

Отладка без исходников

Дмитрий Куркин
Adviqo

Для чего это нужно?



Чтобы видеть больше



О чём пойдет речь

- Когда может пригодиться отладка без исходников
- Примеры практического применения
 - nextKeyResponder
 - UIButton
 - UIViewController.loadView
- Полезные приемы

Кому это нужно?

- Для прикладного разработчика
- Без глубокого владения ассемблером
- Понятно не все, но значительно больше, чем ничего

Библиотеки без исходников

Mapkit

UIKit

GoogleAnalytics

Fabric

AppsFlyer

Необычное поведение

- Иногда совсем не работает
- Падает
- Выполняет ключевые действия в произвольном порядке
- Выдает лишние нотификации

На Stackoverflow никто толком не знает

- Совсем ничего нет
- Одни только вопросы
- Много шаманства с нестабильным результатом

Документация не всегда может помочь

- Документации слишком много. Знать все не реально, а найти не получается
- Документации мало

Служба поддержки это долго

- Заведите Radar, мы разберемся. Когда нибудь. Возможно.
- Ваш звонок очень важен для нас.

Тут хорошо бы подебажить

The screenshot shows the Xcode interface during a debugger session. The top status bar indicates "Running Investigation on iPhone X". The left sidebar displays system resource usage (CPU, Memory, Disk, Network) and a list of threads. Thread 1 is selected, showing its stack trace and assembly code.

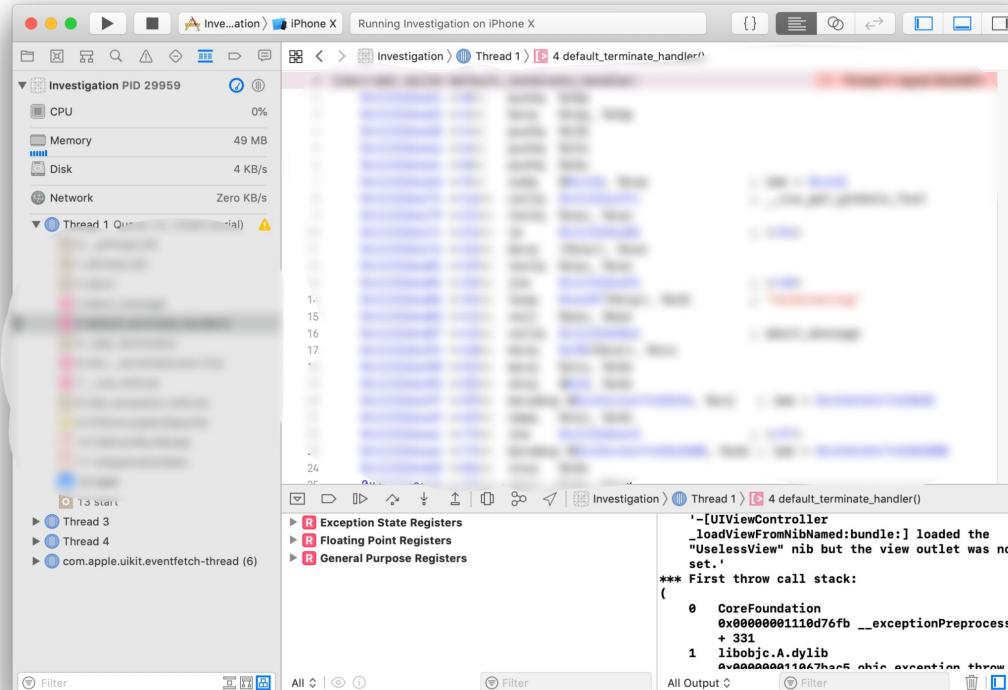
Assembly Code:

```
1 libc++abi.dylib`default_terminate_handler:
 2 0x1132dea64 <+0>: pushq %rbp
 3 0x1132dea65 <+1>: movq %rsp, %rbp
 4 0x1132dea68 <+4>: pushq %r15
 5 0x1132dea6a <+6>: pushq %r14
 6 0x1132dea6c <+8>: pushq %rbx
 7 0x1132dea6d <+9>: subq $0x18, %rsp ; imm = 0x418
 8 0x1132dea74 <+10>: calq 0x1132ea2fc ; __exa_get_globals_fast
 9 0x1132dea79 <+21>: testq %rax, %rax
10 0x1132dea7f <+24>: je 0x1132dea86 ; <+34>
11 0x1132dea7e <+26>: movq (%rax), %rax
12 0x1132dea81 <+29>: testq %rax, %rax
13 0x1132dea84 <+32>: jne 0x1132dea94 ; <+48>
14 0x1132dea88 <+34>: leaq 0xec07(%rip), %rdi ; "terminating"
15 0x1132dea8b <+41>: xorl %eax, %eax
16 0x1132dea8f <+43>: callq 0x1132de8ea ; abort_message
17 0x1132dea94 <+48>: movq 0x58(%rax), %rcx
18 0x1132dea98 <+52>: movq %rcx, %rdx
19 0x1132dea99 <+55>: shr $0x8, %rdx
20 0x1132dea9f <+59>: movabsq $0x434c4e47432b2b, %rsi ; imm = 0x434C4E47432B2B
21 0x1132dea9a <+69>: cmpq %rsi, %rdx
22 0x1132deaa4 <+72>: jne 0x1132deac5 ; <+97>
23 0x1132deaa8 <+74>: movabsq $0x434c4e47432b2b00, %rdx ; imm = 0x434C4E47432B2B00
24 0x1132deab8 <+84>: incq %rdx
```

Crash Log:

```
'-[UIViewController _loadViewFromNibNamed:bundle:] loaded the "UselessView" nib but the view outlet was not set.'  
*** First throw call stack:  
(  
 0 CoreFoundation  
 0x00000001110d76fb __exceptionPreprocess  
 + 331  
 1 libobjc.A.dylib  
 0x0000000110A79e5f objc_exception_throw +
```

Но там только “goobliegoo”



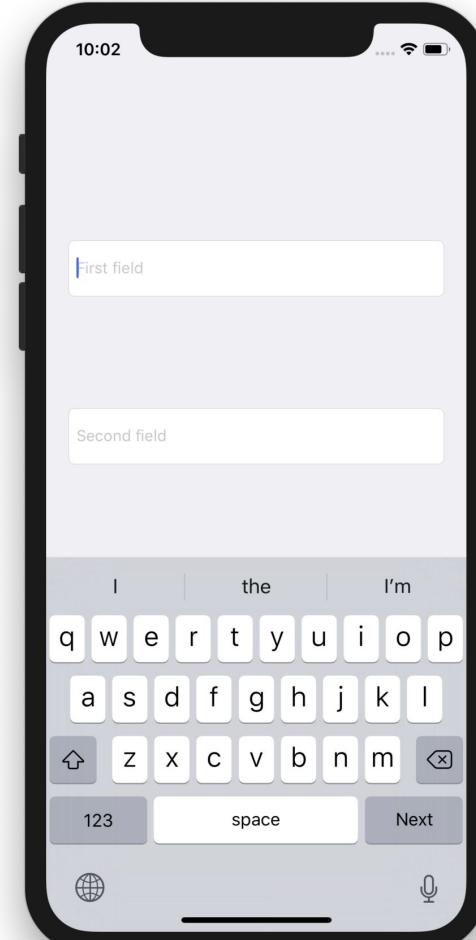
Он в этом разбирается!



nextKeyResponder

Описание проблемы

- ‘Tab’ переводит к следующему полю
- Как получить тоже самое для ‘Next’?

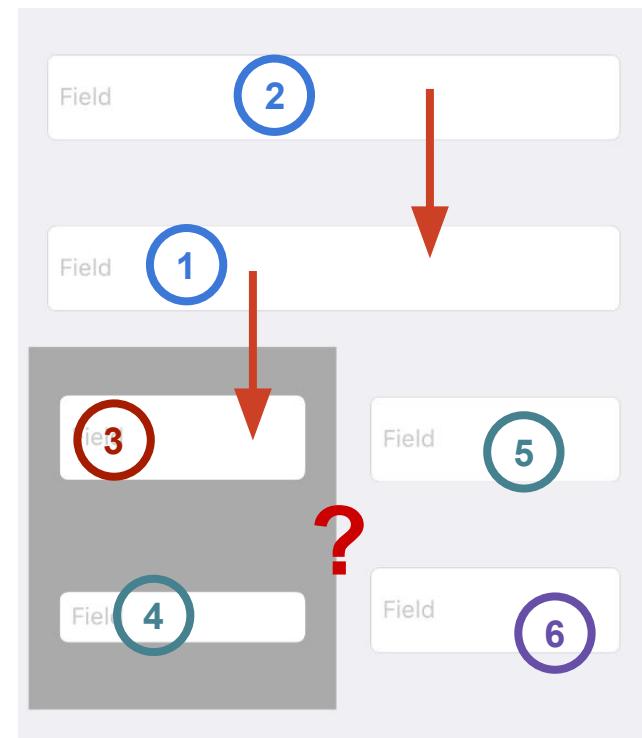


Решения из интернета

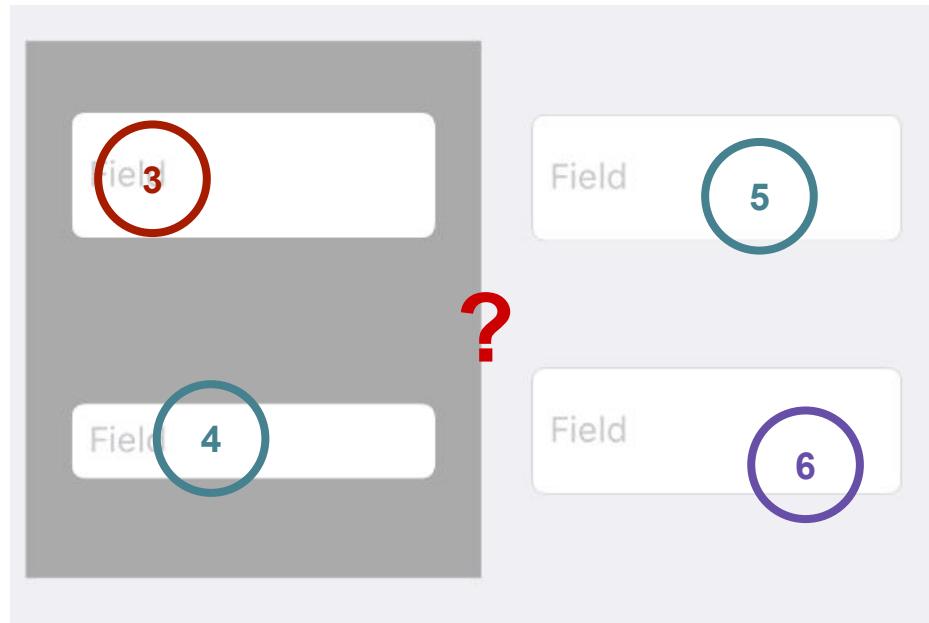
- Вызвать приватный метод
- Взять следующее поле по иерархии UIView
- Задать массив полей и перемещаться по нему

Особенности приватного метода

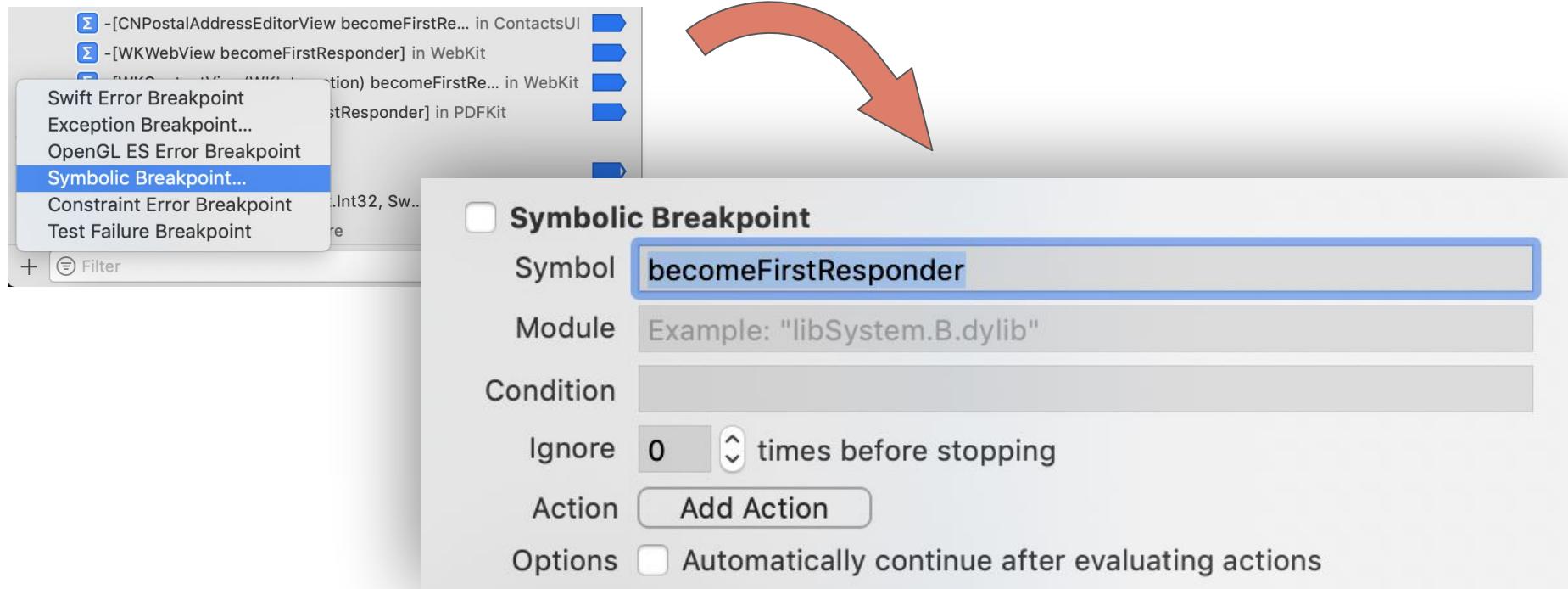
- ▼ View
 - Safe Area
 - Second field
 - First field
 - Sixth field
 - Fifth field
 - ▼ View
 - Third field
 - Fourth field
 - Constraints
 - Constraints



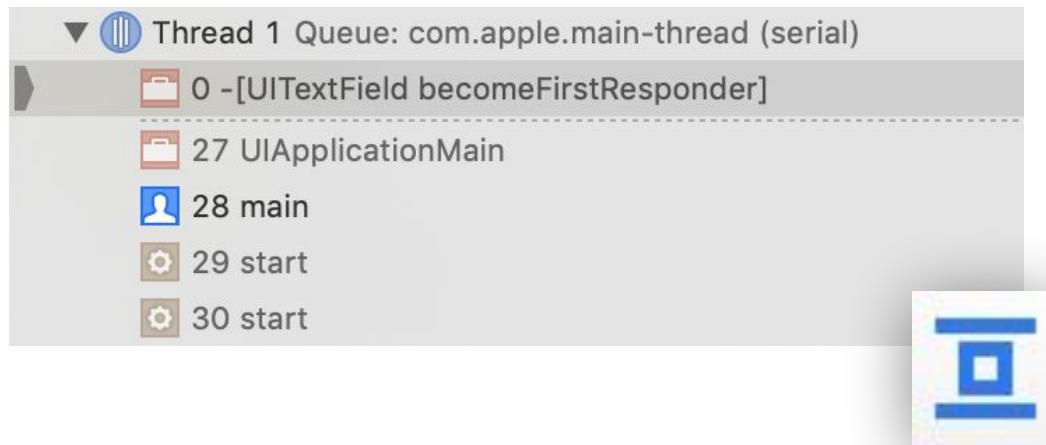
Особенности приватного метода



Отладка. С чего начать?

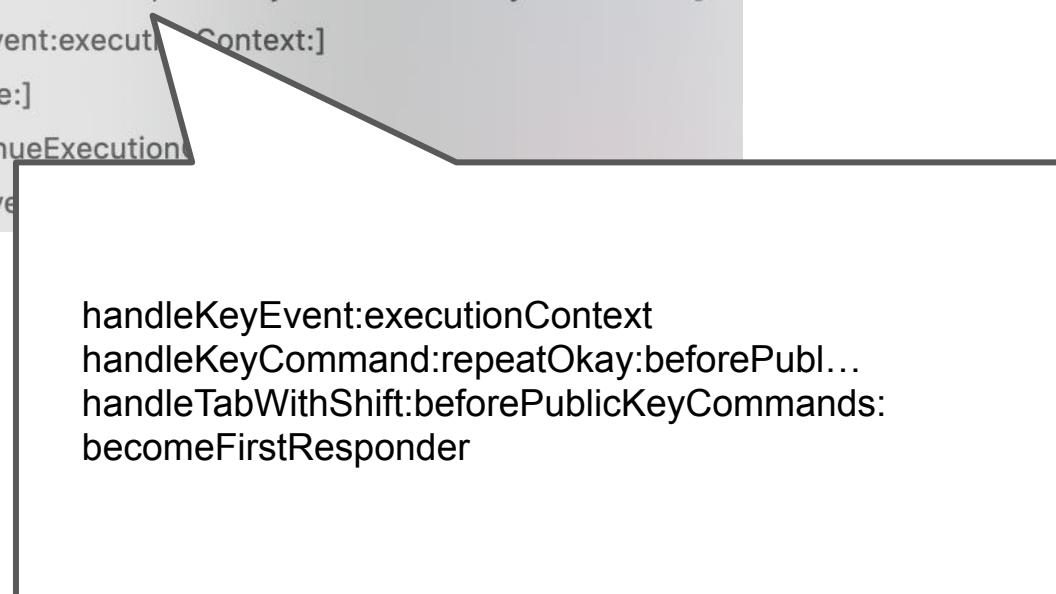


Чтение стека



Чтение стека

- 0 -[UITextField becomeFirstResponder]
- 1 -[UIKeyboardImpl handleTabWithShift:beforePublicKeyCommands:]
- 2 -[UIKeyboardImpl handleKeyCommand:repeatOkay:beforePublicKeyCommands:]
- 3 -[UIKeyboardImpl handleKeyEvent:executionContext:]
- 4 -[UIKeyboardTaskEntry execute:]
- 5 -[UIKeyboardTaskQueue continueExecution]
- 6 -[UIKeyboardImpl handleKeyEvent:executionContext:]



handleKeyEvent:executionContext
handleKeyCommand:repeatOkay:beforePubl...
handleTabWithShift:beforePublicKeyCommands:
becomeFirstResponder

Заглянем внутрь. Аннотации

```
45      0x10ae34d51 <+157>: jmp    0x10ae34d6a
46      0x10ae34d53 <+159>: movq   0xe98056(%rip), %rdi
47      0x10ae34d5a <+166>: movq   0xe4ed87(%rip), %rsi
48      0x10ae34d61 <+173>: callq  *0xa32749(%rip)
49      0x10ae34d67 <+179>: movq   %rax, %r15
50      0x10ae34d6a <+182>: testb  %r14b, %r14b
51      0x10ae34d6d <+185>: je     0x10ae34d78
52      0x10ae34d6f <+187>: leaq   0xe77e2a(%rip), %rbx
53      0x10ae34d76 <+194>: jmp    0x10ae34db9
54      0x10ae34d78 <+196>: movq   %r13, %rdi
55      0x10ae34d7b <+199>: movq   %r12, %rsi
56      0x10ae34d7e <+202>: callq  *0xa3272c(%rip)
57      0x10ae34d84 <+208>: leaq   0xe773fd(%rip), %rbx
58      0x10ae34d8b <+215>: testq  %rax, %rax
59      0x10ae34d8e <+218>: je     0x10ae34db9
60      0x10ae34d90 <+220>: movq   0xe56ba1(%rip), %rsi
61      0x10ae34d97 <+227>: movq   0xa32712(%rip), %r14
62      0x10ae34d9e <+234>: movq   %r13, %rdi
63      0x10ae34da1 <+237>: callq  *%r14
64      0x10ae34da4 <+240>: movq   0xe6f505(%rip), %rsi
```

```
; <+182>
; (void *)0x000000010bd15f80: UIWindow
; "keyWindow"
; (void *)0x0000000106d9e640: objc_msgSend

; <+196>
; "_previousKeyResponder"
; <+261>

; (void *)0x0000000106d9e640: objc_msgSend
; "_nextKeyResponder"

; <+261>
; "textInputTraits"
; (void *)0x0000000106d9e640: objc_msgSend

; "isSingleLineDocument"
```

[UIView _nextKeyResponder]

Селекторы из антоций:

- `"_rootForKeyResponderCycle"`
- `"_collectKeyViews:"`
- `"_topToBottomLeftToRightViewCompare:"`
- `"sortUsingSelector:"`

[UIView _collectKeyViews]

Селекторы из антоций:

- "_requiresKeyboardWhenFirstResponder"
- "addObject:"
- "subviews"
- "countByEnumeratingWithState:objects:count:"
- "_collectKeyViews:visibilityTest:passingTest:"
- "countByEnumeratingWithState:objects:count:"

collectKeyView

```
func collectKeyViews() -> [UIView] {
    var keyViews = [UIView]()
    if requiresKeyboard() {
        if canBecomeFirstResponder {
            keyViews.append(self)
        }
    } else {
        keyViews.append(contentsOf: subviews.flatMap { $0.collectKeyViews() })
    }
    return keyViews
}
```

Метод отбора элементов

[UIResponder _requiresKeyboardWhenFirstResponder]

- _keyboardResponder
- conformsToProtocol
- "respondsToSelector:" "isEditable"

Метод отбора элементов

```
func requiresKeyboard() -> Bool {  
    guard self is UIKeyInput else { return false }  
    guard responds(to: #selector(getter: UITextView.isEditable)) else {  
        return true  
    }  
    return (value(forKey: "isEditable") as? Bool) != false  
}
```

Порядок элементов

- "bounds"
- "convertRect:fromView:"

Порядок элементов

```
func sortedKeyViews() -> [UIView] {
    return collectKeyViews().sorted { [weak self] left, right in
        let leftOrigin = left.convert(left.bounds.origin, to:self)
        let rightOrigin = right.convert(right.frame.origin, to:self)
        if leftOrigin.y != rightOrigin.y {
            return leftOrigin.y < rightOrigin.y
        }
        return leftOrigin.x < rightOrigin.x
    }
}
```

Методика

- Без ассемблера
- Только аннотации
- Только symbol breakpoints

Свое решение

```
extension UIView {  
    func nextResponder() -> UIView  
    func prevResponder() -> UIView  
}
```

<https://github.com/sclown/nextResponder>

`UIViewController.loadView`

Описание проблемы

```
override func loadView() {  
    table = UITableView()  
    view = table  
}
```



```
override func loadView() {  
    view = UIView()  
}
```

Описание проблемы. Crash

```
*** Terminating app due to uncaught exception  
'NSInternalInconsistencyException',  
reason: '-[UIViewController _loadViewFromNibNamed:bundle:]  
loaded the "UselessView" nib but the view outlet was not set.'  
*** First throw call stack:  
...
```

Заглянем внутрь. Стек.

▼  Thread 1 Queue: com.apple.main-thread (serial)

-  0 objc_exception_throw
-  1 +[NSEException raise:format:]
-  2 -[UIViewController _loadViewFromNibNamed:bundle:]
-  3 -[UIViewController loadView]
-  4 -[UIViewController loadViewIfRequired]
-  5 -[UIViewController view]

Заглянем внутрь. Аннотации

```
38      0x11391ed0b <+131>: movq    %rax, %r14
39      0x11391ed0e <+134>: movq    %r12, %rdi
40      0x11391ed11 <+137>: movq    0x1292b30(%rip), %rsi
41      0x11391ed18 <+144>: callq   *%r13
42      0x11391ed1b <+147>: movq    %rax, %rdi
43      0x11391ed1e <+150>: callq   0x114486758
44      0x11391ed23 <+155>: movq    %rax, %rbx
45      0x11391ed26 <+158>: movq    0x1292b53(%rip), %rsi
46      0x11391ed2d <+165>: movq    %r12, %rdi
47      0x11391ed30 <+168>: movq    %r14, %rdx
48      0x11391ed33 <+171>: movq    %rax, %rcx
49      0x11391ed36 <+174>: callq   *%r13
                                         ; "nibBundle"
                                         ; symbol stub for: objc_retainAutoreleasedReturnValue
                                         ; "_loadViewFromNibNamed:bundle:"
```

Вызовы по аннотациям

- "nibName"
- "nibBundle"
- "nibBundle"
- "_loadViewFromNibNamed:bundle:"

Поиск UIViewController nibName

<https://developer.apple.com/documentation/uikit/uiviewcontroller/1621487-nibname>

nibName

The name of the view controller's nib file, if one was specified.

If the view controller class name ends with the word 'Controller', as in MyViewController, it looks for a nib file whose name matches the class name without the word 'Controller', as in MyView.nib.

Методика

- Искать упоминания символов в интернете
- Выход на описание в официальной документации является хорошим результатом
- Среди дампов приватных API можно найти дополнительные символы

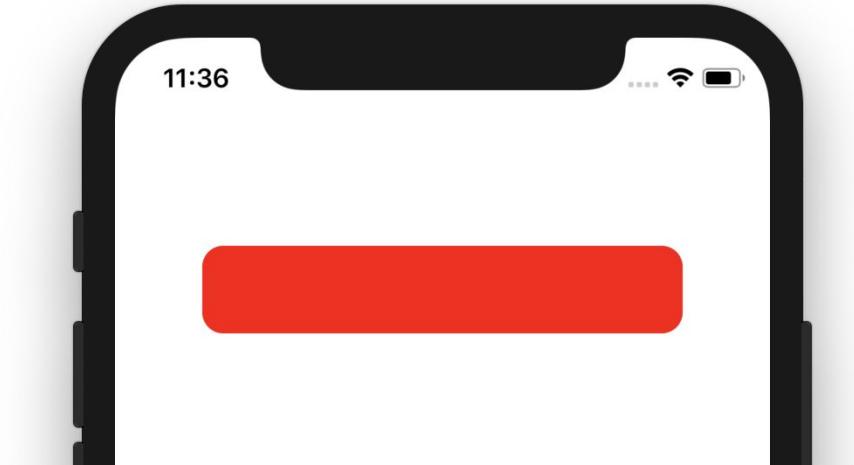
UIButton

UIButton. Описание проблемы



SnapshotTesting

UIButton. Описание проблемы



SnapshotTesting

UIButton. Описание проблемы

```
let font = UIFont(name: "Courier", size: UIFont.systemFontSize)  
xibButton.titleLabel?.font = font  
xibButton.setTitle("Interface builder made", for: .normal)
```

Первые результаты отладки

```
override func layerWillDraw(_ layer: CALayer) {  
    super.layerWillDraw(layer)  
}  
  
override func draw(_ layer: CALayer, in ctx: CGContext) {  
    super.draw(layer, in: ctx)  
}  
  
override func draw(_ rect: CGRect) {  
    super.draw(rect)  
}
```

Схема элементов кнопки

Button.layer.sublayers

```
delegate = <UIButtonLabel: 0x7fb6bc417620;  
frame = (187 25; 0 0);  
text = 'Button';  
opaque = NO;  
userInteractionEnabled = NO;  
layer = <_UILabelLayer: 0x600000d23700>>
```

Как обновить слой?

Ничего не помогает

```
button.setNeedsLayout()  
button.layoutIfNeeded()
```

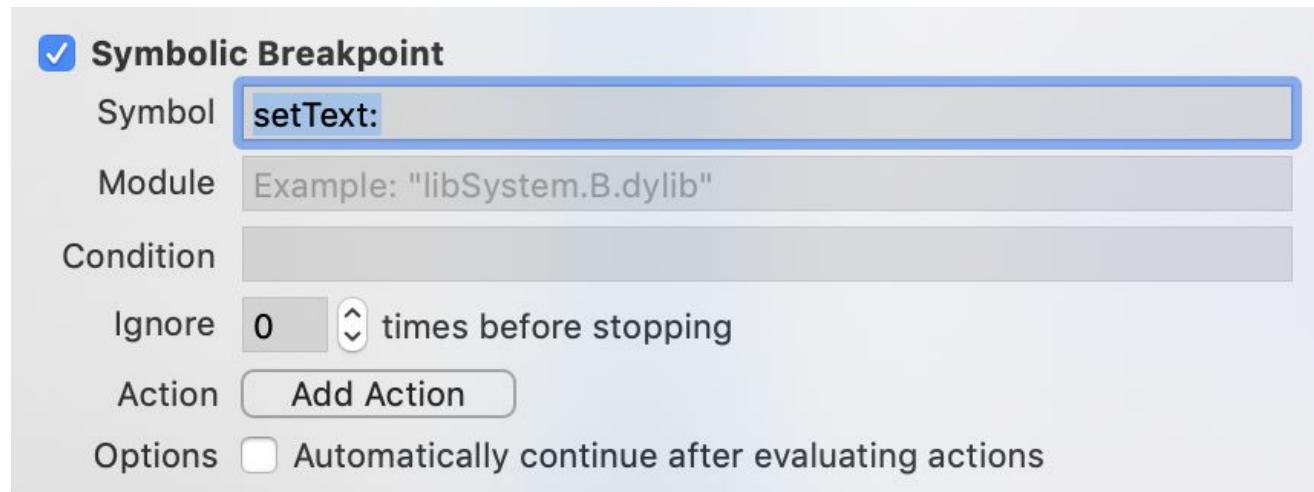
```
button.setNeedsDisplay()  
button.setNeedsLayout()  
button.layoutIfNeeded()
```

```
button.titleLabel?.setNeedsDisplay()  
button.titleLabel?.setNeedsLayout()  
button.titleLabel?.layoutIfNeeded()
```

```
button.isHighlighted = true  
button.isHighlighted = false  
button.titleLabel?.setNeedsDisplay()  
button.titleLabel?.setNeedsLayout()  
button.titleLabel?.layoutIfNeeded()  
button.setNeedsDisplay()  
button.setNeedsLayout()  
button.layoutIfNeeded()
```

Отладка. Второй заход

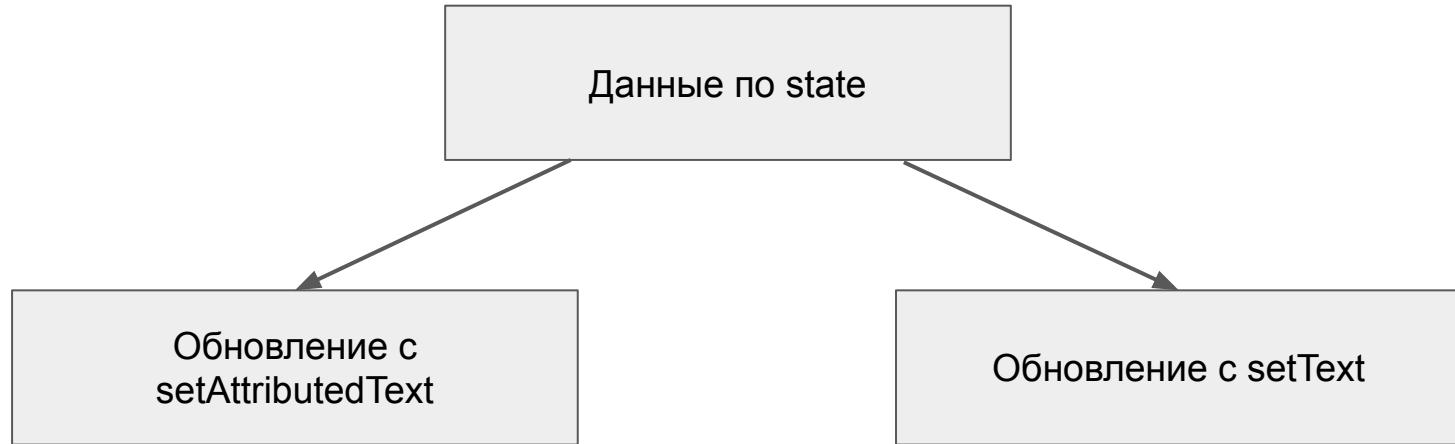
@interface UIButtonLabel : UILabel



Стек с [UILabel setText:]

- 0 -[UILabel setText:]
- 1 -[UIButton _updateTitleView]
- 2 -[UIButton titleLabel]
- 3 ViewController.viewDidLoad()

UIButton._updateTitleView



Как получить доступ к этим данным

```
0x10e42b678 <+471>: movq 0x1285bf1(%rip), %rax ; UIButton._buttonFlags  
0x10e42b67f <+478>: cmpw $0x0, (%r14,%rax)
```

0x1285bf1(%rip), %rax
\$0x0, (%r14,%rax)

Что это?

Адресация в ассемблере

<code>%<reg></code>	значение из регистра
<code>(%<reg>)</code>	чтение из памяти по адресу из регистра
<code>const_offset(base, offset, multiplication)</code>	чтение по адресу $base + offset * multiplication + const_offset$
<code>\$const_value</code>	константа

Адресация в ассемблере

%<reg>	значение из регистра
(%<reg>)	чтение из памяти по адресу из регистра
const_offset(base, offset, multiplication)	чтение по адресу base + offset * multiplication + const_offset
\$const_value	константа

Адресация в ассемблере

%<reg>	значение из регистра
(%<reg>)	чтение из памяти по адресу из регистра
const_offset(base, offset, multiplication)	чтение по адресу base + offset * multiplication + const_offset
\$const_value	константа

Адресация в ассемблере

%<reg>	значение из регистра
(%<reg>)	чтение из памяти по адресу из регистра
const_offset(base, offset, multiplication)	чтение по адресу base + offset * multiplication + const_offset
\$const_value	константа

Работа с регистрами

- ▶ **R Exception State Registers**
- ▶ **R Floating Point Registers**
- ▼ **R General Purpose Registers**

rax = (unsigned long) 0x00000000000d0000

rbx = (unsigned long) 0x00007f8007c15df0

rcx = (unsigned long) 0x0000000002100279

rdx = (unsigned long) 0x0000000000000303

rdi = (unsigned long) 0x00007f8007c15df0

rsi = (unsigned long) 0x00007fff51e13351

rbp = (unsigned long) 0x00007fee359d3a0

rsp = (unsigned long) 0x00007fee359d388

r8 = (unsigned long) 0x00007fff89b938a0

r9 = (unsigned long) 0x4032000000000000

Работа с регистрами

register read

General Purpose Registers:

rax = 0x000000000000d0000

rbx = 0x00007f8007c15df0

...

r8 = 0x00007fff89b938a0 libsystem_pthread.dylib`_pthread_keys

r9 = 0x4032000000000000

r10 = 0x00007fff8675b438 (void *)0x000001150000001b

r11 = 0x00007fff46b4e60d UIKitCore`-[UIButton _updateTitleView]

...

r15 = 0x00007fff503b1780 libobjc.A.dylib`objc_msgSend

rip = 0x00007fff46b4e60d UIKitCore`-[UIButton _updateTitleView]

Работа с регистрами

po \$arg1

```
<UIButton: 0x7f8007c15df0; frame = (50 144; 314 100); opaque = NO; autoresize = RM+BM; layer = <CALayer: 0x60000335d7e0>>
```

	arg1	arg2	arg3	arg4	arg5	arg6	arg7	arg8
simulator	rax	rsi	rdx	rcx	r8	r9	-	-
iPhone	x0	x1	x2	x3	x4	x5	x6	x7

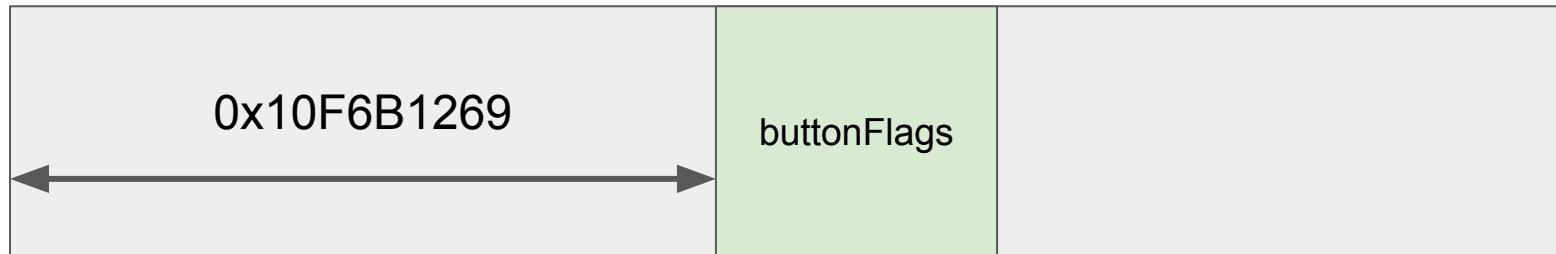
Расчет смещения

```
0x10e42b678 <+471>: movq    0x1285bf1(%rip), %rax      ; UIButton._buttonFlags  
0x10e42b67f <+478>:  cmpw    $0x0, r14,%rax
```

$$0x1285bf1 + \%rip = 0x1285bf1 + 0x10e42b678 = 0x10F6B1269$$

Расчет смещения

UIButton



Адрес структуры в данном объекте

```
0x10e42b678 <+471>: movq    0x1285bf1(%rip), %rax      ; UIButton._buttonFlags  
0x10e42b67f <+478>: cmpw    $0x0, (%r14,%rax)
```



%r14 + %rax

по (_buttonFlags*)(\$r14 + \$rax)

UIButton._buttonFlags

```
struct {
    unsigned reversesTitleShadowWhenHighlighted : 1;
    unsigned adjustsImageWhenHighlighted : 1;
    unsigned adjustsImageWhenDisabled : 1;
    unsigned autosizeToFit : 1;
    unsigned disabledDimsImage : 1;
    unsigned showsTouchWhenHighlighted : 1;
    unsigned buttonType : 8;
    unsigned shouldHandleScrollerMouseEvent : 1;
    unsigned titleFrozen : 1;
    unsigned resendTraitToImageViews : 2;
    unsigned animateNextHighlightChange : 1;
    unsigned blurEnabled : 1;
    unsigned visualEffectViewEnabled : 1;
    unsigned suppressAccessibilityUnderline : 1;
    unsigned requiresLayoutForPropertyChange : 1;
    unsigned adjustsImageSizeForAccessibilityContentSizeCategory : 1;
    unsigned disableAutomaticTitleAnimations : 1;
} _buttonFlags;
```

Чтение приватных структур

```
struct UIButtonFlags{
    unsigned reversesTitleShadowWhenHighlighted : 1;
    unsigned adjustsImageWhenHighlighted : 1;
    unsigned adjustsImageWhenDisabled : 1;
    unsigned autosizeToFit : 1;
    unsigned disabledDimsImage : 1;
    unsigned showsTouchWhenHighlighted : 1;
    unsigned buttonType : 8;
...
    unsigned disableAutomaticTitleAnimations : 1;
};
```

Чтение данных в отладке

```
NSDictionary* printFlags(struct UIButtonFlags* flags) {  
    return @ {  
        @"reversesTitleShadowWhenHighlighted": @(flags->reversesTitleShadowWhenHighlighted),  
        @"adjustsImageWhenHighlighted": @(flags->adjustsImageWhenHighlighted),  
        ...  
> po printFlags((UIButtonFlags*)0x0075509483045)
```

Значения в buttonFlags

```
buttonType : 8;  
titleFrozen : 1;  
requiresLayoutForPropertyChange : 1;
```

Итоговое решение

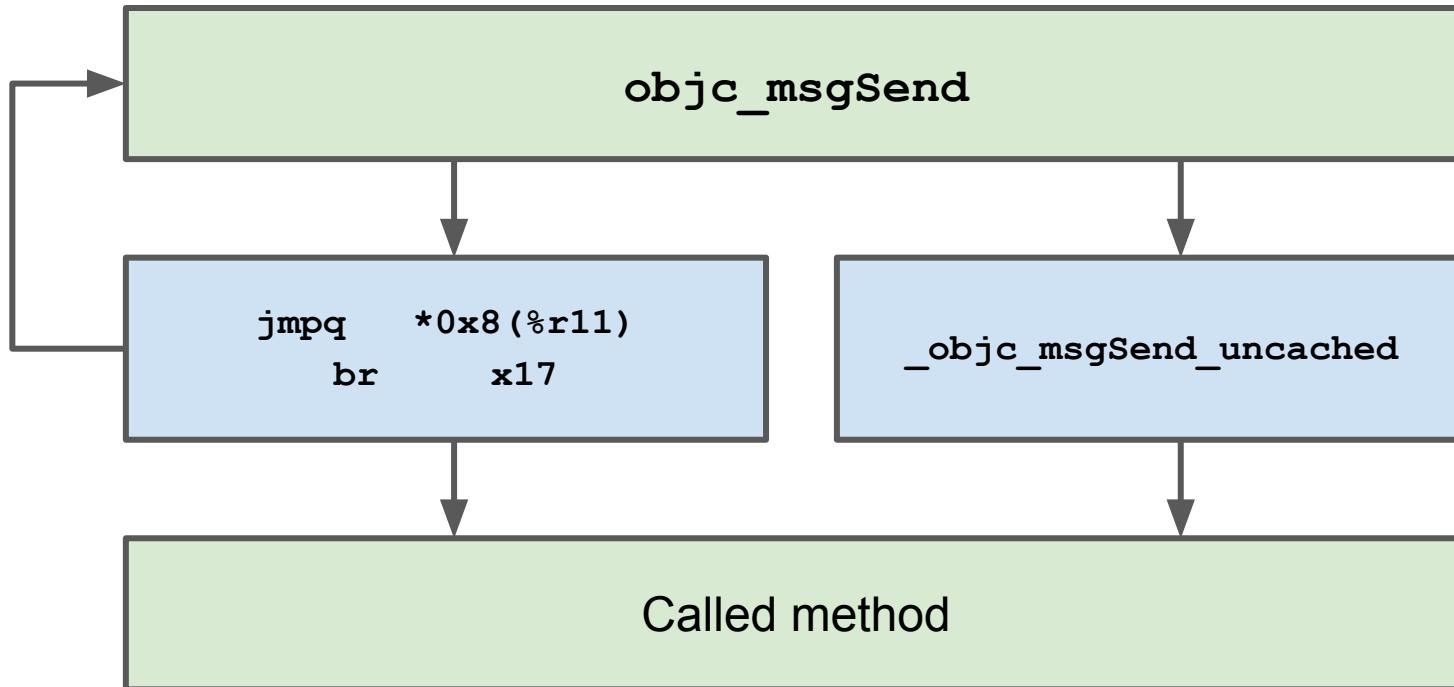
```
override func setUp() {  
    UIView.setAnimationsEnabled(false)  
}  
  
override func tearDown() {  
    UIView.setAnimationsEnabled(true)  
}
```

Полезные приемы

LLDB. Вычисления и распечатка

po \$arg1, \$r14, \$rax	значение из регистра
p/w 0x1285bf1 + \$rip	вычисление с результатом в HEX
x 0x1285bf1 (me r, memory read)	чтение памяти по адресу

Прохождение sendMessage



Заключение

- Можно отлаживаться без знания ассемблера
- В качестве результата можно получить официальную документацию о том, как действовать правильно
- Для сбора символов можно использовать дампы приватного API
- Ассемблер читать не так уж и сложно. Хотя бы даже “со словарем”