unnamed_symbol12575$$akd:
-> 0x102e704d4 +116 : ldrsw  x9, [x25, w9, sxtw #2]
0x102e704d8 +120 : nop
0x102e704dc +124 : ldr    x10, #0x58c7c
0x102e704e0 +128 : add    x9, x9, x10
0x102e704e4 +132 : mov    w22, #-0xafc9
0x102e704e8 +136 : br     x9
0x102e704ec +140 : mov    x23, x1
0x102e704f0 +144 : mov    x28, x0
0x102e704f4 +148 : eor    w8, w8, #0x1
0x102e704f8 +152 : sub    w9, w11, #0x29       ; =0x29
0x102e704fc +156 : madd   w22, w8, w9, w26
0x102e70500 +160 : mov    x20, #0xa930
0x102e70504 +164 : movk   x20, #0xea59, lsl #16
0x102e70508 +168 : movk   x20, #0x9bdd, lsl #32
0x102e7050c +172 : movk   x20, #0xd570, lsl #48
0x102e70510 +176 : add    w8, w22, #0xb       ; =0xb
0x102e70514 +180 : adr    x24, #0x59acc
0x102e70518 +184 : nop
0x102e7051c +188 : ldr    x8, [x24, w8, sxtw #3]
0x102e70520 +192 : sub    x8, x8, #0x2       ; =0x2
(lldb)

```

```
(lldb) c
Process 869 resuming
Process 869 stopped
* thread #3, queue = 'com.apple.akd.anisette', stop reason = breakpoint 2.1
  frame #0: 0x000000102e704d4 akd`__lldb_unnamed_symbol2575$$akd + 116
akd`__lldb_unnamed_symbol2575$$akd:
-> 0x102e704d4 <-+116>: ldrsw x9, [x25, w9, sxtw #2]
0x102e704d8 <-+120>: nop
0x102e704dc <-+124>: ldr x10, #0x58c7c
0x102e704e0 <-+128>: add x9, x9, x10
Target 0: (akd) stopped.
(lldb) dis -c 20
akd`__lldb_unnamed_symbol2575$$akd:
0x102e70460 <-+0>: sub sp, sp, #0xf0 ; =0xf0
0x102e70464 <-+4>: stp x28, x27, [sp, #0x90]
0x102e70468 <-+8>: stp x26, x25, [sp, #0xa0]
0x102e7046c <-+12>: stp x24, x23, [sp, #0xb0]
0x102e70470 <-+16>: stp x22, x21, [sp, #0xc0]
0x102e70474 <-+20>: stp x20, x19, [sp, #0xd0]
0x102e70478 <-+24>: stp x29, x30, [sp, #0xe0]
0x102e7047c <-+28>: add x29, sp, #0xe0 ; =0xe0
0x102e70480 <-+32>: nop
0x102e70484 <-+36>: ldr x8, #0x54354 ; (void *)0x00000001f13db058: __stack_chk_guard
0x102e70488 <-+40>: ldr x8, [x8]
0x102e7048c <-+44>: stur x8, [x29, #-0x58]
0x102e70490 <-+48>: mov w26, #0x5a87
0x102e70494 <-+52>: movk w26, #0x6c24, lsl #16
0x102e70498 <-+56>: add x8, x0, #0x6 ; =0x6
0x102e7049c <-+60>: cmp x8, #0x5 ; =0x5
0x102e704a0 <-+64>: ccmn x0, #0x8, #0x4, hs
0x102e704a4 <-+68>: mov w8, #0x1
0x102e704a8 <-+72>: csel w8, wzr, w8, eq
0x102e704ac <-+76>: mov w11, #0xa5a2
(lldb) dis --pc -c 20
akd`__lldb_unnamed_symbol2575$$akd:
-> 0x102e704d4 <-+116>: ldrsw x9, [x25, w9, sxtw #2]
0x102e704d8 <-+120>: nop
0x102e704dc <-+124>: ldr x10, #0x58c7c
0x102e704e0 <-+128>: add x9, x9, x10
0x102e704e4 <-+132>: mov w22, #-0xafc9
0x102e704e8 <-+136>: br x9
0x102e704ec <-+140>: mov x23, x1
0x102e704f0 <-+144>: mov x28, x0
0x102e704f4 <-+148>: eor w8, w8, #0x1
0x102e704f8 <-+152>: sub w9, w11, #0x29 ; =0x29
0x102e704fc <-+156>: madd w22, w8, w9, w26
0x102e70500 <-+160>: mov x20, #0xa930
0x102e70504 <-+164>: movk x20, #0xea59, lsl #16
0x102e70508 <-+168>: movk x20, #0x9bdd, lsl #32
0x102e7050c <-+172>: movk x20, #0xd570, lsl #48
0x102e70510 <-+176>: add w8, w22, #0xb ; =0xb
0x102e70514 <-+180>: adr x24, #0x59acc
0x102e70518 <-+184>: nop
0x102e7051c <-+188>: ldr x8, [x24, w8, sxtw #3]
0x102e70520 <-+192>: sub x8, x8, #0x2 ; =0x2
(lldb)
```

disassemble语法

(lldb) **help** disassemble

Disassemble specified instructions in the current target. Defaults to the current function for the current thread and stack frame.

Syntax: disassemble [<cmd-options>]

Command Options Usage:

disassemble [-bmr] -s address-expression [-A arch] [-C <num-lines>] [-e address-expression] [-F disassembly-flavor] [-P <plugin>]

disassemble [-bmr] -s address-expression [-A arch] [-C <num-lines>] [-c num-lines] [-F <disassembly-flavor>] [-P <plugin>]

disassemble [-bmr] [-A arch] [-C <num-lines>] [-c <num-lines>] [-F disassembly-flavor] [-n function-name] [-P <plugin>]

disassemble [-bfmr] [-A arch] [-C <num-lines>] [-c num-lines] [-F disassembly-flavor] [-P <plugin>]

disassemble [-bmpr] [-A arch] [-C <num-lines>] [-c num-lines] [-F disassembly-flavor] [-P <plugin>]

disassemble [-blmr] [-A arch] [-C <num-lines>] [-F disassembly-flavor] [-P <plugin>]

disassemble [-bmr] [-a address-expression] [-A <arch>] [-C num-lines] [-c num-li

```
nes] [-F <disassembly-flavor>] [-P <plugin>]

--force
    Force disassembly of large functions.

-A <arch> ( --arch <arch> )
    Specify the architecture to use from cross disassembly.

-C <num-lines> ( --context <num-lines> )
    Number of context lines of source to show.

-F <disassembly-flavor> ( --flavor <disassembly-flavor> )
    Name of the disassembly flavor you want to use. Currently the only
    valid options are default, and for Intel architectures, att and
    intel.

-P <plugin> ( --plugin <plugin> )
    Name of the disassembler plugin you want to use.

-a <address-expression> ( --address <address-expression> )
    Disassemble function containing this address.

-b ( --bytes )
    Show opcode bytes when disassembling.

-c <num-lines> ( --count <num-lines> )
    Number of instructions to display.

-e <address-expression> ( --end-address <address-expression> )
    Address at which to end disassembling.

-f ( --frame )
    Disassemble from the start of the current frame's function.

-l ( --line )
    Disassemble the current frame's current source line instructions if
    there is debug line table information, else disassemble around the
    pc.

-m ( --mixed )
    Enable mixed source and assembly display.

-n <function-name> ( --name <function-name> )
    Disassemble entire contents of the given function name.

-p ( --pc )
    Disassemble around the current pc.

-r ( --raw )
    Print raw disassembly with no symbol information.

-s <address-expression> ( --start-address <address-expression> )
    Address at which to start disassembling.
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 21:36:17

thread

TODO:

【记录】lldb命令使用心得: thread

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

frame

TODO:

- 【记录】lldb命令使用心得: frame
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:27:22

breakpoint

TODO:

- 【记录】lldb命令使用心得: breakpoint
- 【已解决】Xcode中lldb中b list不是breakpoint list
- 【未解决】Xcode中无法给下一行将要运行的汇编指令加断点
- 【已解决】XCode中如何给符号断点加上判断条件
- 【未解决】XCode和lldb如何根据函数地址加断点

- 详见独立子教程
 - [iOS逆向之动态调试: 断点](#)

breakpoint举例

```
breakpoint set -a 函数地址
```

```
breakpoint set -a ASLR偏移量 + 静态分析的函数地址
```

```
breakpoint set -a (image list -o -f看到的库的加载的起始地址) + (IDA等工具) 静态分析的函数地址
```

```
(lldb) breakpoint set -a 0x1102d3348
Breakpoint 54: where = AwemeCore`___lldb_unnamed_symbol1462804$$AwemeCore + 480, address = 0x00000001102d3348
(lldb) breakpoint list
Current breakpoints:
...
54: address = AwemeCore[0x00000000ee2b348], locations = 1, resolved = 1, hit count = 0
    54.1: where = AwemeCore`___lldb_unnamed_symbol1462804$$AwemeCore + 480, address = 0x00000001102d3348, resolved, hit count = 0
```

```
breakpoint set --name foo --condition '(int)strcmp(y,"hello") == 0'
```

==

```
br s -n foo -c '(int)strcmp(y,"hello") == 0'
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-12-21 22:14:23

breakpoint的help语法

help breakpoint

```
(lldb) help breakpoint
Commands for operating on breakpoints (see 'help b' for shorthand.)
```

```
Syntax: breakpoint <subcommand> [ <command-options> ]
```

The following subcommands are supported:

```
clear    -- Delete or disable breakpoints matching the specified source
          file and line.
command  -- Commands for adding, removing and listing LLDB commands
          executed when a breakpoint is hit.
delete   -- Delete the specified breakpoint(s). If no breakpoints are
          specified, delete them all.
disable  -- Disable the specified breakpoint(s) without deleting them. If
          none are specified, disable all breakpoints.
enable   -- Enable the specified disabled breakpoint(s). If no breakpoints
          are specified, enable all of them.
list     -- List some or all breakpoints at configurable levels of detail.
modify   -- Modify the options on a breakpoint or set of breakpoints in
          the executable. If no breakpoint is specified, acts on the
          last created breakpoint. With the exception of -e, -d and -i,
          passing an empty argument clears the modification.
name     -- Commands to manage name tags for breakpoints
read     -- Read and set the breakpoints previously saved to a file with
          "breakpoint write"_.
set      -- Sets a breakpoint or set of breakpoints in the executable.
write    -- Write the breakpoints listed to a file that can be read in
          with "breakpoint read"_. If given no arguments, writes all
          breakpoints.
```

For more help on any particular subcommand, type 'help <command> <subcommand>'.

help breakpoint set

```
(lldb) help breakpoint set
Sets a breakpoint or set of breakpoints in the executable.
```

```
Syntax: breakpoint set <cmd-options>
```

Command Options Usage:

```
breakpoint set [-DHD] -l <linenum> [-G boolean] [-C command] [-c expr] [-i <count>] [-o boolean] [-q queue-name] [-t <thread-id>] [-x <thread-index>] [-T thread-name] [-R address] [-N breakpoint-name] [-u <column>] [-f filename] [-m boolean] [-s <shlib-name>] [-K boolean]
breakpoint set [-DHD] -a <address-expression> [-G boolean] [-C <command>] [-c <expr>] [-i <count>] [-o boolean] [-q <queue-name>] [-t <thread-id>] [-x <thread-index>] [-
```

```

T thread-name ] [-N breakpoint-name ] [-s shlib-name ]
  breakpoint set [-DHd] -n <function-name> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t thread-id ] [-x thread-index ] [-T thread-name ] [-R address ] [-N breakpoint-name ] [-f filename ] [-L source-language ] [-s shlib-name ] [-K boolean ]
  breakpoint set [-DHd] -F <fullname> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t <thread-id> ] [-x <thread-index> ] [-T thread-name ] [-R <address> ] [-N <breakpoint-name> ] [-f <filename> ] [-L <source-language> ] [-s <shlib-name> ] [-K <boolean> ]
  breakpoint set [-DHd] -S <selector> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t <thread-id> ] [-x <thread-index> ] [-T thread-name ] [-R <address> ] [-N <breakpoint-name> ] [-f <filename> ] [-L <source-language> ] [-s <shlib-name> ] [-K <boolean> ]
  breakpoint set [-DHd] -M <method> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t thread-id ] [-x thread-index ] [-T thread-name ] [-R address ] [-N breakpoint-name ] [-f filename ] [-L source-language ] [-s <shlib-name> ] [-K boolean ]
  breakpoint set [-DHd] -r <regular-expression> [-G boolean ] [-C <command> ] [-c <expr> ] [-i <count> ] [-o boolean ] [-q <queue-name> ] [-t <thread-id> ] [-x <thread-index> ] [-T thread-name ] [-R <address> ] [-N <breakpoint-name> ] [-f <filename> ] [-L <source-language> ] [-s <shlib-name> ] [-K <boolean> ]
  breakpoint set [-DHd] -b <function-name> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t thread-id ] [-x thread-index ] [-T thread-name ] [-R address ] [-N breakpoint-name ] [-f filename ] [-L source-language ] [-s <shlib-name> ] [-K boolean ]
  breakpoint set [-ADHd] -p <regular-expression> [-G boolean ] [-C <command> ] [-c <expr> ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t <thread-id> ] [-x thread-index ] [-T thread-name ] [-N <breakpoint-name> ] [-f <filename> ] [-m boolean ] [-s <shlib-name> ] [-X <function-name> ]
  breakpoint set [-DHd] -E <source-language> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t thread-id ] [-x thread-index ] [-T <thread-name> ] [-N <breakpoint-name> ] [-O <type-name> ] [-h boolean ] [-w boolean ]
  breakpoint set [-DHd] -P <python-class> [-k none ] [-v none ] [-G boolean ] [-C command ] [-c <expr> ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t thread-id ] [-x <thread-index> ] [-T <thread-name> ] [-N <breakpoint-name> ] [-f filename ] [-s shlib-name ]
  breakpoint set [-DHd] -y <linespec> [-G boolean ] [-C command ] [-c expr ] [-i count ] [-o boolean ] [-q <queue-name> ] [-t <thread-id> ] [-x <thread-index> ] [-T thread-name ] [-R address ] [-N <breakpoint-name> ] [-m boolean ] [-s <shlib-name> ] [-K boolean ]

```

-A (--all-files)

All files are searched for **source** pattern matches.

-C **command** (--command **command**)

A **command** to run when the breakpoint is hit, can be provided more than once, the commands will get run in order left to right.

-D (--dummy-breakpoints)

Act on Dummy breakpoints - i.e. breakpoints **set** before a **file** is provided, **which** prime new targets.

-E **source-language** (--language-exception **source-language**)

Set the breakpoint on exceptions thrown by the specified language (without options, on throw but not catch.)

- F** `fullname` (`--fullname` `fullname`)
Set the breakpoint by fully qualified `function` names. For C++ this means namespaces and all arguments, and for Objective-C this means a full `functionprototype` with class and selector. Can be repeated multiple `times` to `make` one breakpoint for multiple names.
- G** `boolean` (`--auto-continue` `boolean`)
The breakpoint will auto-continue after running its commands.
- H** (`--hardware`)
Require the breakpoint to use hardware breakpoints.
- K** `boolean` (`--skip-prologue` `boolean`)
Skip the prologue if the breakpoint is at the beginning of a function. If not `set` the `target.skip-prologue` setting is used.
- L** `source-language` (`--language` `source-language`)
Specifies the Language to use when interpreting the breakpoint's `expression` (note: currently only implemented for setting breakpoints on identifiers). If not `set` the `target.language` setting is used.
- M** `<method>` (`--method` `<method>`)
Set the breakpoint by C++ method names. Can be repeated multiple times to `make` one breakpoint for multiple methods.
- N** `<breakpoint-name>` (`--breakpoint-name` `<breakpoint-name>`)
Adds this to the list of names for this breakpoint.
- O** `<type-name>` (`--exception-typename` `<type-name>`)
The breakpoint will only stop if an exception Object of this type is thrown. Can be repeated multiple times to stop for multiple object types. If you just specify the type's base name it will match against that `type` in all modules, or you can specify the full `type` name including modules. Other submatches are not supported at present. Only supported for Swift at present.
- P** `python-class` (`--script-class` `python-class`)
The name of the class that will manage a scripted breakpoint.
- R** `address` (`--address-slide` `<address>`)
Add the specified offset to whatever address(es) the breakpoint resolves to. At present this applies the offset directly as given, and doesn't try to align it to instruction boundaries.
- S** `<selector>` (`--selector` `<selector>`)
Set the breakpoint by ObjC selector name. Can be repeated multiple times to `make` one breakpoint for multiple Selectors.
- T** `<thread-name>` (`--thread-name` `<thread-name>`)
The breakpoint stops only for the thread whose thread name matches this argument.
- X** `<function-name>` (`--source-regexp-function` `<function-name>`)
When used with `'-p'` limits the `source` regex to `source` contained in the `namedfunctions`. Can be repeated multiple times.

- a** `address-expression` (`--address` `address-expression`)
Set the breakpoint at the specified address. If the address maps uniquely to a particular binary, then the address will be converted to a `file` address, so that the breakpoint will track that binary+offset no matter where the binary eventually loads. Alternately, if you also specify the module - with the `-s` option - then the address will be treated as a `file` address in that module, and resolved accordingly. Again, this will allow lldb to track that offset on subsequent reloads. The module need not have been loaded at the `time` you specify this breakpoint, and will get resolved when the module is loaded.
- b** `function-name` (`--basename` `function-name`)
Set the breakpoint by `function` `basename` (C++ namespaces and arguments will be ignored). Can be repeated multiple `times` to `make` one breakpoint `for` multiple symbols.
- c** `expr` (`--condition` `expr`)
The breakpoint stops only if this condition expression evaluates to true.
- d** (`--disable`)
Disable the breakpoint.
- f** `filename` (`--file` `filename`)
Specifies the `source file` in which to `set` this breakpoint. Note, by default lldb only looks for files that are `#included` if they use the standard include `file` extensions. To `set` breakpoints on `.c/.cpp/.m/.mm` files that are `#included`, set `target.inline-breakpoint-strategy` to `always`.
- h** `boolean` (`--on-catch` `boolean`)
Set the breakpoint on exception catch.
- i** `count` (`--ignore-count` `count`)
Set the number of `times` this breakpoint is skipped before stopping.
- k** `none` (`--structured-data-key` `none`)
The key for a key/value pair passed to the implementation of a scripted breakpoint. Pairs can be specified `more` than once.
- l** `linenum` (`--line` `linenum`)
Specifies the line number on which to `set` this breakpoint.
- m** `boolean` (`--move-to-nearest-code` `boolean`)
Move breakpoints to nearest code. If not `set` the `target.move-to-nearest-code-setting` is used.
- n** `function-name` (`--name` `function-name`)
Set the breakpoint by `function` name. Can be repeated multiple `times` to `make` one breakpoint `for` multiple names
- o** `boolean` (`--one-shot` `boolean`)
The breakpoint is deleted the first `time` it stop causes a stop.

```
-p regular-expression ( --source-pattern-regexp regular-expression )
  Set the breakpoint by specifying a regular expression which is
  matched against the source text in a source file or files specified
  with the -f can be specified more than once. If no source files
  are specified, uses the current default source file . If you want
  to match against all source files, pass the --all-files option.

-q queue-name ( --queue-name <queue-name> )
  The breakpoint stops only for threads in the queue whose name is
  given by this argument.

-r regular-expression ( --func-regex <regular-expression> )
  Set the breakpoint by function name, evaluating a
  regular-expression to find the function name(s).

-s shlib-name ( --shlib shlib-name )
  Set the breakpoint only in this shared library. Can repeat this
  option multiple times to specify multiple shared libraries.

-t thread-id ( --thread-id <thread-id> )
  The breakpoint stops only for the thread whose TID matches this
  argument.

-u column ( --column <column> )
  Specifies the column number on which to set this breakpoint.

-v none ( --structured-data-value <none> )
  The value for the previous key in the pair passed to the
  implementation of a scripted breakpoint. Pairs can be specified
  more than once.

-w boolean ( --on-throw boolean )
  Set the breakpoint on exception throw.

-x thread-index ( --thread-index <thread-index> )
  The breakpoint stops only for the thread whose index matches this
  argument.

-y linespec ( --joint-specifier linespec )
  A specifier in the form filename:line[:column] for setting file &
  line breakpoints.
```


watchpoint

TODO:

- 【已解决】Xcode中lldb中如何给watchpoint加上条件判断过滤
- 【已解决】Xcode中lldb的条件watchpoint报错: error user expression indirection requires pointer operand long invalid
- 【已解决】Xcode的lldb中如何监控结构体变量值的变化
- 【未解决】研究YouTube逻辑: 监控NSArray的_allTrackRenderers值被改动
- 【未解决】YouTube的HAMPlayerInternal的playerLoop中监控_currentTime变量值变化
- 【未解决】iOS逆向: watchpoint添加失败报错: error Watchpoint creation failed sending gdb watchpoint packet failed

watchpoint举例

默认: type=write

```
(lldb) watchpoint set expr 0x000000011ceb5818
Watchpoint created: Watchpoint 1: addr = 0x11ceb5818 size = 8 state = enabled type = w
new value: 0
```

触发时打印:

```
Watchpoint 1 hit:
old value: 0
new value: 0
```

type=read

```
(lldb) watchpoint set expression -w read -- 0x10b121828
Watchpoint created: Watchpoint 1: addr = 0x10b121828 size = 8 state = enabled type = r
watchpoint spec = '0x10b121828'
new value: 4476770681
```

控制watchpoint

关闭所有:

```
(lldb) watchpoint disable
All watchpoints disabled. (4 watchpoints)
```

开启所有:

```
(lldb) watchpoint enable  
All watchpoints enabled. (4 watchpoints)
```

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-12-21 22:12:02

watchpoint的help语法

help watchpoint

```
(lldb) help watchpoint
Commands for operating on watchpoints.
Syntax: watchpoint <subcommand> [ <command-options> ]
The following subcommands are supported:
  command -- Commands for adding, removing and examining LLDB commands
            executed when the watchpoint is hit (watchpoint 'commands').
  delete  -- Delete the specified watchpoint(s). If no watchpoints are
            specified, delete them all.
  disable -- Disable the specified watchpoint(s) without removing it/them.
            If no watchpoints are specified, disable them all.
  enable  -- Enable the specified disabled watchpoint(s). If no watchpoints
            are specified, enable all of them.
  ignore  -- Set ignore count on the specified watchpoint(s). If no
            watchpoints are specified, set them all.
  list    -- List all watchpoints at configurable levels of detail.
  modify  -- Modify the options on a watchpoint or set of watchpoints in
            the executable. If no watchpoint is specified, act on the
            last created watchpoint. Passing an empty argument clears the
            modification.
  set     -- Commands for setting a watchpoint.
For more help on any particular subcommand, type 'help <command> <subcommand>'.

```

help watchpoint set

```
(lldb) help watchpoint set
Commands for setting a watchpoint.
Syntax: watchpoint set <subcommand> [ <subcommand-options> ]
The following subcommands are supported:
  expression -- Set a watchpoint on an address by supplying an expression.
                Use the '-i' option to specify the language of the
                expression. Use the '-w' option to specify the type of
                watchpoint and the '-s' option to specify the byte size to
                watch for. If no '-w' option is specified, it defaults to
                write. If no '-s' option is specified, it defaults to the
                target's pointer byte size. Note that there are limited
                hardware resources for watchpoints. If watchpoint setting
                fails, consider disable/delete existing ones to free up
                resources. Expects 'raw' input (see 'help raw-input'.)
  variable  -- Set a watchpoint on a variable. Use the '-w' option to
                specify the type of watchpoint and the '-s' option to
                specify the byte size to watch for. If no '-w' option is
                specified, it defaults to write. If no '-s' option is
                specified, it defaults to the variable's byte size. Note
                that there are limited hardware resources for watchpoints.
                If watchpoint setting fails, consider disable/delete
                existing ones to free up resources.

```

For more [help](#) on any particular subcommand, type `'help <command> <subcommand>'`.

help watchpoint set expression

```
(lldb) help watchpoint set expression
```

Set a watchpoint on an address by supplying an expression. Use the `'-l'` option to specify the language of the expression. Use the `'-w'` option to specify the [type](#) of watchpoint and the `'-s'` option to specify the byte size to watch for. If no `'-w'` option is specified, it defaults to write. If no `'-s'` option is specified, it defaults to the target's pointer byte size. Note that there are limited hardware resources for watchpoints. If watchpoint setting fails, consider disable/delete existing ones to free up resources. Expects 'raw' input (see `'help raw-input'`.)

Syntax: `watchpoint set expression <cmd-options> -- <expr>`

Command Options Usage:

```
watchpoint set expression [-w <watch-type>] [-s <byte-size>] [-l <source-language>] -  
- <expr>
```

```
watchpoint set expression <expr>
```

```
-l <source-language> ( --language <source-language> )
```

```
Language of expression to run
```

```
-s <byte-size> ( --size <byte-size> )
```

```
Number of bytes to use to watch a region.
```

```
Values: 1 | 2 | 4 | 8
```

```
-w <watch-type> ( --watch <watch-type> )
```

```
Specify the type of watching to perform.
```

```
Values: read | write | read_write
```

Examples:

```
watchpoint set expression -w write -s 1 -- foo + 32
```

```
Watches write access for the 1-byte region pointed to by the address 'foo +  
32'
```

This command takes options and free-form arguments. If your arguments resemble option specifiers (i.e., they start with a `-` or `--`), you must use `' -- '` between the end of the [command](#) options and the beginning of the arguments.

调试控制

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

run

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

continue

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

next

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

nexti

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

step

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

stepi

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

jump

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

finish

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

exit

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

LLDB心得

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

导出结果到文件

lldb调试期间，对于命令：

```
disassemble -f
```

的结果，想要输出/导出到文件，可以：

方法1: `session save`

- 方法1: `session save`
 - 优点：
 - 方便，无需引入额外命令
 - （对部分是优点）可以把session的所有内容都保存输出到文件
 - 缺点：没有针对性（针对自己想要的特定的代码的输出结果）

不借助外部脚本和命令，直接用：

```
session save /Users/crifan/dev/tmp/sub_1000A0460.txt
```

即可：

把当前会话session == 当前这次lldb的调试，自动启动开始，到现在的输出的内容

都输出到文件中

其中就包括，之前刚已运行的命令的输出结果了。

方法2: 用 `lldb` 的 (`python`) 脚本

- 方法2: 用 `lldb` 的 (`python`) 脚本
 - 缺点：要额外新引入lldb的脚本
 - 优点：可以有针对性的，只输出单个命令的结果到文件
 - 而不用保存当前session的所有内容

比如：

[4iar/lldb-write: Write the output of an lldb command to file \(github.com\)](https://github.com/4iar/lldb-write)

用法：

先安装：

下载代码

```
git clone https://github.com/4iar/lldb-write.git
```

然后去加到lldb启动脚本中：


```
``bash
vi ~/.lldbinit
```

加上:

```
command script import /{change\_to\_your\_path}/lldb-write/write.py
```

重启lldb后, 可以看到提示:

```
The "write" command has been loaded and is ready for use.
```

然后即可使用:

```
write /some/path/outputFile.txt yourOriginLldbCommand
```

比如:

```
write /Users/crifan/dev/tmp/lldb_akd_symbol12575_disassemble.txt dis -f
```

即可把当前lldb命令的输出结果, 导出到文件中。

同时: 当前lldb终端中, 仍能正常看到结果。还是很好用的。

crifan.org, 使用[署名4.0国际\(CC BY 4.0\)协议](#)发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 21:35:07

命令缩写

lldb命令的缩写：所有命令都支持任意前缀字符的缩写，只要不产生混淆

lldb中的命令，可以缩写

比如：

```
print
```

常见缩写是：

```
p
```

但其实底层逻辑是：

从第一个字母 `p` 到最后一个字母 `t`，缩写到任意位置都是可以的

前提是，只要不（和其他命令的前缀）产生混淆

举例

print

- `print`
 - `p`
 - 是 lldb 专门为 `print` 保留的 `p`，所以可以用 `p`
 - 否则按道理，也会和其他 `process` 等命令产生冲突，也不能把 `print` 缩写为 `p`
 - `pr`
 - 和 `process` 的 `pr` 是一样的前缀字符
 - lldb 无法确定是哪个，所以就属于会产生冲突、混淆
 - 所以不能用 `pr`
 - `pri`
 - 可以
 - `prin`
 - 可以
 - `print`
 - 可以 所以总体结论就是：
- `print`
 - 可以用特定的缩写：`p`
 - 也可以用其他普通的，不产生冲突的缩写：`pri`、`prin`、`print`

breakpoint

断点的命令

```
breakpoint
```

可以缩写/简写为：

```
breakpoint  
breakpoi  
breakpo  
breakp  
break  
brea  
bre  
br
```

而不能用：

- `b`
 - 特殊：属于lldb中专门保留的特定的缩写
 - 含义是：以某种特定的格式去添加断点

-> 有了上面的缩写逻辑，则普通的：

```
breakpoint list
```

就可以写为：

```
breakpoint list  
breakpoi list  
breakpo list  
breakp list  
break list  
break list  
brea list  
bre list  
br list
```

都是可以的，都是等价的

子命令也支持缩写

当然命令的子命令，参数，也是同样支持缩写

比如此处

```
br list
```

的 `list` 也可以缩写：

```
br lis  
br li  
br l
```

只要不产生冲突即可

此处就是 `breakpoint` 的子命令中，上述缩写不会冲突混淆即可。

注：

此处可以用 `help breakpoint` 去查看，`breakpoint` 有哪些子命令

```
(lldb) help breakpoint
Commands for operating on breakpoints (see 'help b' for shorthand.)

Syntax: breakpoint -subcommand- [command-options-]

The following subcommands are supported:

clear    -- Delete or disable breakpoints matching the specified source
          file and line.
command  -- Commands for adding, removing and listing LLDB commands
          executed when a breakpoint is hit.
delete   -- Delete the specified breakpoint(s). If no breakpoints are
          specified, delete them all.
disable  -- Disable the specified breakpoint(s) without deleting them. If
          none are specified, disable all breakpoints.
enable   -- Enable the specified disabled breakpoint(s). If no breakpoints
          are specified, enable all of them.
list     -- List some or all breakpoints at configurable levels of detail.
modify   -- Modify the options on a breakpoint or set of breakpoints in
          the executable. If no breakpoint is specified, acts on the
          last created breakpoint. With the exception of -e, -d and -i,
          passing an empty argument clears the modification.
name     -- Commands to manage name tags for breakpoints
read     -- Read and set the breakpoints previously saved to a file with
          "breakpoint write"_.
set      -- Sets a breakpoint or set of breakpoints in the executable.
write    -- Write the breakpoints listed to a file that can be read in
          with "breakpoint read"_.

For more help on any particular subcommand, type 'help <command> <subcommand>'_.
```

其中可见，`breakpoint` 的子命令：

- `clear`
- `command`
- `delete`
- `disable`
- `enable`
- `modify`
- `name`
- `read`
- `set`
- `write`

不会和上面的缩写 `lis`、`li`、`l`，有冲突和混淆

-> 所以你会看到，很多人常把：

```
breakpoint list
```

写成：

```
br l
```

就是这个目的：

- 尽量用缩写
 - -> 减少输入的字符数
 - -> 提高调试效率

其他常见缩写

其他常用缩写：

- `expression` -> `e`、`exp`
- `breakpoint` = `br` == 特殊写法: `b`
 - `breakpoint set` -> `br s`
 - `breakpoint list` -> `br l`
 - 举例
 - `breakpoint delete 1` -> `br del 1`
 - `breakpoint disable 1` -> `br dis 1`
 - `breakpoint enable 1` -> `br en 1`
 - `breakpoint set --name main` -> `br s -n main` -> `b main`
 - `breakpoint set --name foo --condition '(int)strcmp(y,"hello") == 0'`
 - -> `br s -n foo -c '(int)strcmp(y,"hello") == 0'`
- `disassemble` -> `dis`
 - 举例
 - `disassemble -s 0x1167b1974` -> `dis -s 0x1167b1974`
- `register` -> `reg`
 - `register read` -> `reg r`
 - 举例
 - `register r x0` -> `reg r x0`
- `image` -> `im`
 - `image lookup` = `im loo`
 - 举例
 - `image lookup --address 0x1ec4` -> `im loo -a 0x1ec4`
- `memory` -> `mem`

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:31:37

Xcode中lldb

TODO:

- **【整理】** Xcode的lldb调试心得：F7单步进入无名的汇编代码
- **【未解决】** XCode和lldb如何根据函数地址加断点
- **【已解决】** XCode和lldb调试常见用法和调试心得
- **【已解决】** XCode的lldb中如何调试找到当前函数_dyld_get_image_name的返回值
- **【已解决】** Xcode调试：lldb中临时变量

此处整理 xcode 中的 lldb 的一些心得：

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:50:34

支持自动补全

Xcode 中 lldb 中支持自动补全：

- 输入 `regi` , 自动补全列出: `register`

◦

- 输入 `disa` 自动补全出: `disassemble`

◦

- `register read x` 后的自动补全效果

-

-
- image 1 后的自动补全效果

-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:50:23

查看函数调用堆栈

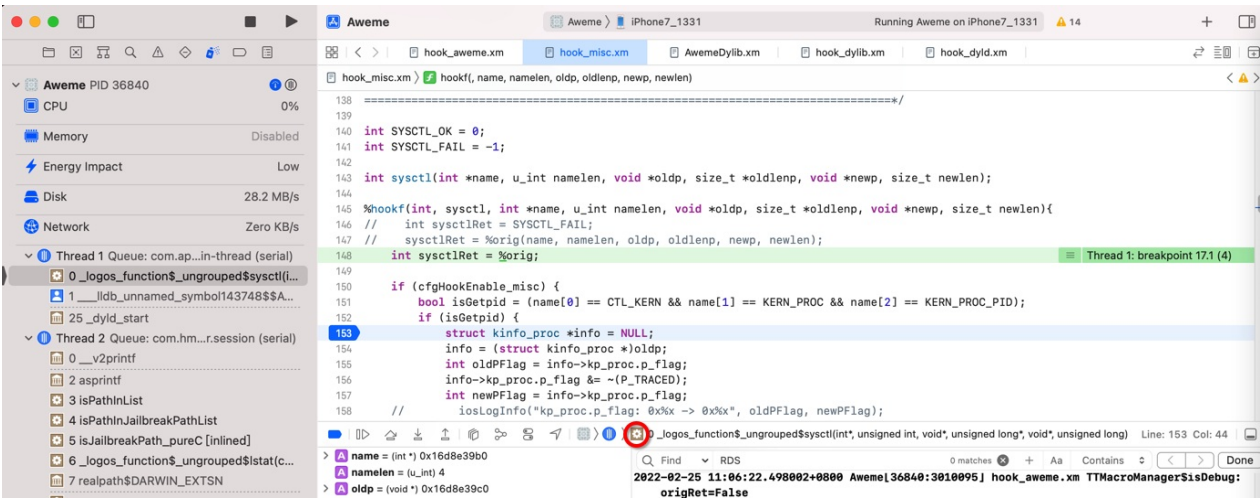
lldb 和 xCode 中查看 函数调用堆栈 = backtrace :

XCode调试期间, 想要查看: 函数调用堆栈

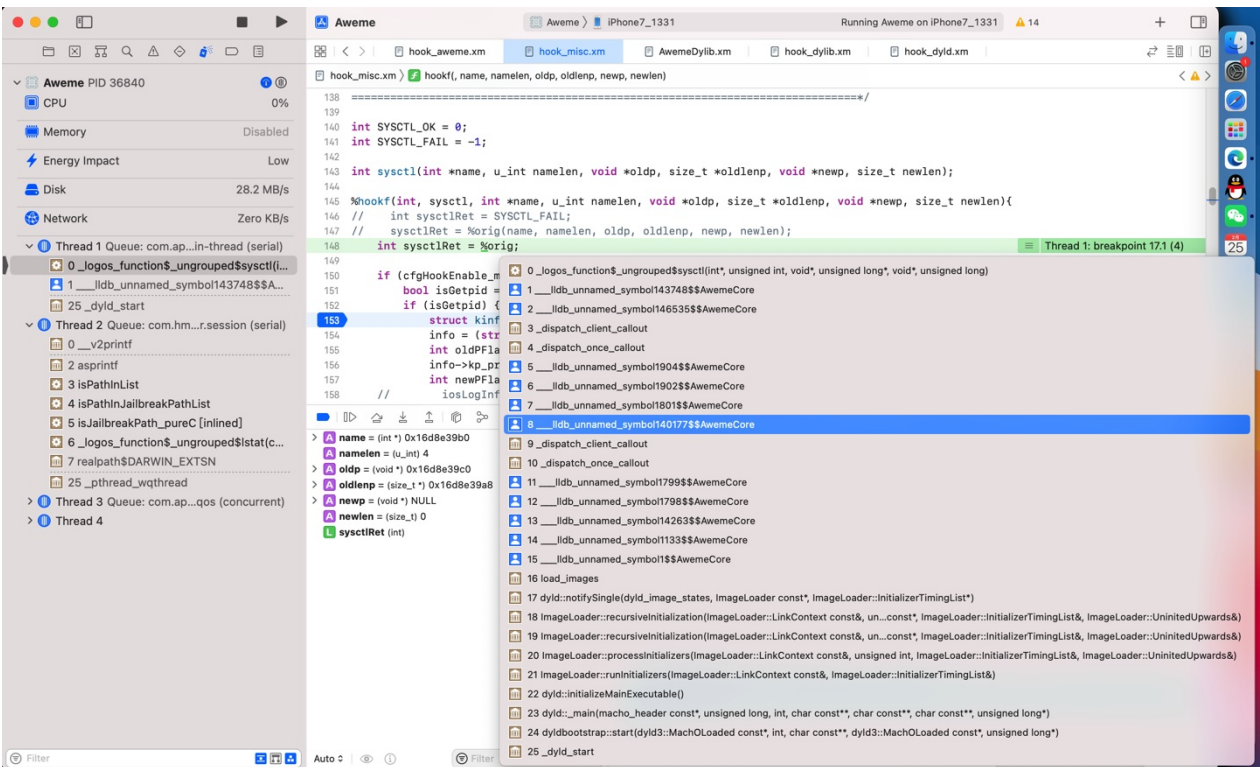
至少有2种方法:

XCode的UI界面中

XCode 中, Command + 鼠标单击 :



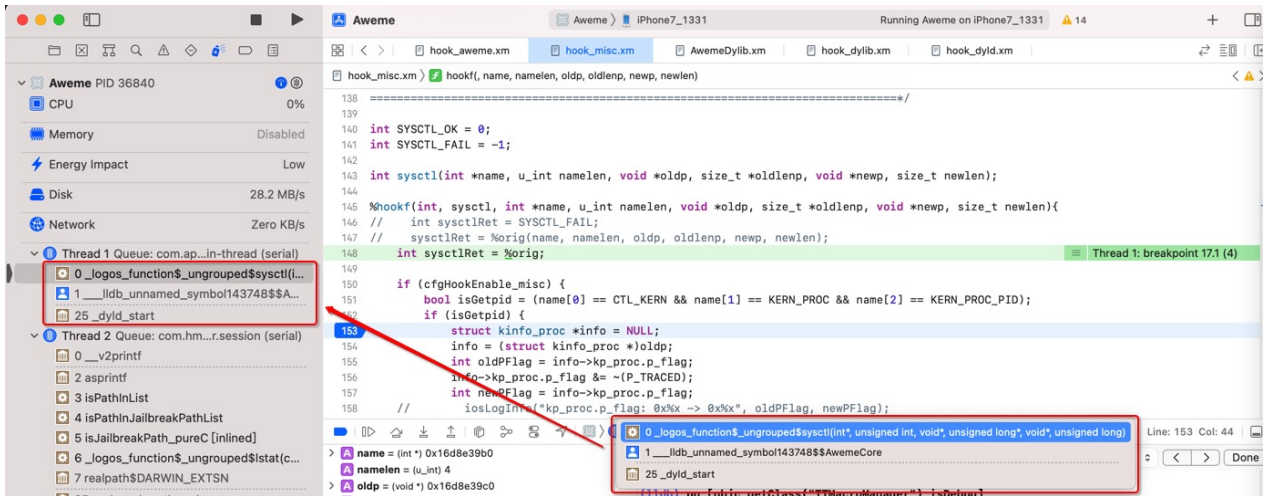
即可看到全部的函数调用堆栈:



注:

直接 鼠标点击 （不加 Command 键），则只显示缩略后的信息：

且和 Debug Navigator 中的线程下面的 函数调用堆栈 简略信息 是一致的：



lldb命令bt

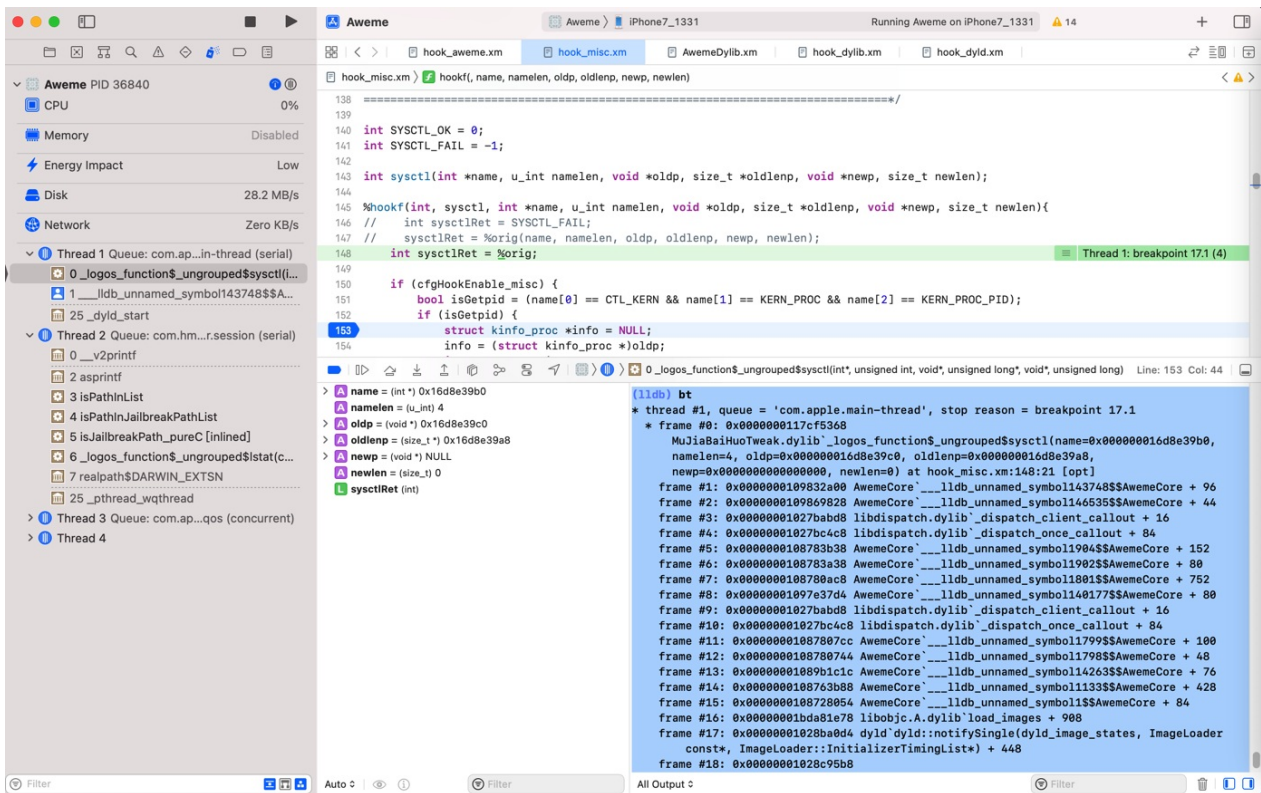
- bt = thread backtrace
 - = th b
 - = th ba

举例：

```
(lldb) bt
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 17.1
  frame #0: 0x0000000117cf5368 XxxTweak.dylib`_logos_function$ungrouped$sysctl(name=
0x000000016d8e39b0, namelen 4, oldp 0x000000016d8e39c0, oldlenp 0x000000016d8e39a8, newp
=0x0000000000000000, newlen 0) at hook_misc.xm:148:21 [opt]
  frame #1: 0x0000000109832a00 AwemeCore`__lldb_unnamed_symbol143748$$AwemeCore + 96
  frame #2: 0x0000000109869828 AwemeCore`__lldb_unnamed_symbol146535$$AwemeCore + 44
  frame #3: 0x00000001027babd8 libdispatch.dylib`_dispatch_client_callout + 16
  frame #4: 0x00000001027bc4c8 libdispatch.dylib`_dispatch_once_callout + 84
  frame #5: 0x0000000108783b38 AwemeCore`__lldb_unnamed_symbol11904$$AwemeCore + 152
  frame #6: 0x0000000108783a38 AwemeCore`__lldb_unnamed_symbol11902$$AwemeCore + 80
  frame #7: 0x0000000108780ac8 AwemeCore`__lldb_unnamed_symbol11801$$AwemeCore + 752
  frame #8: 0x00000001097e37d4 AwemeCore`__lldb_unnamed_symbol1140177$$AwemeCore + 80
  frame #9: 0x00000001027babd8 libdispatch.dylib`_dispatch_client_callout + 16
  frame #10: 0x00000001027bc4c8 libdispatch.dylib`_dispatch_once_callout + 84
  frame #11: 0x00000001087807cc AwemeCore`__lldb_unnamed_symbol11799$$AwemeCore + 100
  frame #12: 0x0000000108780744 AwemeCore`__lldb_unnamed_symbol11798$$AwemeCore + 48
  frame #13: 0x00000001089b1c1c AwemeCore`__lldb_unnamed_symbol114263$$AwemeCore + 76
  frame #14: 0x0000000108763b88 AwemeCore`__lldb_unnamed_symbol11133$$AwemeCore + 428
  frame #15: 0x0000000108728054 AwemeCore`__lldb_unnamed_symbol11$$AwemeCore + 84
  frame #16: 0x00000001bda81e78 libobjc.A.dylib`load_images + 908
  frame #17: 0x00000001028ba0d4 dyld`dyld::notifySingle(dyld_image_states, ImageLoader
r const*, ImageLoader::InitializerTimingList) + 448
  frame #18: 0x00000001028c95b8 dyld`ImageLoader::recursiveInitialization(ImageLoader
::LinkContext const*, unsigned int, char const*, ImageLoader::InitializerTimingList, I
mageLoader::UninitedUpwards) + 524
  frame #19: 0x00000001028c953c dyld`ImageLoader::recursiveInitialization(ImageLoader
```

```

LinkContext const&, unsigned int, char const*, ImageLoader::InitializerTimingList, ImageLoader::UninitiatedUpwards) + 400
    frame #20: 0x00000001028c8334 dyld`ImageLoader::processInitializers(ImageLoader::LinkContext const&, unsigned int, ImageLoader::InitializerTimingList, ImageLoader::UninitiatedUpwards) + 184
    frame #21: 0x00000001028c83fc dyld`ImageLoader::runInitializers(ImageLoader::LinkContext const&, ImageLoader::InitializerTimingList) + 92
    frame #22: 0x00000001028ba420 dyld`dyld::initializeMainExecutable() + 216
    frame #23: 0x00000001028bedb4 dyld`dyld::_main(macho_header const*, unsigned long, int, char const**, char const**, char const**, unsigned long*) + 4616
    frame #24: 0x00000001028b9208 dyld`dyldbootstrap::start(dyld3::MachOLoaded const*, int, char const**, dyld3::MachOLoaded const*, unsigned long*) + 396
    frame #25: 0x00000001028b9038 dyld`_dyld_start + 56
    
```



crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新: 2023-10-25 22:50:50

iOS逆向

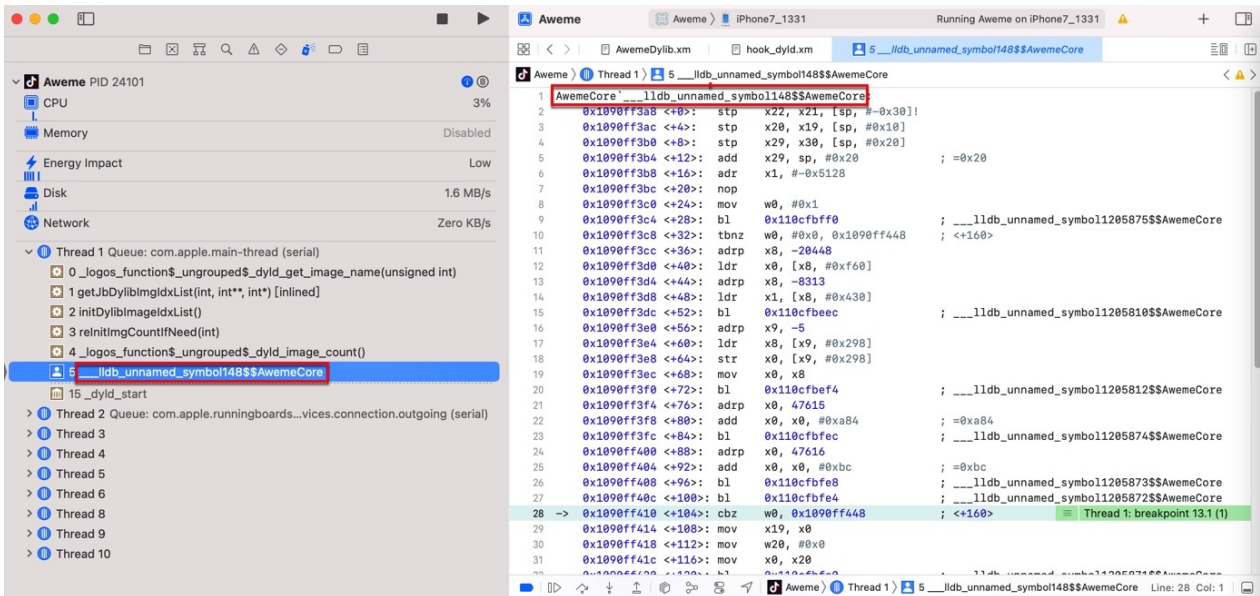
TODO:

- 整理常用的命令和举例
 - image
 - po
 - bt
 - reg
 - 等
- iOS逆向时用LLDB调试iOS中ObjC的对象和相关内容
- 【记录】iOS逆向Xcode调试心得：bl后cmn再b.eq很像是switch case或if else的代码逻辑跳转
- 【未解决】Xcode的lldb调试iOS的ObjC或Swift时如何打印出objc_msgSend第一个参数是什么类的实例
- 【已解决】Xcode的lldb中如何访问类的实例的内部属性值
- 【未解决】Xcode的lldb的po中如何判断对象是否是某个类的实例
- 【已解决】Xcode的lldb中如何调试运行iOS的ObjC代码

无名函数

iOS逆向期间，往往可以看到这种函数名：

AwemeCore`___lldb_unnamed_symbol148\$\$AwemeCore



其实就是个： 无名函数

完整的解释是：

- AwemeCore`___lldb_unnamed_symbol148\$\$AwemeCore`
 - AwemeCore：函数所属的（哪个）二进制

- 注：理论上，同一个函数，可能会出现在多个二进制中
- `___lldb_unnamed_symbol1148$$AwemeCore`
 - 无名函数
 - `___lldb_unnamed_symbol1148`：函数名的部分
 - `AwemeCore`：二进制的名字

-> 由此可以总结出：

- lldb中的无名函数的命名规则
 - `___lldb_unnamed_symbolNNN$$BinaryName`
 - `___lldb_unnamed_symbol1148$$AwemeCore`
 - `NNN = 148`
 - 从1开始编号
 - `BinaryName = AwemeCore`
 - 对应着当前lldb正在调试的二进制是 `AwemeCore`

->

- 知道这个能干什么？
 - 后续去给某个无名函数去加断点时，要注意把函数名写完整了，不要漏写成：
 - `___lldb_unnamed_symbol1148`
 - 否则是无法触发断点的
 - 要写成完整的函数名：
 - `___lldb_unnamed_symbol1148$$AwemeCore`
 - 才能正常触发断点

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

chisel

TODO:

- 【记录】用chisel调试iOS的app用法和心得
 - 【未解决】YouTube的HAMPlayerInternal的playerLoop中监控_currentTime变量值变化
-

- chisel
 - 是什么: llldb 的一个插件
 - 用途: 主要用于iOS逆向期间辅助调试
 - 主页
 - <https://github.com/facebook/chisel>
 - facebook/chisel: Chisel is a collection of LLDB commands to assist debugging iOS apps.

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

LLVM

此处也顺带去整理 LLDB 所属的开源项目 LLVM 的相关内容：

- LLVM
 - = Low Level Virtual Machine
 - 是什么=一句话描述
 - 一套用于构建出高度优化的编译器、优化器、运行环境的工具集合的开源项目
 - a toolkit for the construction of highly optimized compilers, optimizers, and runtime environments.
 - 主要包含3个部分
 - LLVM套件 = LLVM Suite
 - 包含各种
 - 工具
 - 汇编器 = assembler
 - 反汇编器 = disassembler
 - 位码分析器 = bitcode analyzer
 - 位码优化器 = bitcode optimizer
 - 简单的回归测试
 - 用于测试LLVM工具和Clang前端
 - 库
 - 头文件
 - Clang = Clang前端 = Clang front end
 - 是什么：LLVM的内置的原生的 C / C++ / Objective-C 编译器
 - 可以把 C , C++ , Objective-C 和 Objective-C++ 的代码，编译成 LLVM bitcode
 - 然后就可以用LLVM套件去操作此（编译后的）程序了
 - 测试套件 = Test Suite
 - 一堆工具的集合
 - 测试LLVM的功能和性能
 - 子项目
 - LLVM Core libraries
 - a modern source- and target-independent optimizer, along with code generation support for many popular CPUs
 - Clang
 - an LLVM native C/C++/Objective-C compiler
 - LLDB
 - a great native debugger
 - 基于 LLVM 和 Clang
 - libc++ 和 libc++ ABI
 - a standard conformant and high-performance implementation of the C++ Standard Library
 - including full support for C++11 and C++14
 - compiler-rt
 - provides highly tuned implementations of the low-level code generator

- MLIR
 - a novel approach to building reusable and extensible compiler infrastructure
- OpenMP
 - an OpenMP runtime for use with the OpenMP implementation in Clang
- polly
 - a suite of cache-locality optimizations as well as auto-parallelism and vectorization using a polyhedral model
- libclc
 - implement the OpenCL standard library
- klee
 - implements a "symbolic virtual machine" which uses a theorem prover to try to evaluate all dynamic paths through a program in an effort to find bugs and to prove properties of functions
- LLD
 - a new linker
 - a drop-in replacement for system linkers and runs much faster
- 资料
 - 官网
 - The LLVM Compiler Infrastructure Project
 - <https://llvm.org>
 - 快速上手
 - Getting Started with the LLVM System — LLVM 12 documentation
 - <https://llvm.org/docs/GettingStarted.html>
- 相关
 - 概念
 - IR = Intermediate Representation = 中间表示层

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

附录

下面列出相关参考资料。

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

文档

- 官网
 - LLDB Homepage — The LLDB Debugger
 - <http://lldb.llvm.org>
- 教程
 - Tutorial — The LLDB Debugger
 - <https://lldb.llvm.org/use/tutorial.html>
- LLDB和GDB命令对比
 - GDB to LLDB command map — The LLDB Debugger
 - <https://lldb.llvm.org/use/map.html>
 - 背景：由于GDB使用更广泛，所以LLDB为了让从GDB转过来的人，更快上手，而整理了GDB命令到LLDB命令的映射的文档，介绍的很详细，值得参考

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-07-13 17:28:19

参考资料

- 【记录】lldb命令使用心得: image
- 【已解决】lldb命令使用心得: image lookup
- 【记录】lldb命令使用心得: image lookup
- 【已解决】XCode的lldb中把image list输出结果重定向保存到本地文件
- 【已解决】lldb中image中搜索symbol函数名
- 【未解决】研究抖音越狱逻辑: lldb中image查询stafps的函数实现有几个
- 【已解决】iOS逆向: lldb调试时image lookup限定在某个二进制库内搜索
-
- 【已解决】XCode和lldb调试常见用法和调试心得
- 【记录】iOS逆向: lldb中memory和memory read命令的语法help
- 【已解决】lldb子命令: 反汇编disassemble
- 【已解决】debugserver+lldb调试arm汇编时输出当前函数的反汇编代码
- 【已解决】lldb调试如何显示整个函数的汇编代码
- 【已解决】lldb中把命令输出结果导出到文件
- 【整理Book】iOS逆向开发: dyld动态库链接器
- 【未解决】iOS逆向Preferences: 疑似下一步按钮处理函数_handlePressGesture:
- 【未解决】iOS逆向Apple账号: -[AAUISignInViewController _nextButtonSelected:]
- 【未解决】iOS逆向Apple账号: -[AAUISignInController
_performAuthenticationForAccount:serviceType:inViewController:completion:]
- 【未解决】iOS逆向Apple账号: -[AAUISignInController authenticationContext]
- 【未解决】iOS逆向MSFindSymbol报错: dyld missing symbol called
- 【未解决】iOS逆向Apple账号: 研究+[AADeviceInfo init]看是否能找到udid生成的逻辑
- 【已解决】iOS逆向: Xcode中给lldb无名函数加断点指定属于哪个库文件
- 【已解决】Unicorn模拟arm64: PC在+4404时报错UC_ERR_MAP
- 【未解决】iOS逆向Apple账号: +[UMUserManager sharedManager]
- 【基本解决】Xcode的lldb中动态调试objc_msgSend第一个参数self是哪个类
- 【未解决】iOS逆向AppleStore: -[ACAccountStore accountsWithAccountType:options:error:]的block函数
- 【未解决】iOS逆向AppleStore: -[__NSXPCInterfaceProxy_ACRemoteAccountStoreProtocol
accountTypeWithIdentifier:handler:]
- 【未解决】iOS逆向Apple账号: Xcode调试找不到和没有触发NSURLRequest相关断点
- 【未解决】iOS逆向Apple账号: CFURLRequestSetRequestPriority
- 【已解决】iOS逆向Apple账号: 哪个类实现了函数didReceiveData等去获取response的data
- 【规避解决】lldb中给ObjC函数加断点报错: WARNING Unable to resolve breakpoint to any actual
locations
- 【记录】iOS逆向Apple账号: objc_alloc_init当是类AADeviceInfo时Xcode调试和hook代码两者函数调用堆栈不同
- 【未解决】iOS逆向Apple账号: debugserver+lldb调试+[AADeviceInfo uidid]函数逻辑
- 【已解决】iOS逆向Apple账号: 类AADeviceInfo相关
- 【未解决】iOS逆向: 通过查看NSXPCConnection的属性值搞清楚目标处理的类是哪个
- 【已解决】iOS逆向: Frida如何hook拦截akd中某个lldb无名函数
- 【未解决】iOS逆向Apple账号: 用debugserver+lldb去调试-[AALoginAccountRequest urlRequest]看

断点是否触发

- 【未解决】iOS逆向Apple账号：-[AKAppleIDAuthenticationContextManager shouldContinueWithAuthenticationResults:error:forContextID:completion:]
- 【未解决】iOS逆向Apple账号：debugserver+lldb调试+[AADeviceInfo uid]函数逻辑
- 【已解决】iOS逆向Apple账号：+[AADeviceInfo uid]断点没生效换调试对象Preferences
- 【已解决】Xcode中lldb调试遇到Block类型变量**NSMallocBlock**
- 【已解决】研究抖音关注逻辑：__lldb_unnamed_symbol1462804的<+456>的bl
- 【已解决】研究抖音关注逻辑：__lldb_unnamed_symbol1189326\$\$AwemeCore
- 【已解决】研究抖音越狱逻辑：获取网络请求返回的响应response
- 【已解决】研究抖音关注逻辑：__lldb_unnamed_symbol1588524\$\$AwemeCore
- 【已解决】研究抖音关注逻辑：__lldb_unnamed_symbol1588526\$\$AwemeCore
- 【记录】研究YouTube逻辑：__lldb_unnamed_symbol22084\$\$YouTube
- 【已解决】研究YouTube逻辑：ctier=A时的 request和response的函数调用顺序
- 【未解决】研究YouTube逻辑：YTPlayerViewController的 updateActiveOverlayWithActiveVideoCurrentState中如何获取currentVideoTime并重置当前播放时间
- 【未解决】研究YouTube广告重置时间逻辑：MLHAMQueuePlayerSegmentList的 updatePeriodCurrentTimeForSegment的block
- 【未解决】研究YouTube逻辑：HAMPlayerInternal的pause
- 【未解决】iOS逆向：libsubstrate.dylib中找不到_dyld_get_all_image_infos而报错
- 【未解决】YouTube广告重置时间：MLHAMQueuePlayer的failWithError
- 【未解决】iOS的app中的内存：**DATA_DIRTY**的objc_data
- 【未解决】iOS逆向Apple账号：-[AAUISignInViewController _attemptAuthenticationWithContext:]
- 【未解决】用debugserver+lldb去调试寻找间接跳转objc_msgSend的逻辑
- 【已解决】研究抖音关注逻辑：__lldb_unnamed_symbol1653310\$\$AwemeCore
- 【已解决】研究抖音关注逻辑：__lldb_unnamed_symbol1462862\$\$AwemeCore
- 【已解决】抖音AwemeCore恢复符号表后导致部分函数显示错乱
- 【已解决】ARM汇编指令：PAC相关指令xpaci
- 【未解决】iOS逆向Apple账号：给iOS 15.1的CFNetwork恢复符号表
- 【未解决】debugserver+lldb调试__lldb_unnamed_symbol2567\$\$akd函数的真实逻辑
- 【未解决】iOS逆向Apple：arm64的-[AKAppleIDAuthenticationController authenticateWithContext:completion:]
- 【未解决】iOS逆向Apple账号：-[AALoginAccountRequest urlRequest]
- 【已解决】iOS逆向Apple账号：换tweak插件去调试+[AADeviceInfo uid]
- 【已解决】iOS逆向时lldb中添加了ObjC函数的断点却没触发到
- 【记录】研究YouTube逻辑：HAMPlayerInternal的setStatus的block
- 【已解决】iOS逆向akd：找arm64的akd函数sub_1000A0460的断点地址
- 【未解决】iOS逆向AppleStore：__lldb_unnamed_symbol3027\$\$AppleStoreCore
- 【未解决】iOS逆向AppleStore证书无效：多个SSL函数的断点都没触发到
- 【未解决】Xcode中如何给iOS的Swift函数加断点：AppleStoreCore的User的initialize
- 【已解决】iOS逆向Apple账号：哪个类实现了函数didReceiveData等去获取response的data
- 【未解决】Xcode的lldb中动态调试objc_msgSend第一个参数self是哪个类
-
- [lldb调试器知多少 - 掘金 \(juejin.cn\)](#)
- [LLDB调试器使用简介 | 南峰子的技术博客 \(southpeak.github.io\)](#)
- [ObjC 中国 - 与调试器共舞 - LLDB 的华尔兹 \(objccn.io\)](#)
- [GDB to LLDB command map — The LLDB Debugger](#)

- [LLDB 调试命令使用指南 - 链滴 \(ld246.com\)](#)
- [LLDB Homepage — The LLDB Debugger \(llvm.org\)](#)
- [Tutorial — The LLDB Debugger \(llvm.org\)](#)
- [LLDB \(debugger\) - Wikipedia](#)
- [Dancing in the Debugger — A Waltz with LLDB · objc.io](#)
- [lldb cheat sheet](#)
- [4iar/lldb-write: Write the output of an lldb command to file \(github.com\)](#)
- [ios - How does the `-n` option to `image lookup` in LLDB operate compared to the `-s` option? - Stack Overflow](#)
- [debugging - iOS lldb function lookup - Stack Overflow](#)
- [lldb常用命令与调试技巧_ios_iOSer_InfoQ写作社区](#)
- [ios - What is `__lldb_unnamed_symbol?` - Stack Overflow](#)
- [Symbolication — The LLDB Debugger \(llvm.org\)](#)
- [iOS开发之LLDB常用命令 - 简书 \(jianshu.com\)](#)
- [Xcode lldb调试x命令 - 简书 \(jianshu.com\)](#)
- [LLDB调试命令 - 简书 \(jianshu.com\)](#)
- [iOS_Reverse/iOS逆向（三）：强大的断点调试工具.md at master · OPTJoker/iOS_Reverse \(github.com\)](#)
- [iOS 逆向指南：动态分析 - 掘金 \(juejin.cn\)](#)
- [deresz/funcap: IDA Pro script to add some useful runtime info to static analysis \(github.com\)](#)
- [iOS逆向--LLDB调试 - 掘金 \(juejin.cn\)](#)
-

crifan.org, 使用署名4.0国际(CC BY 4.0)协议发布 all right reserved, powered by Gitbook最后更新:
2023-10-25 22:41:04