

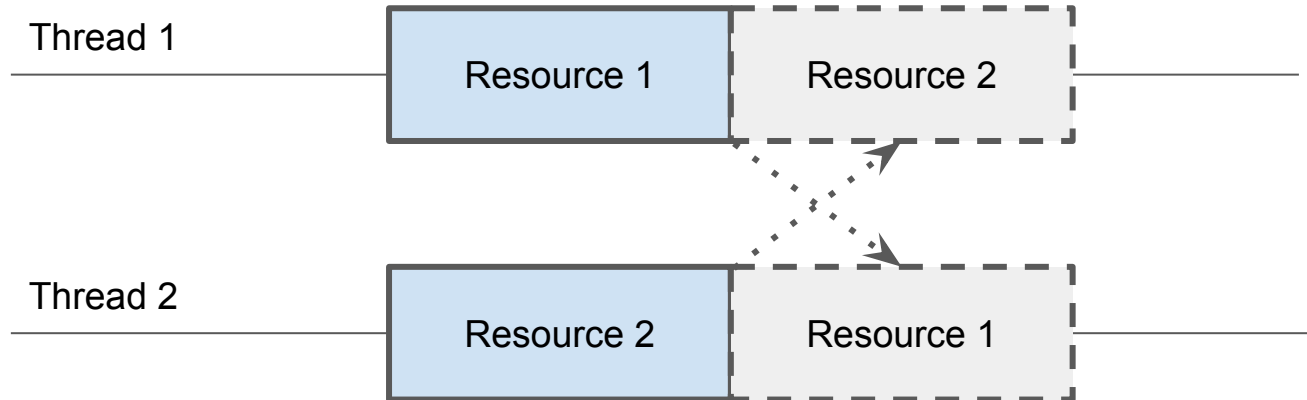
Отладка многопоточных приложений

Дмитрий Куркин
Spark Networks

План

1. Проблемы
2. Инструменты
3. Логи
4. Watchdog

Deadlock



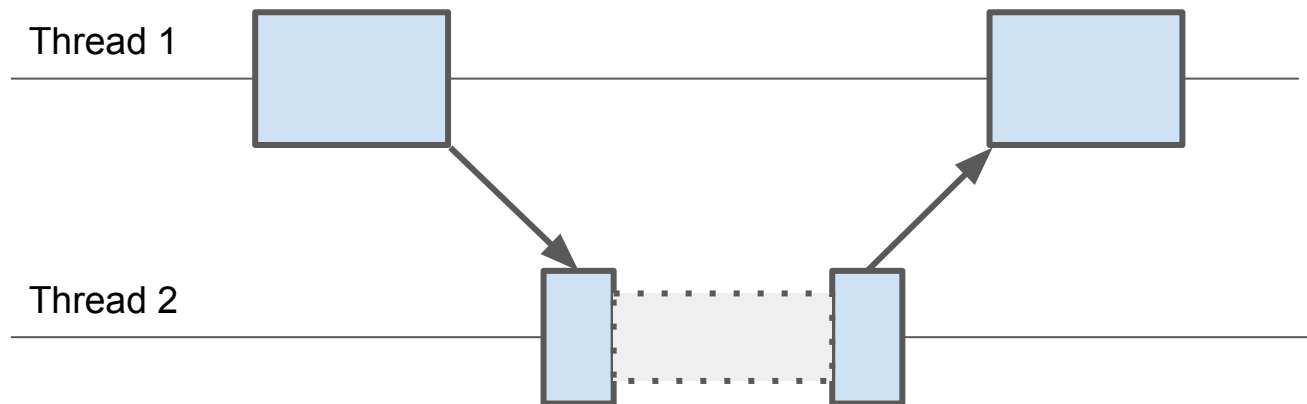
HE Deadlock

DispatchQueue.main.sync

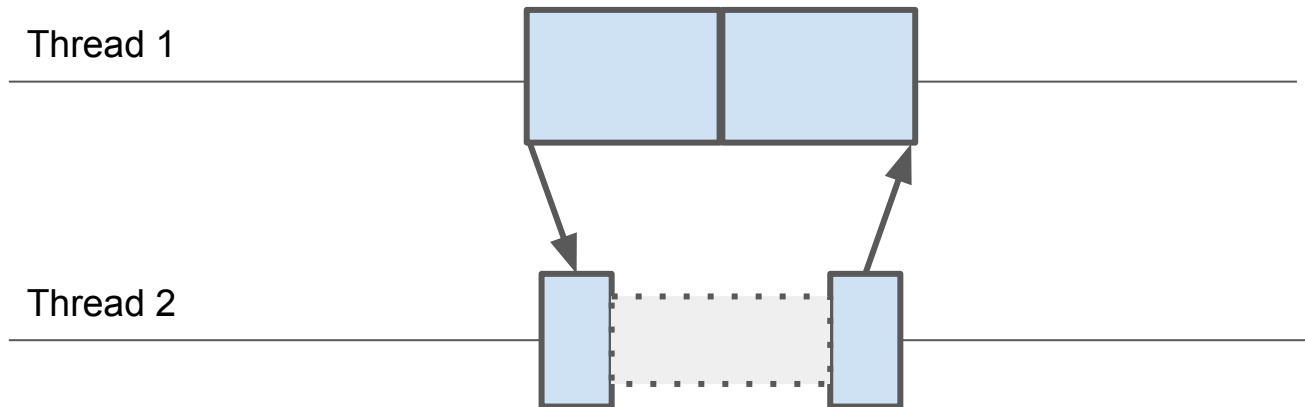
HE Concurrency

... `async`

Производительность



Производительность



Ошибки синхронизации

Открыть заказ

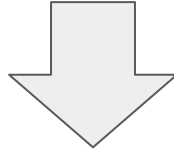
Добавить товар

Добавить товар

Закрыть заказ

Добавить товар

Thread



Queue

Простой случай
(воспроизводится)

Debugger XCode (LLDB)

```
66     private func parse(_ data: Data) -> Matches? {  
67         if !isOk {  
68             print("isOk is not OK")  
69         }
```

Debugger XCode (LLDB)

The screenshot shows the Xcode interface with a Swift code editor. A breakpoint is set at line 68 of ListingService.swift. A dialog box is open for configuring the breakpoint. The dialog box has the following fields:

- Enable Breakpoint in ListingService.swift:68**
- Name:
- Condition:
- Ignore: 2 times before stopping
- Action: Debugger Command
- Command: po isOK
- Options: Automatically continue after evaluating actions

```
private func parse(_ data: Data) -> Matches? {
    if !isOK {
        print("isOK is not OK")
    }
}
```

```
encoder()
Strategy = .convertFromS
der.decode(Matches.self,
```

Debugger XCode (LLDB)

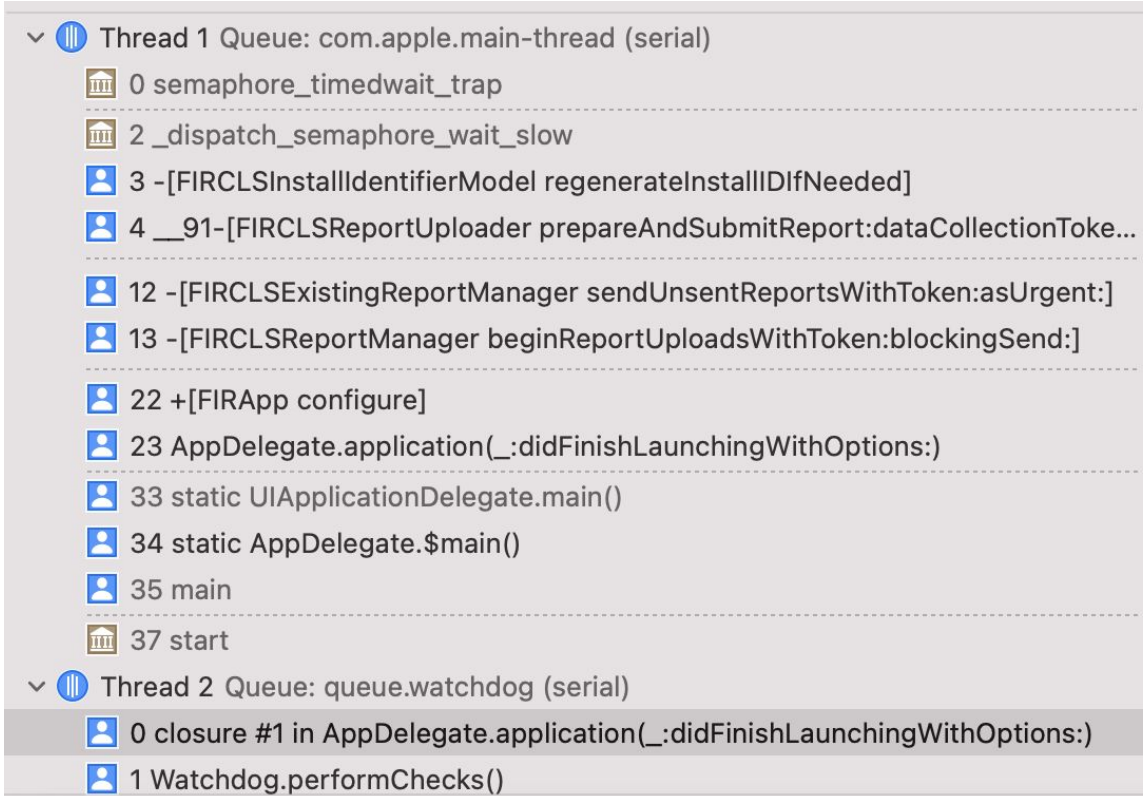
The screenshot shows the LLDB debugger interface with the following content:

- Disk:** 68 KB/s
- Network:** Zero KB/s
- Thread 1:** Queue: com.apple.main-thread (serial)
- Thread 2:** Queue: queue.watchdog (serial)
 - 0 closure #1 in AppDelegate.application(_:did...)
 - 1 Watchdog.performChecks()
 - 2 closure #1 in Watchdog.schedule()
 - 11 start_wqthread

At the bottom, it indicates: "Enqueued from queue.watchdog (Thread 2)"

```
15 @main
16 class AppDelegate: UIResponder
17     func application(
18         _ application: UIAppli
19         didFinishLaunchingWith
20     ) -> Bool {
21         Watchdog.ping(queue: .
22             Watchdog.crash()
23         })
24         FirebaseApp.configure(
25             Crashlytics.crashlytic
26         return true
27     }
28
```

Debugger XCode (LLDB)



The screenshot shows the LLDB thread list in Xcode. It is organized into two main sections: Thread 1 and Thread 2. Thread 1 is the main thread and contains a list of frames from 0 to 37. Thread 2 is a watchdog thread and contains two frames, 0 and 1. Each frame is preceded by an icon: a building for system or library frames, a person for application frames, and a person with a plus sign for the main thread's start frame. The frames are separated by dashed lines.

- Thread 1 Queue: com.apple.main-thread (serial)
 - 0 semaphore_timedwait_trap
 - 2 _dispatch_semaphore_wait_slow
 - 3 -[FIRCLSInstallIdentifierModel regenerateInstallIDIfNeeded]
 - 4 __91-[FIRCLSReportUploader prepareAndSubmitReport:dataCollectionToke...
 - 12 -[FIRCLSExistingReportManager sendUnsentReportsWithToken:asUrgent:]
 - 13 -[FIRCLSReportManager beginReportUploadsWithToken:blockingSend:]
 - 22 +[FIRApp configure]
 - 23 AppDelegate.application(_:didFinishLaunchingWithOptions:)
 - 33 static UIApplicationDelegate.main()
 - 34 static AppDelegate.\$main()
 - 35 main
 - 37 start
- Thread 2 Queue: queue.watchdog (serial)
 - 0 closure #1 in AppDelegate.application(_:didFinishLaunchingWithOptions:)
 - 1 Watchdog.performChecks()

Thread Sanitizer

>  **Build**
3 targets

>  **Run**
Debug

>  **Test**
Debug

>  **Profile**
Release

>  **Analyze**
Debug

>  **Archive**

 matches >  iPod touch (7th generation) (14.5)

Info

Arguments



Options

Diagnostics

Runtime Sanitization Address Sanitizer ⓘ
Requires recompilation Detect use of stack after return
 Thread Sanitizer
 Undefined Behavior Sanitizer ⓘ

Runtime API Checking Main Thread Checker

Thread Sanitizer

```
...
45     private func performChecks() {
46         if (!isCheckedPassed) {  Data race in matches.Watchdog.performChecks() -> () at 0x7b1400039da0
47             issueHandler?()
48         }
49         isCheckedPassed = false
50         DispatchQueue.main.async { [weak self] in
51             self?.isCheckedPassed = true
52             self?.checkThreadSanitiser = "Bad on main queue"  Data race in closure #1 () -> () in matches...
53         }
54         checkThreadSanitiser = "Bad on background queue"
55     }
```

Только Simulator

Thread Sanitizer

WARNING: ThreadSanitizer: data race (pid=87600)

Read of size 8 at 0x7b1400039dd8 by main thread:

#0 closure #1 in Watchdog.performChecks() WatchDog.swift:52 (matches:x86_64+0x100008195)

...

#6 main <compiler-generated> (matches:x86_64+0x10002a350)

Previous write of size 8 at 0x7b1400039dd8 by thread T1:

#0 Watchdog.performChecks() WatchDog.swift:54 (matches:x86_64+0x100007d14)

...

#5 _dispatch_client_callout <null>:2 (libdispatch.dylib:x86_64+0x474d)

Location is heap block of size 65 at 0x7b1400039da0 allocated by main thread:

#0 __sanitizer_mz_malloc <null>:2 (libclang_rt.tsan_iossim_dynamic.dylib:x86_64+0x522ca)

#1 _malloc_zone_malloc <null>:2 (libsystem_malloc.dylib:x86_64+0x12d4c)

#2 static Watchdog.ping(queue:interval:handleIssue:) WatchDog.swift:23 (matches:x86_64+0x100006e6c)

...

#7 main <compiler-generated> (matches:x86_64+0x10002a350)

Thread T1 (tid=17660846, running) is a GCD worker thread

SUMMARY: ThreadSanitizer: data race WatchDog.swift:52 in closure #1 in Watchdog.performChecks()

Thread Sanitizer

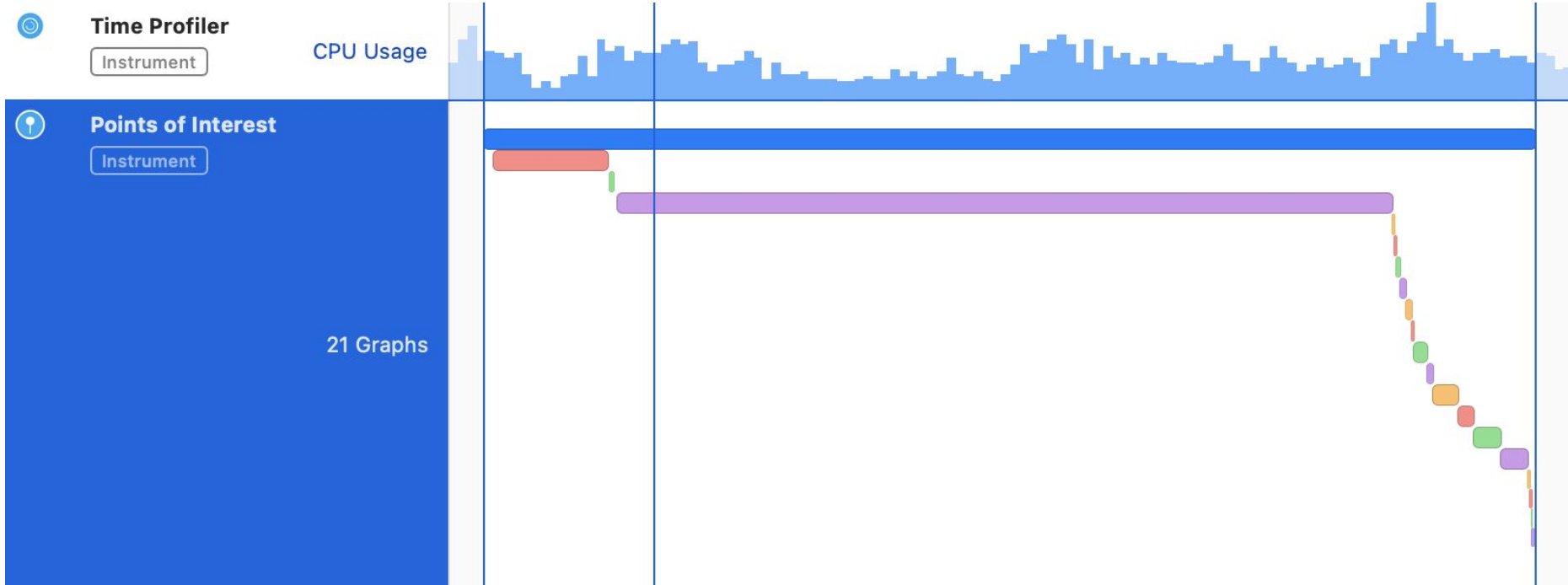
К сожалению:

- 2x - 20x performance downgrade
- Only Simulator

Time Profiler



Points of Interest



Signpost API

```
let signposter = OSSignposter()  
let signpostID = signposter.makeSignpostID()  
let state = signposter.beginInterval("processRequest", id: signpostID)  
signposter.endInterval("processRequest", state)
```

Signpost API

```
static let log = OSLog(subsystem: "boot", category: .pointsOfInterest)
os_signpost(.begin, log: .points, name: "Boot")
os_signpost(.end, log: .points, name: "Boot")
```

Signpost API

```
kdebug_signpost_start(24, 0, 0, 0, 1)  
kdebug_signpost_end(24, 0, 0, 0, 1)
```

Повторяется на Staging-сборке
(Разбираемся по логам)

Как забрать логи

No SIM 12:08 91%

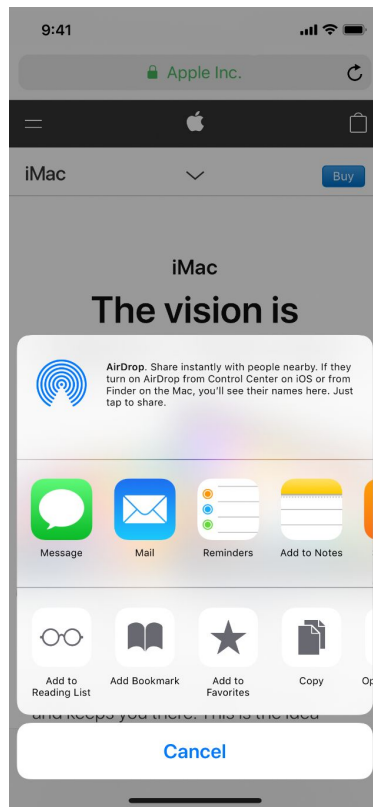
← Send feedback Preview

Description (optional)

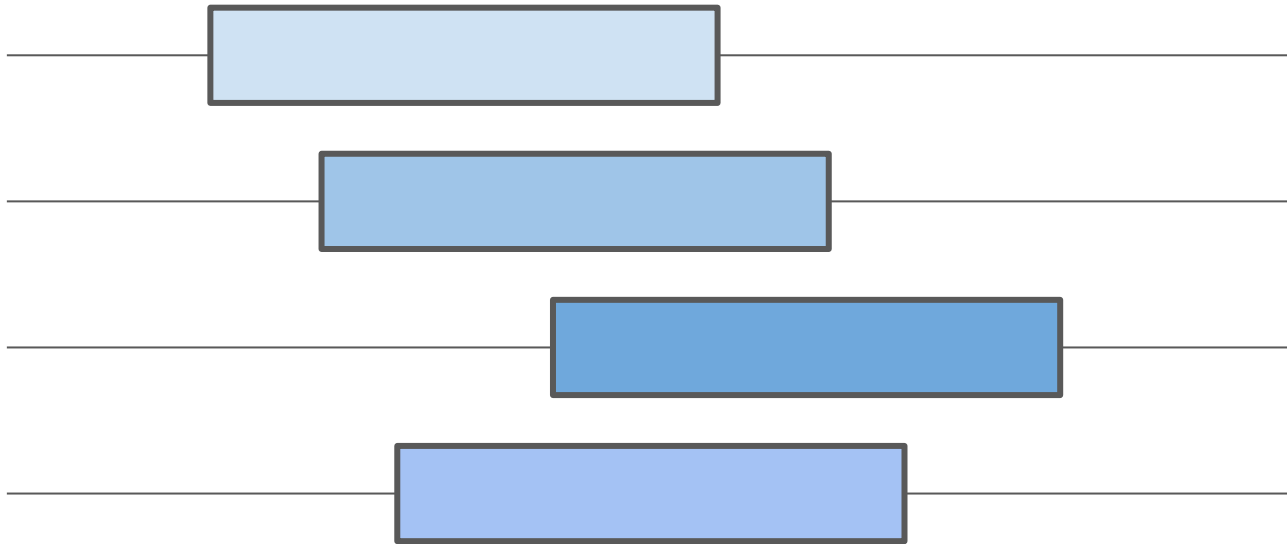
Include system data
Share system data with Google to better diagnose the problem. ✓

Include screenshot
Send screenshot of the last page that you were on. ✓

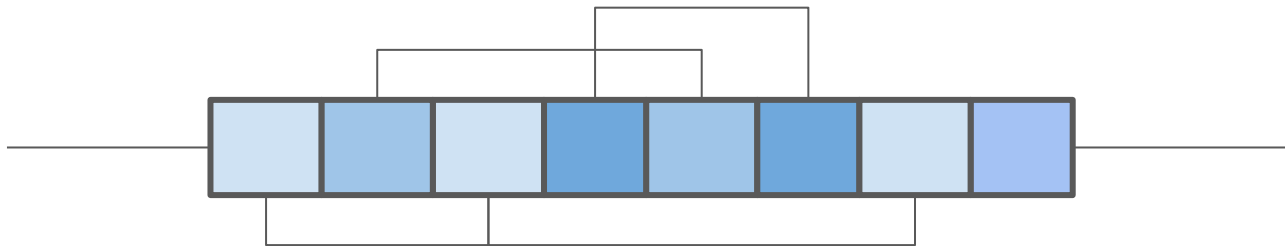
Как забрать логи



Множественные операции



Множественные операции



Множественные операции

```
po self  
Unmanaged.passUnretained(self).toOpaque()
```

```
Operation().name
```

```
OperationQueue.main.name
```

```
DispatchQueue.main.label
```

Анализ логов

```
grep 'some text' file-name.log
```

Grep. Основы

```
grep 'some text' file-name.log
```

'some text' – текст для поиска. Регулярное выражение
file-name.log – файл с текстом. Wildcard

Поиск нескольких выражений

```
grep -e 'Some Text' -e 'Another Text' -v 'Wrong Text' *
```

-e - для поиска нескольких выражений

-v - исключить лишнее

Поиск нескольких выражений

```
grep -e 'sceneDidBecomeActive' -e 'sceneWillResignActive' *
```

```
2021-11-17 12:37:48.973377+0100 matches[90502:5128926] sceneDidBecomeActive
2021-11-17 12:38:07.334660+0100 matches[90502:5128920] sceneWillResignActive
2021-11-17 12:38:12.922395+0100 matches[90502:5128920] sceneDidBecomeActive
2021-11-17 12:38:20.017386+0100 matches[90502:5128920] sceneWillResignActive
2021-11-17 12:38:35.841717+0100 matches[90502:5129937] sceneDidBecomeActive
```

Вырезать лишнее

```
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/matches
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/profile
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/sendsmile
request https://api.backend.com/status
request https://api.backend.com/status
request https://api.backend.com/status
```

Вырезать лишнее

```
grep -e 'https://api.backend.com' -v 'status' *
```

```
request https://api.backend.com/matches  
request https://api.backend.com/profile  
request https://api.backend.com/sendsmile
```

Подсчет количества

```
grep -c 'sendsmile' *
```

```
./Logs/log-2021-11-12.log:0  
./Logs/log-2021-11-13.log:1  
./Logs/log-2021-11-14.log:0  
./Logs/log-2021-11-15.log:28  
./Logs/log-2021-11-16.log:0  
./Logs/log-2021-11-17.log:2
```

Подсчет количества

```
grep -c 'lock' *  
grep -c 'unlock' *
```

Контекст

- -А – контекст ниже
- -В – контекст выше
- -С – контекст выше и ниже

КОНТЕКСТ

```
sceneDidBecomeActive
reloadData
contentOffset: 20.0
contentOffset: 154.0
contentOffset: 387.0
reloadData
contentOffset: 154.0
sceneWillResignActive
reloadData
reloadData
sceneDidBecomeActive
contentOffset: 5.0
contentOffset: 240.0
contentOffset: 534.0
reloadData
contentOffset: 89.0
request https://api.backend.com/sendsmile
contentOffset: 1045.0
contentOffset: 273.0
contentOffset: 534.0
reloadData
contentOffset: 89.0
sceneWillResignActive
reloadData
reloadData
sceneDidBecomeActive
```


КОНТЕКСТ

```
grep -C 3 'sendsmile' *
```

```
--
```

```
contentOffset: 534.0
```

```
reloadData
```

```
contentOffset: 89.0
```

```
request https://api.backend.com/sendsmile
```

```
contentOffset: 1045.0
```

```
contentOffset: 273.0
```

```
contentOffset: 534.0
```

```
--
```

Конвейер

```
grep -e 'sendsmile' -e 'Active' | grep --color -C 3 'send'
```

Конвейер

```
grep -e 'sendsmile' -e 'Active'
```

```
sceneWillResignActive  
sceneDidBecomeActive  
request https://api.backend.com/sendsmile  
sceneWillResignActive  
sceneDidBecomeActive  
sceneWillResignActive  
sceneDidBecomeActive  
sceneWillResignActive  
sceneDidBecomeActive  
sceneWillResignActive  
sceneDidBecomeActive  
sceneWillResignActive  
sceneDidBecomeActive  
request https://api.backend.com/sendsmile  
sceneWillResignActive  
sceneDidBecomeActive
```

Конвейер

```
grep -e 'sendsmile' -e 'Active' | grep --color -C 2 'send'
```

```
sceneWillResignActive  
sceneDidBecomeActive  
request https://api.backend.com/sendsmile  
sceneWillResignActive  
sceneDidBecomeActive  
--  
--  
sceneWillResignActive  
sceneDidBecomeActive  
request https://api.backend.com/sendsmile  
sceneWillResignActive  
sceneDidBecomeActive
```

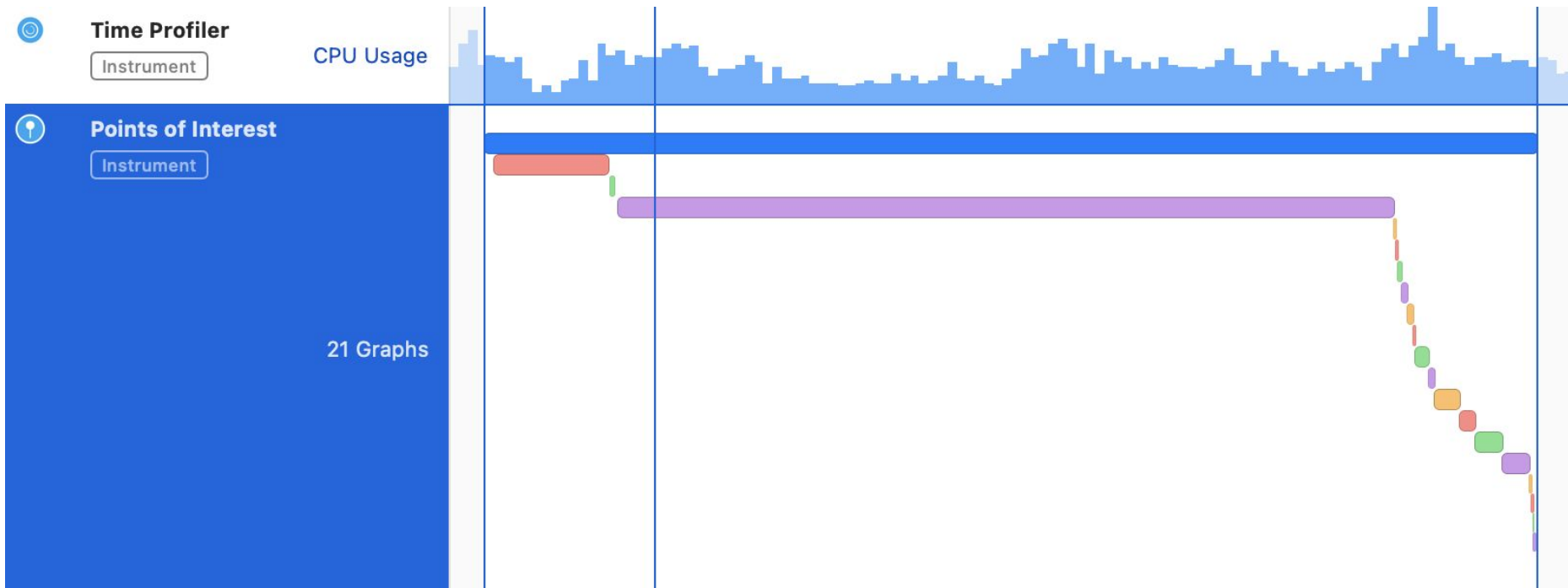
Формат логов

```
⦿⦿ DefaultAPIRequest - Headers: {  
  "X-App-Version" = "5.2.7";  
  "X-Device-Id" = 45f4e01c7a8a42bab33b96fbd00eaa059c4bd382;  
  "X-Device-Model" = "x86_64";  
  "X-Device-Platform" = iOS;  
  "X-Device-Version" = "15.0";  
  "X-Locale" = "en_US";  
  "X-Service-Auth" = "SOME-TOKEN";  
  "X-Service-Key" = "SOME-KEY";  
}
```

Логи - это не беллетристика



Логи в нашей голове



СЛИШКОМ ДЛИННЫЙ

```
👁👁 DefaultAPIRequest - Headers: {  
  "X-App-Version" = "5.2.7";  
  "X-Device-Id" = 45f4e01c7a8a42bab33b96fbd00eaa059c4bd382;  
  "X-Device-Model" = "x86_64";  
  "X-Device-Platform" = iOS;  
  "X-Device-Version" = "15.0";  
  "X-Locale" = "en_US";  
  "X-Service-Auth" = "SOME-TOKEN";  
  "X-Service-Key" = "SOME-KEY";  
}
```


Сложно забрать в контекст

```
DefaultAPIRequest - Headers: {  
  "X-App-Version" = "5.2.7";  
  "X-Device-Id" = 45f4e01c7a8a42bab33b96fbd00eaa059c4bd382;  
  "X-Device-Model" = "x86_64";  
  "X-Device-Platform" = iOS;  
  "X-Device-Version" = "15.0";  
  "X-Locale" = "en_US";  
  "X-Service-Auth" = "SOME-TOKEN";  
  "X-Service-Key" = "SOME-KEY";  
}
```

Эмоджи



DefaultAPIRequest - Headers: {

```
dkurkin@NB0695 Projects % pwd
/Users/dkurkin/Documents/Projects
dkurkin@NB0695 Projects % grep --color "???"
```

Не воспроизводится.
Не все замечают проблему

Watchdog

Фоновый поток, который следит за другими.

Watchdog

Таймер на фоновой очереди

```
private fun schedule() {  
    watchdogQueue.asyncAfter(deadline: .now() + checkInterval) {  
        self.performChecks()  
        self.schedule()  
    }  
}
```

Проверки

Выполнение ring-задачи на целевой очереди

```
if worker?.wait(timeout: .now()) == .timedOut {  
    issueHandler?()  
}  
let worker = DispatchWorkItem(block: { })  
queue?.async(execute: worker)  
self.worker = worker
```

Проверки

Проверка значений

```
if networkService.operations.count > 20 {  
    issueHandler()  
}
```

```
if storage.state.isValid {  
    issueHandler()  
}
```

Результаты. Стек всех потоков

- main - только Crash
- остальные - pthread API

Результаты. Не критическая ошибка Crashlytics

```
Crashlytics.crashlytics().record(error: error)
```

```
Crashlytics.crashlytics().record(exceptionModel:report)
```

Итого

- Логи + Grep
- Watchdog
- Xcode
- Instruments

Однопоточных приложений И удачной охоты

Вопросы

Дмитрий Куркин

dmitry.kurkin@spark.net

Telegram: @kurkinm

<https://www.facebook.com/sclown.kurkin>

<https://github.com/sclown>