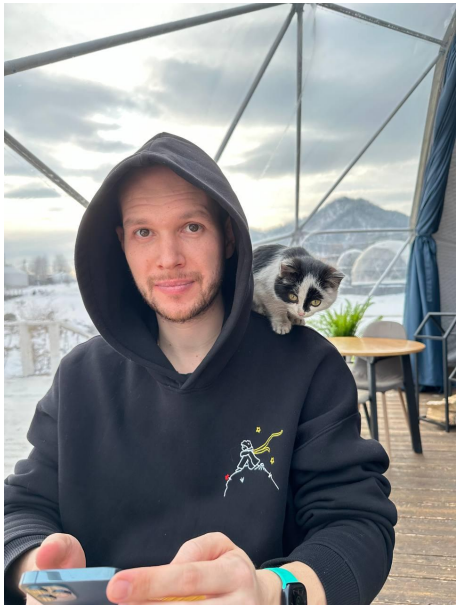
Фантастические баги

как их ловить и править



Дмитрий Кузнецов
Мобильный разработчик

Кратко о моём опыте

- Более 13 лет в коммерческой разработке, из них 8+ в мобильной
- Windows, macOS, Android, iOS
- C++, Objective-C, Swift, Java, Kotlin
- Qt Framework, Kotlin Multiplatform
- Обработка мультимедиа, в том числе на мобильных устройствах

Виды багов

- долгожитель

Виды багов

- долгожитель
- разноразмерный

Виды багов

- долгожитель
- разноразмерный
- мимикрирующий

Виды багов

- долгожитель
- разноразмерный
- мимикрирующий
- красный

Виды багов

- долгожитель
- разноразмерный
- мимикрирующий
- красный
- жадный

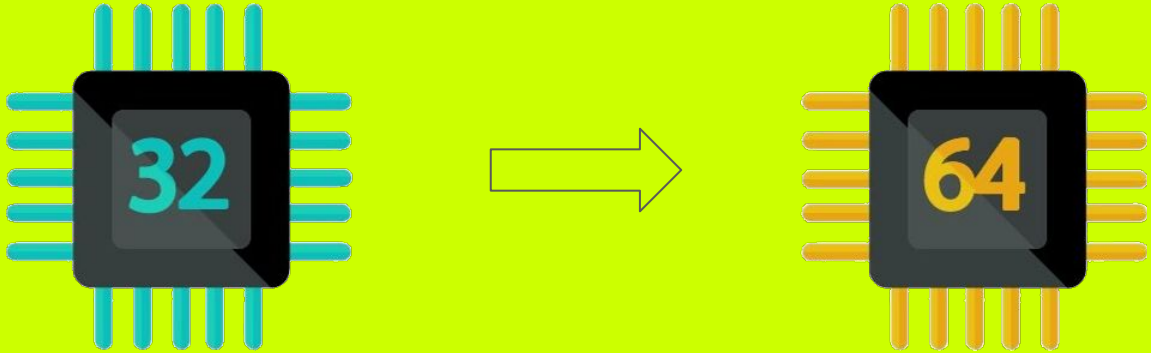
Виды багов

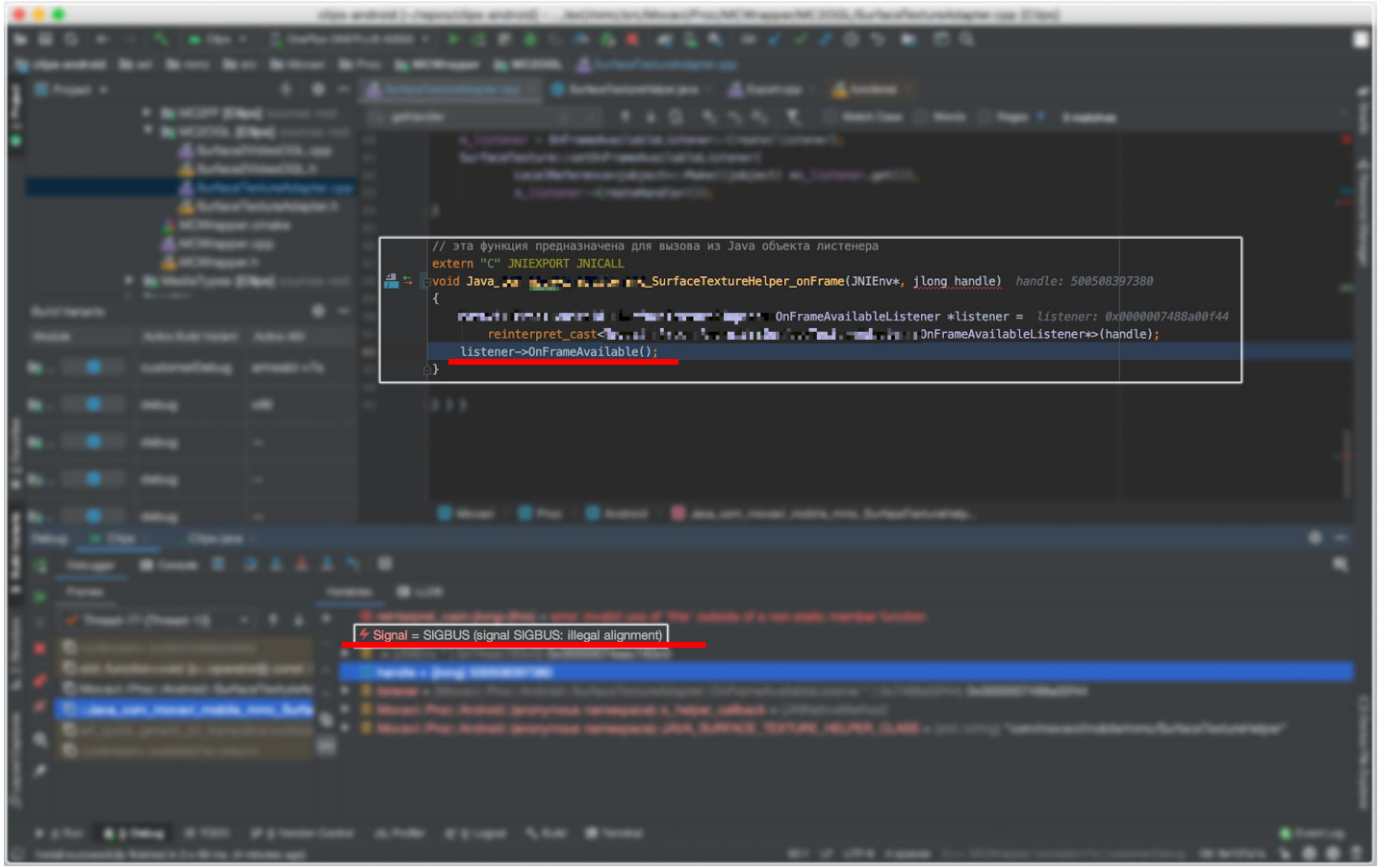
- долгожитель
- разноразмерный
- мимикрирующий
- красный
- жадный
- беспамятный

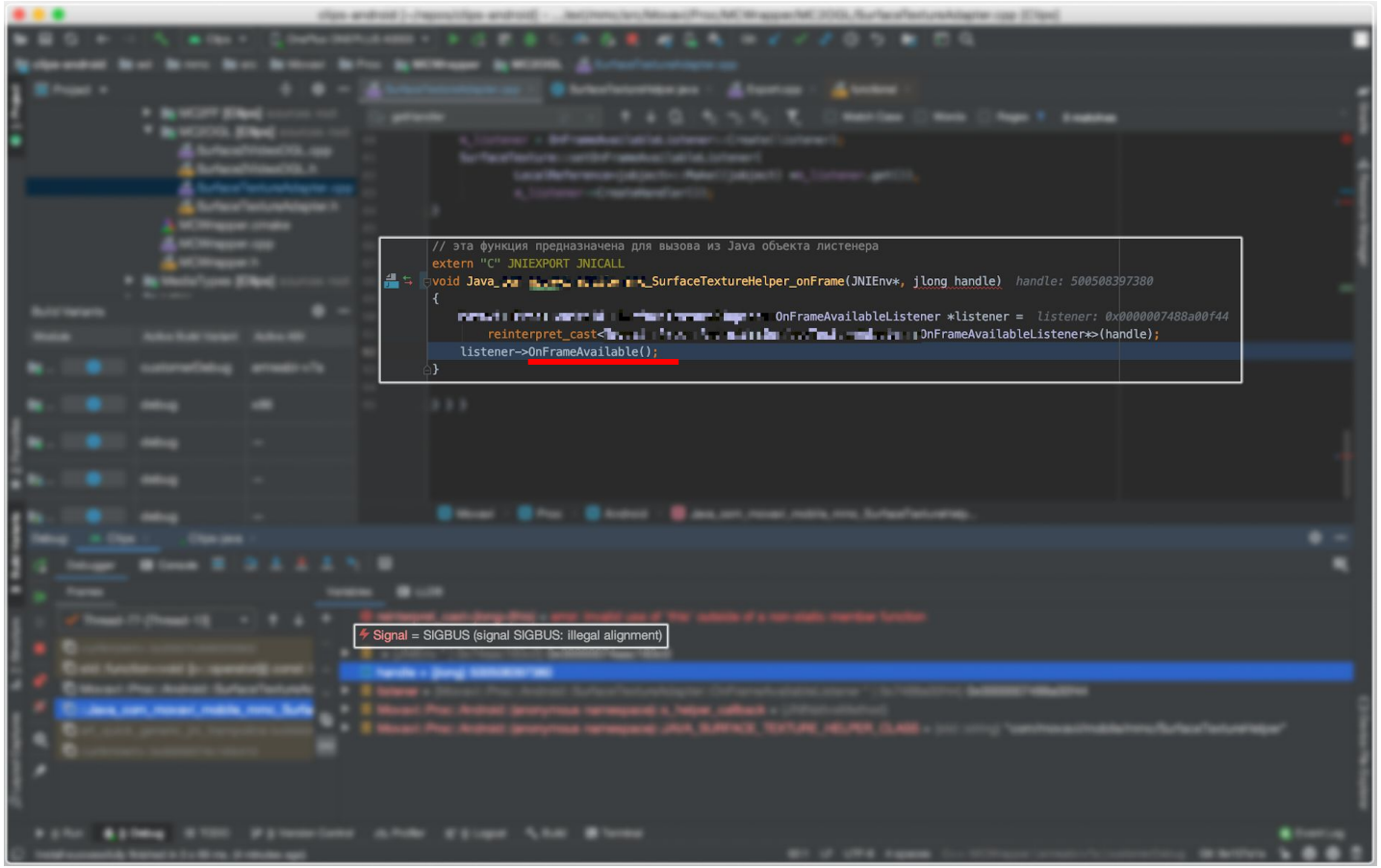
Баг долгожитель

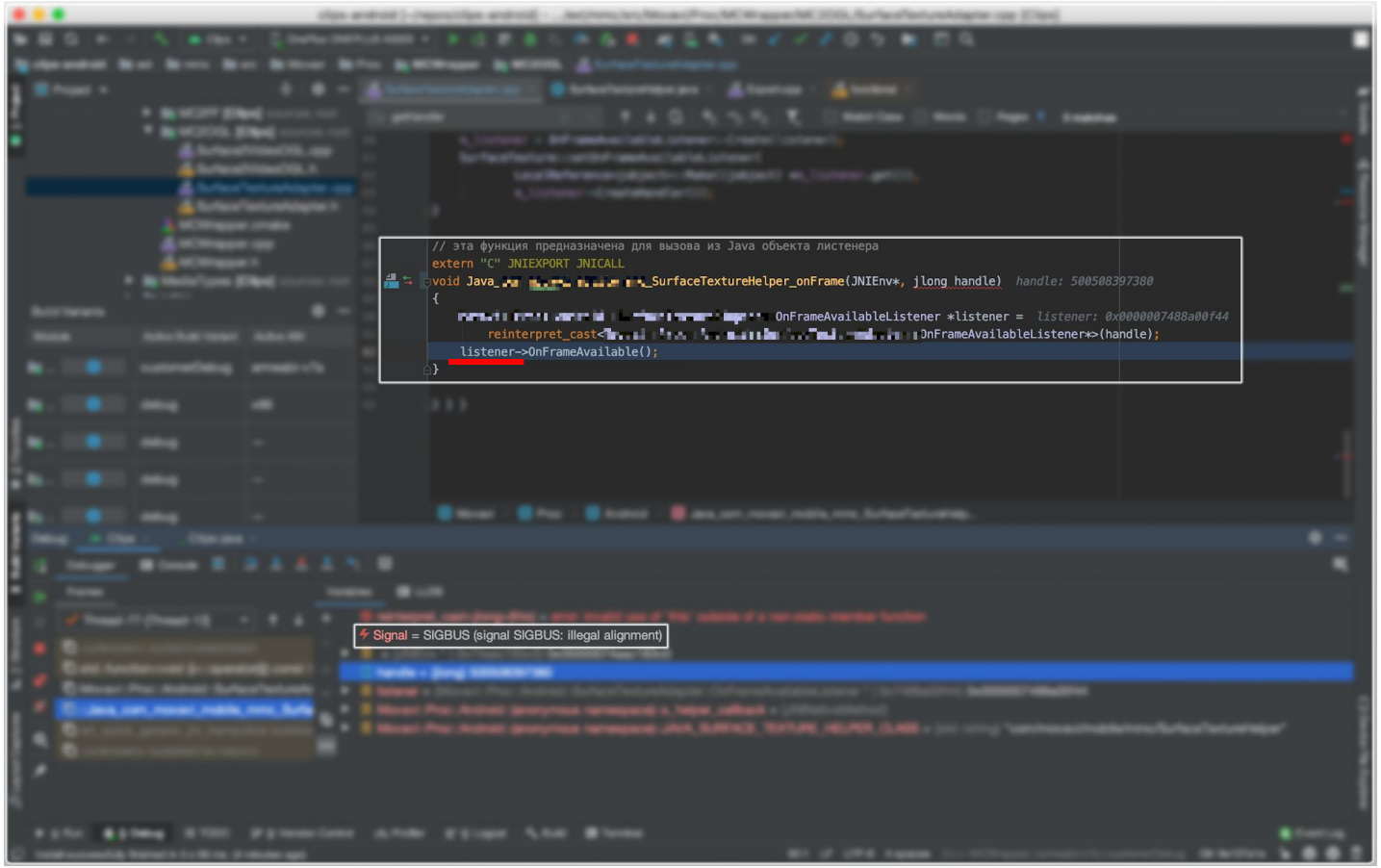
- Появился на свет в первых числах февраля 2017 года в новом файле в большой семье коммитов.
- Но с самого начала пошёл по скользкой дорожке и до последнего скрывался от всех в проде.

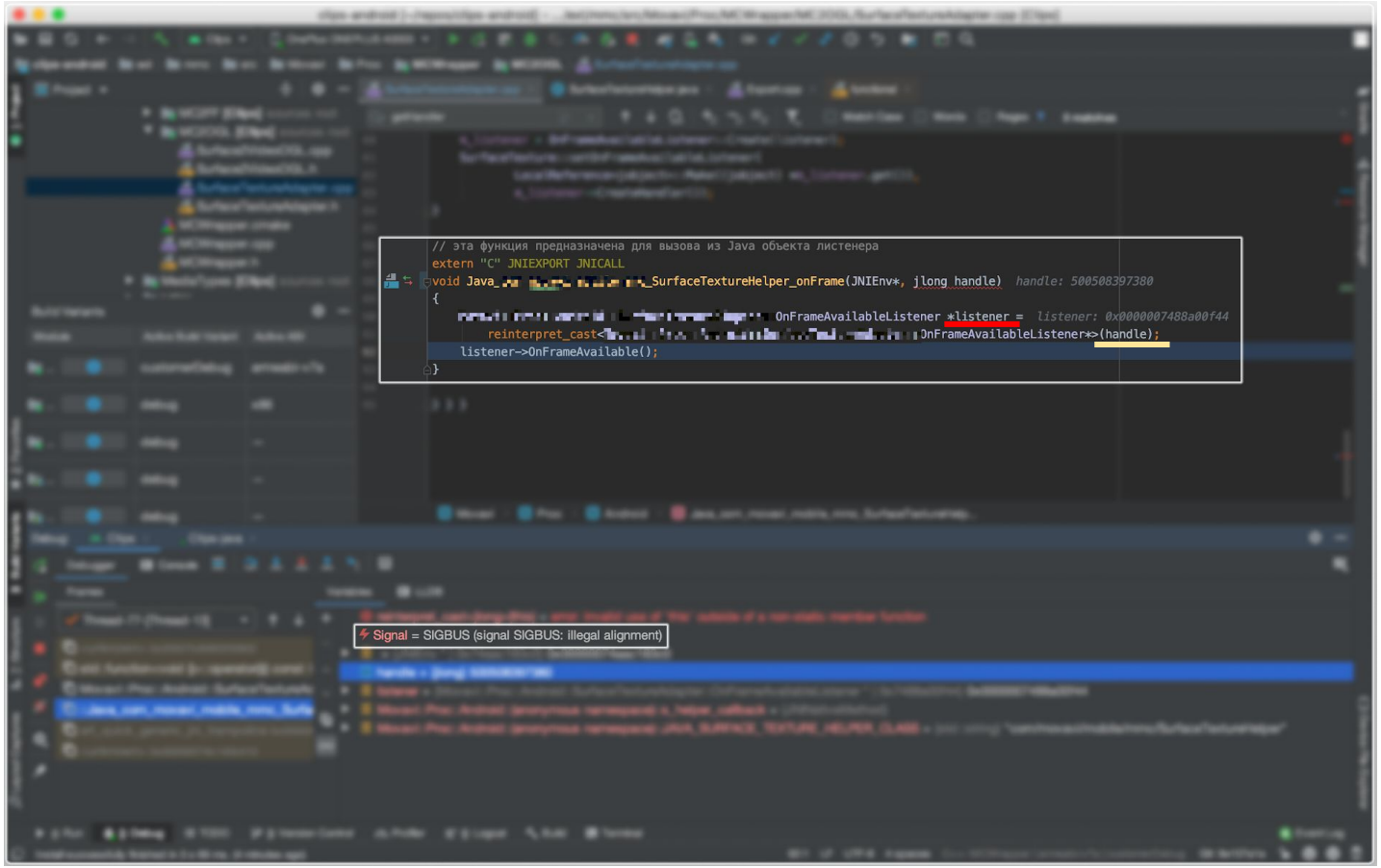
Баг долгожитель

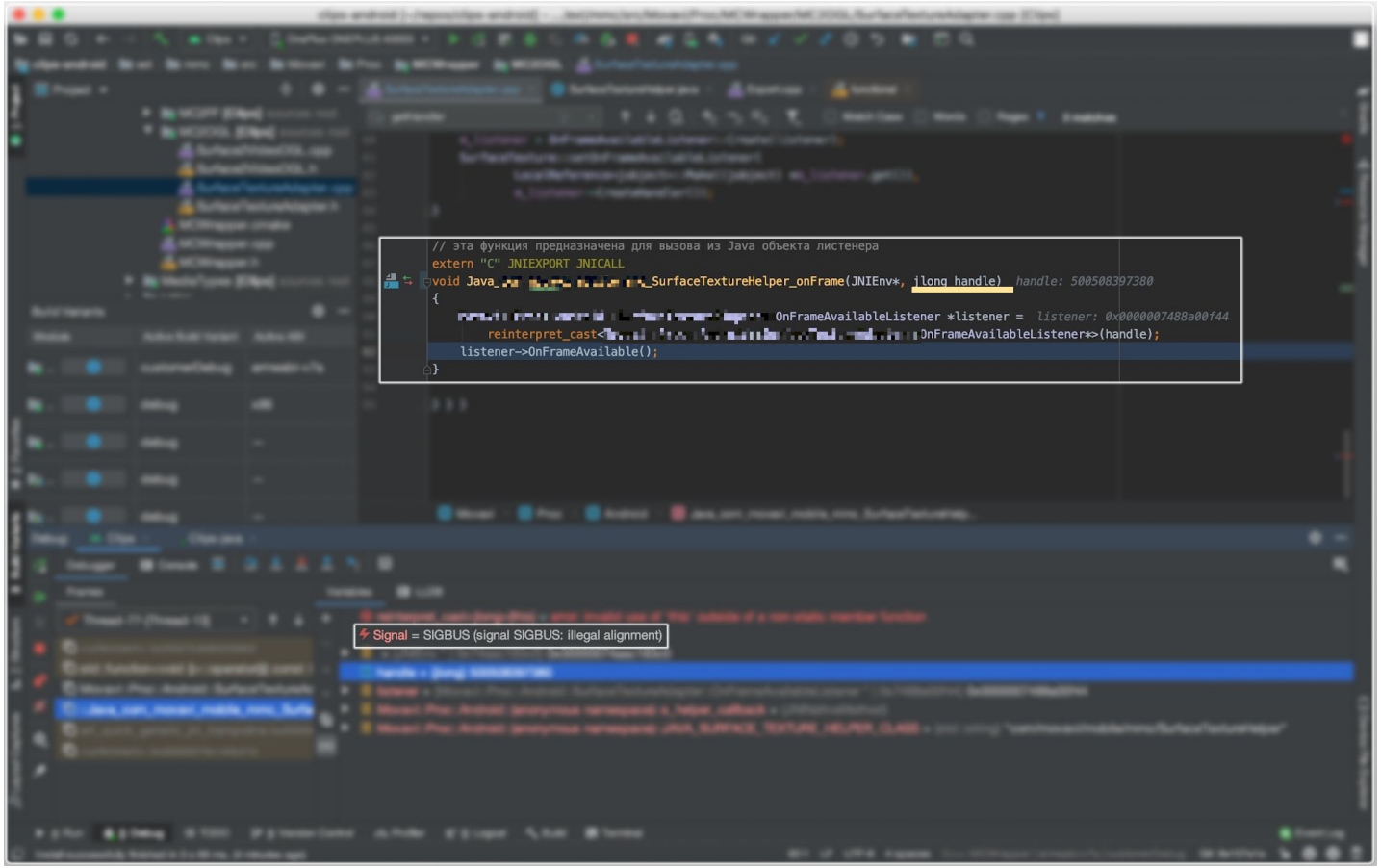


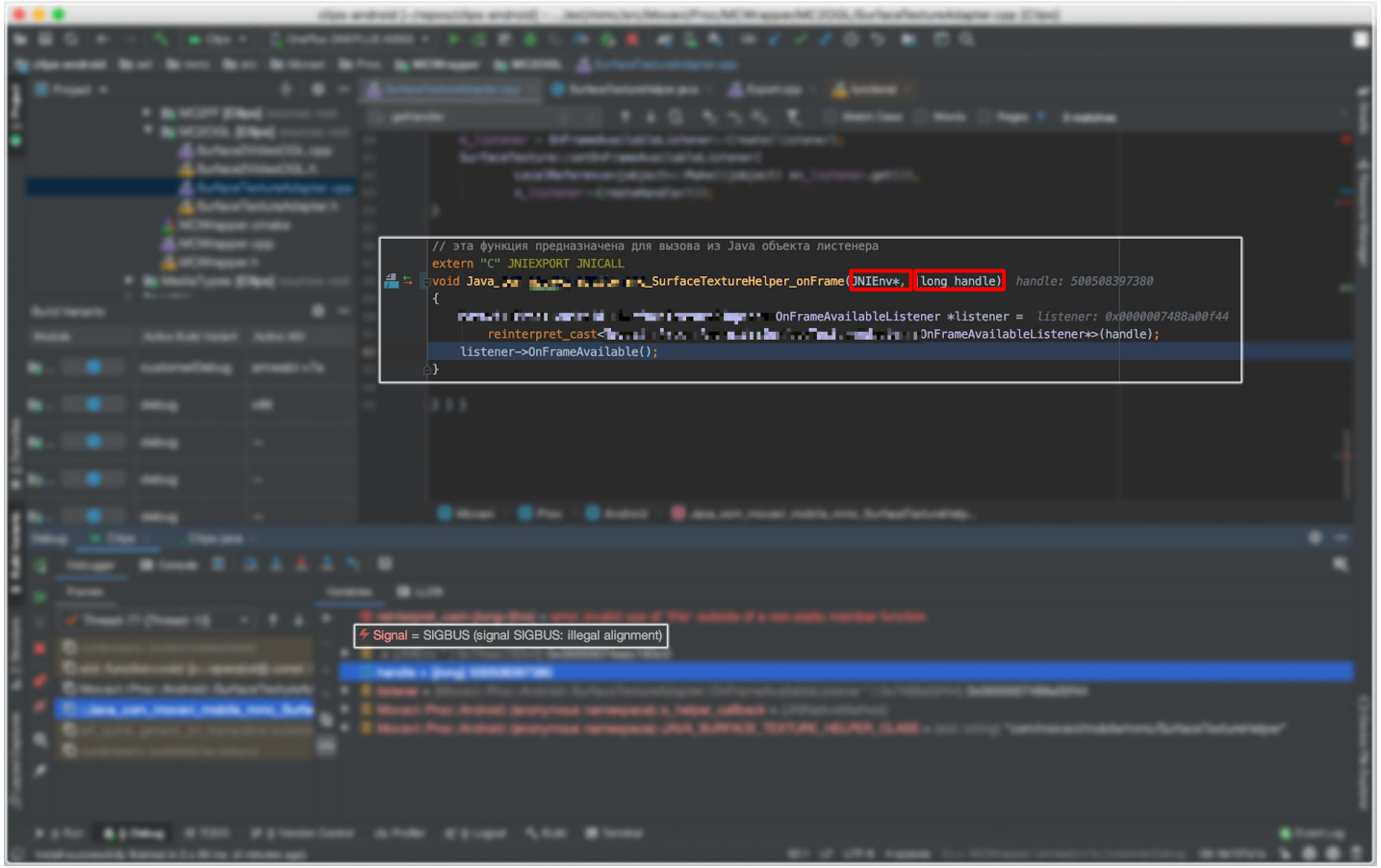

// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ [redacted] _SurfaceTextureHelper_onFrame(JNIEnv*, jlong handle) handle: 500508397380
{
[redacted] OnFrameAvailableListener *listener = listener: 0x0000007488a00f44
reinterpret_cast<[redacted] OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}
Signal = SIGBUS (signal SIGBUS: illegal alignment)

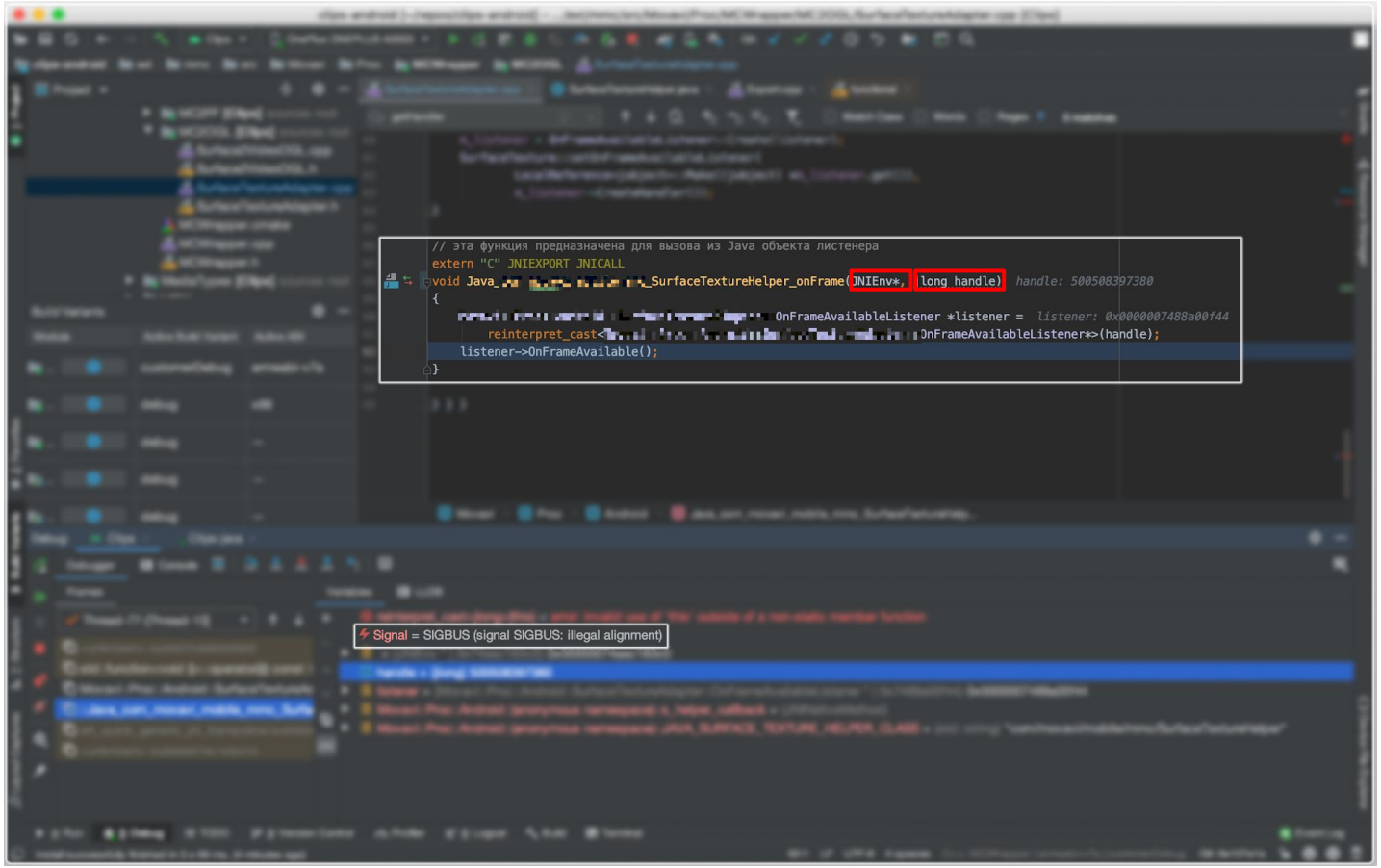

// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, jlong handle) handle: 500508397380
{
OnFrameAvailableListener *listener = listener: 0x0000007488a00f44
reinterpret_cast< OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}
Signal = SIGBUS (signal SIGBUS: illegal alignment)

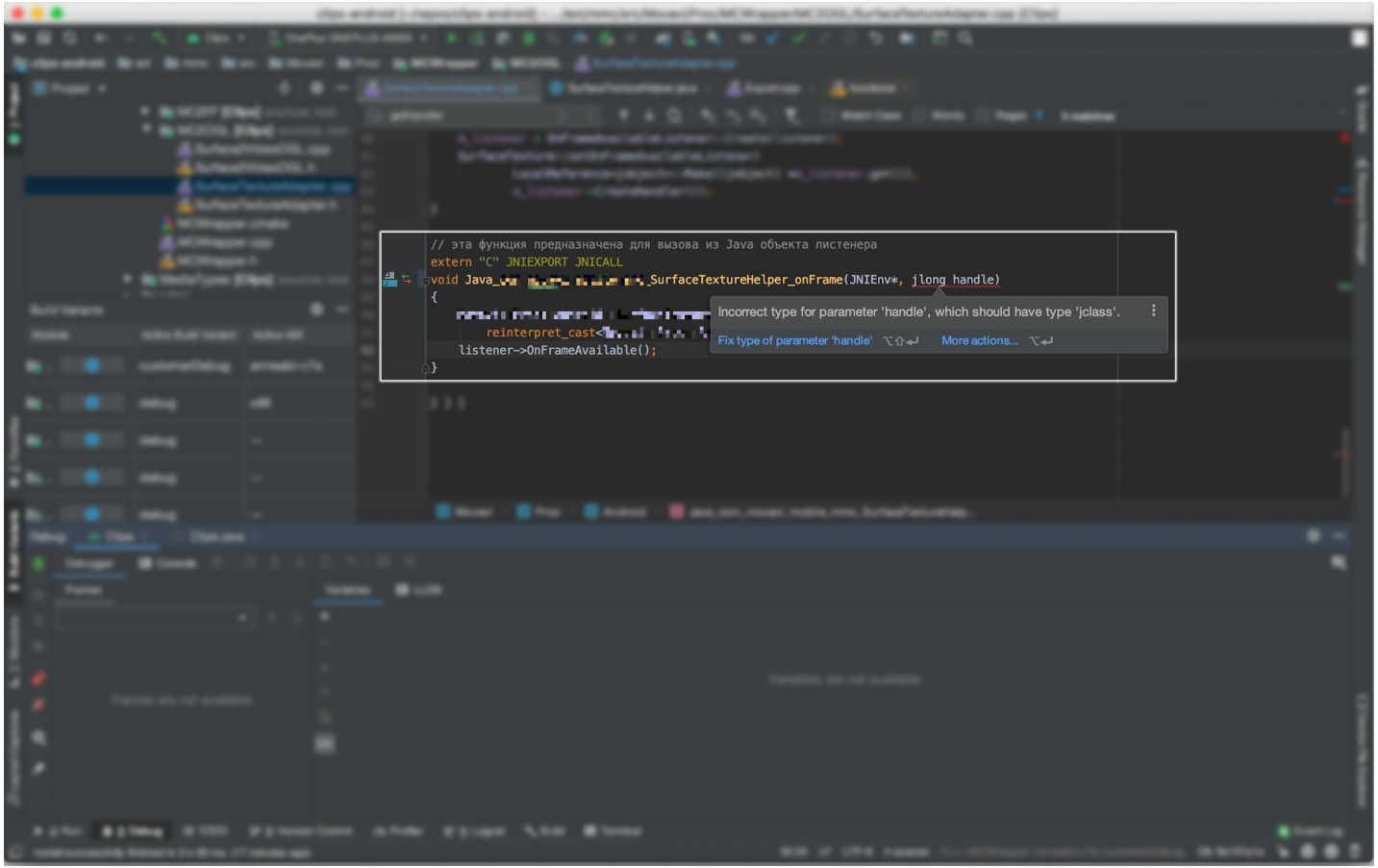

// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, jlong handle) handle: 500508397380
{
OnFrameAvailableListener *listener = listener: 0x0000007488a00f44
reinterpret_cast< OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}
Signal = SIGBUS (signal SIGBUS: illegal alignment)

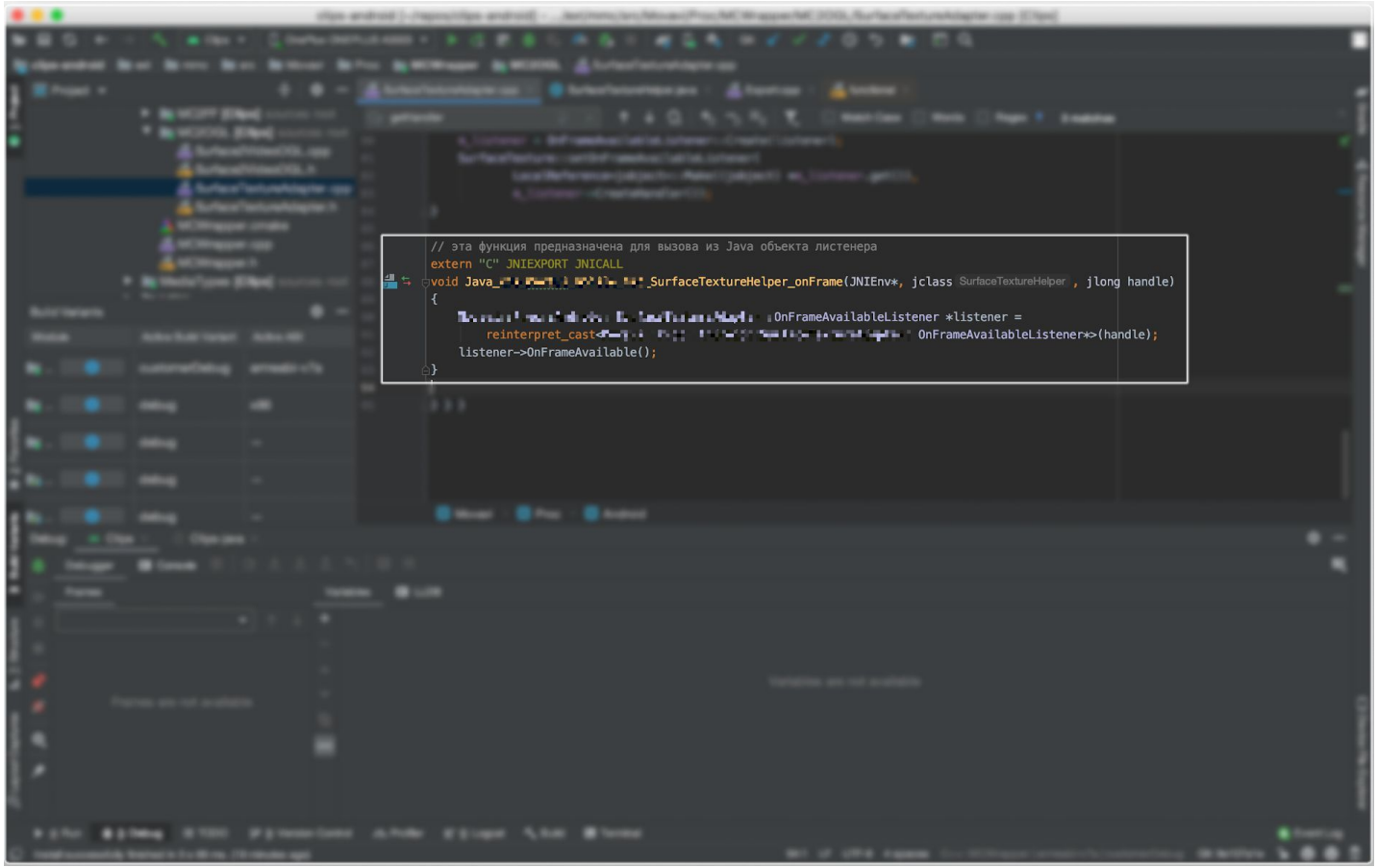

// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, jlong handle) handle: 500508397380
{
OnFrameAvailableListener *listener = listener: 0x0000007488a00f44
reinterpret_cast< OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}
Signal = SIGBUS (signal SIGBUS: illegal alignment)


// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, jlong handle) handle: 500508397380
{
OnFrameAvailableListener *listener = listener: 0x0000007488a00f44
reinterpret_cast< OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}
Signal = SIGBUS (signal SIGBUS: illegal alignment)


// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, long handle)
handle: 500508397380
OnFrameAvailableListener *listener =
listener: 0x0000007488a00f44
reinterpret_cast<
OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
Signal = SIGBUS (signal SIGBUS: illegal alignment)


// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, long handle) handle: 500508397380
{
OnFrameAvailableListener *listener = listener: 0x0000007488a00f44
reinterpret_cast< OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}
Signal = SIGBUS (signal SIGBUS: illegal alignment)


// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_
_SurfaceTextureHelper_onFrame(JNIEnv*, jlong handle)
{
reinterpret_cast<
listener->OnFrameAvailable();
}
Incorrect type for parameter 'handle', which should have type 'jclass'.
Fix type of parameter 'handle'
More actions...


// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, jclass SurfaceTextureHelper , jlong handle)
{
OnFrameAvailableListener *listener =
reinterpret_cast OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}

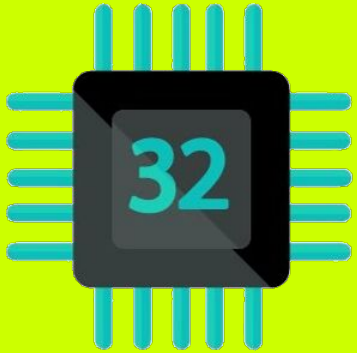
Причины

- Java и C++ код на этапе сборки проекта ничего не знают друг о друге.
- Соглашение о вызовах (calling convention), регламентирует как вызывать методы, описывает порядок и способ передачи аргументов, получения результата и т.п.

Выводы

- Сборка под новую платформу может вскрыть новые, ранее не известные проблемы, и то, что годами работало, внезапно может сразу же сломаться.
- На code review замечают не все проблемы.
- Статические анализаторы кода помогают отловить часть ошибок.
- IDE развиваются, используйте актуальные версии.
- Автоматизация рутинных действий повышает стабильность системы.

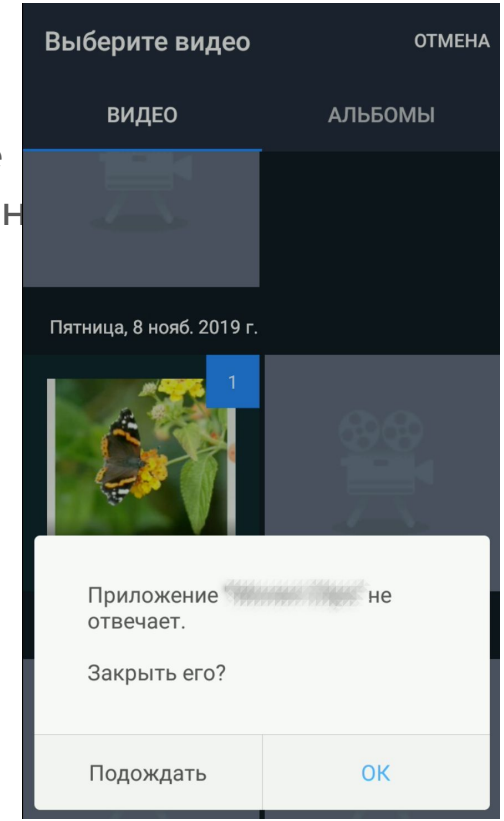
Баг разноразмерный



VS

Баг разноразмерный

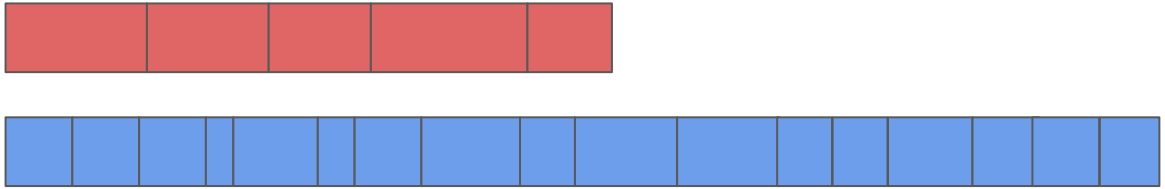
- Один из видеофайлов, который открывался в сборке приложения под armeabi-v7a, перестал открываться н сборке под arm64-v8a.
- Приложение зависает, система детектирует ANR.



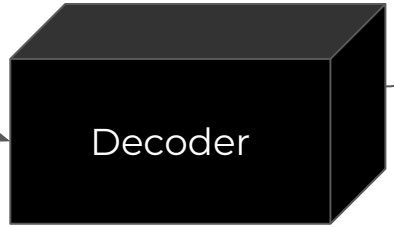
Обработка мультимедиа
Медиаконтейнер

Обработка мультимедиа
Parser, demuxer

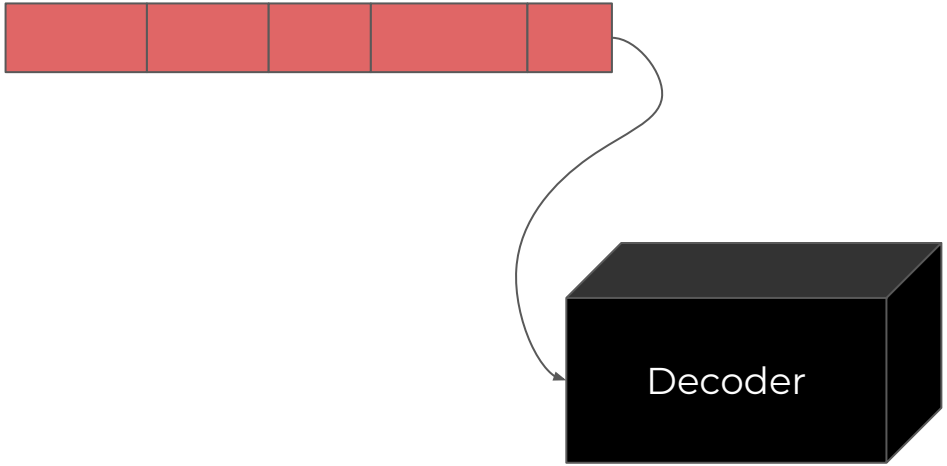
Обработка мультимедиа
Decoder

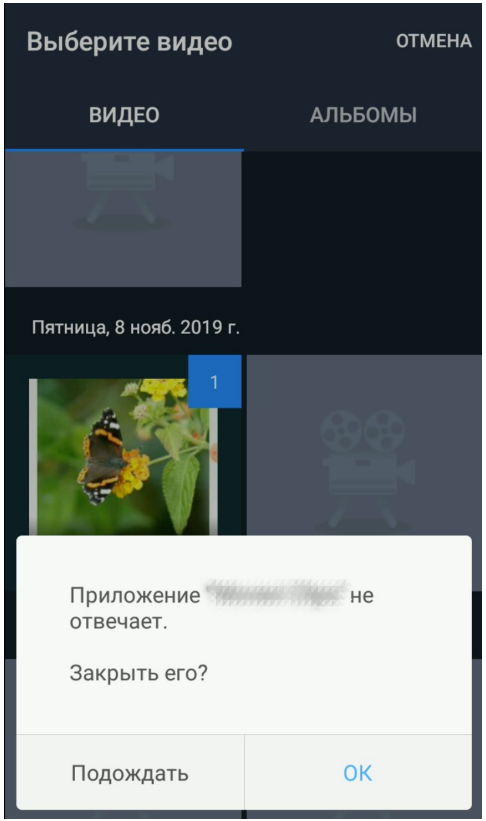


Обработка мультимедиа
Decoder



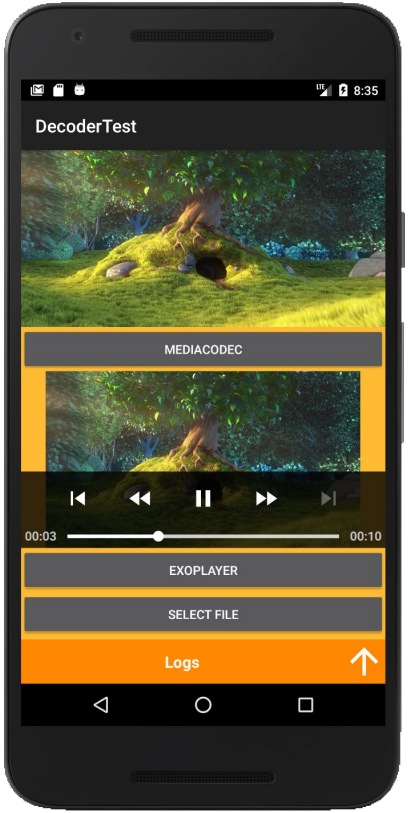
Обработка мультимедиа
Decoder





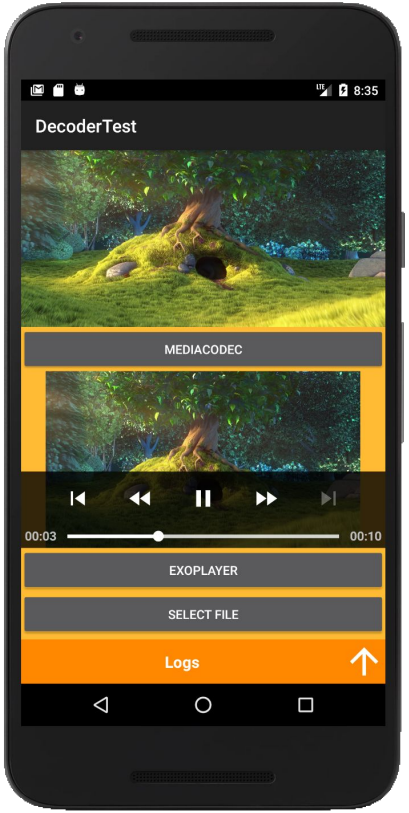
Тестовое приложение

- Kotlin, MediaExtractor + MediaCodec



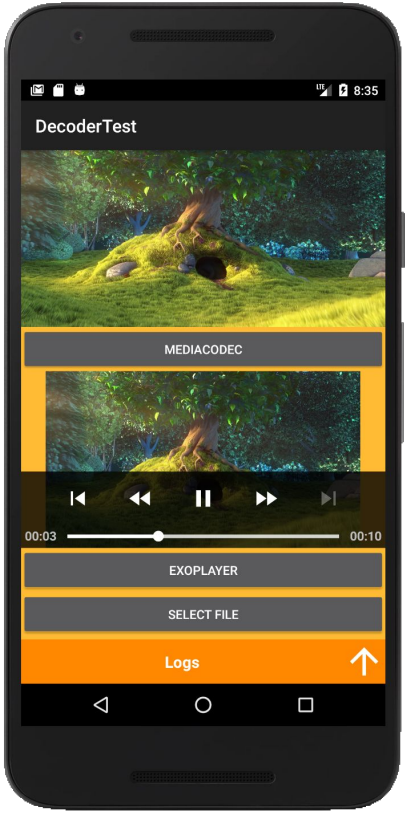
Тестовое приложение + “эталонная” реализация

- Kotlin, MediaExtractor + MediaCodec
- ExoPlayer (декодерует через MediaCodec)



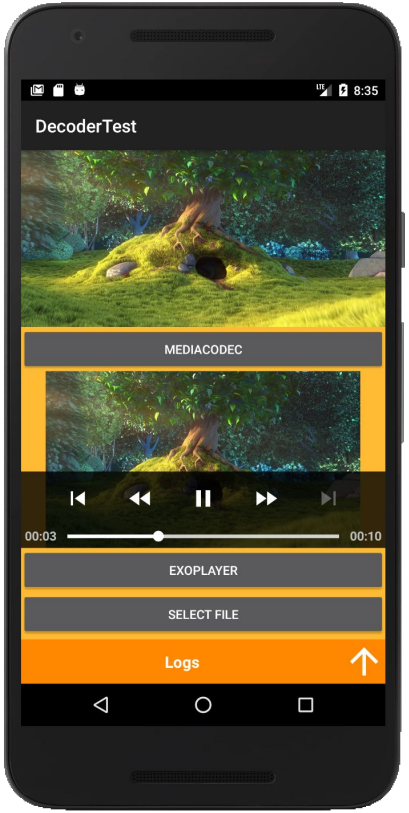
Тестовое приложение + “эталонная” реализация

- Kotlin, MediaExtractor + MediaCodec: НЕ ОК
- ExoPlayer (декодерует через MediaCodec): ОК



Тестовое приложение + “эталонная” реализация

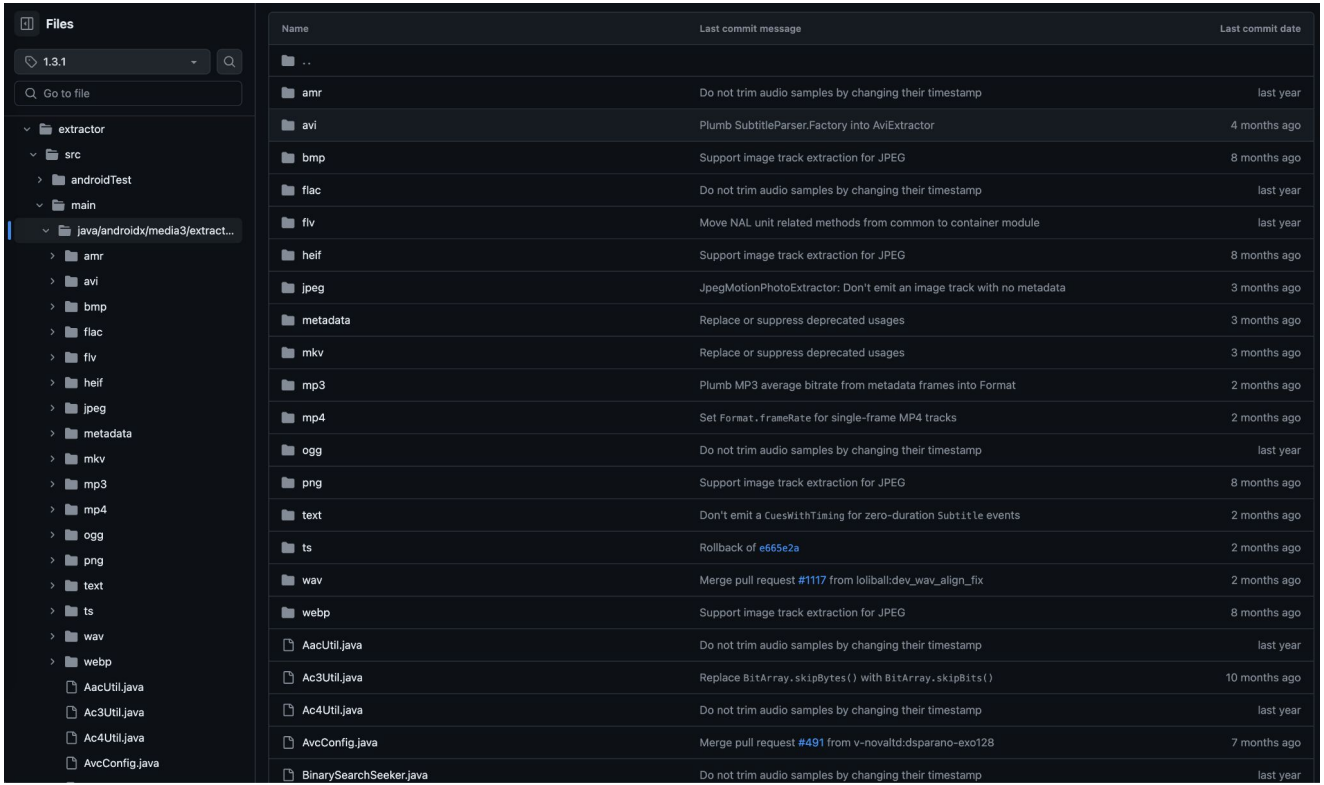
- Kotlin, MediaExtractor + MediaCodec: НЕ ОК
- ExoPlayer (декодерует через MediaCodec): ОК

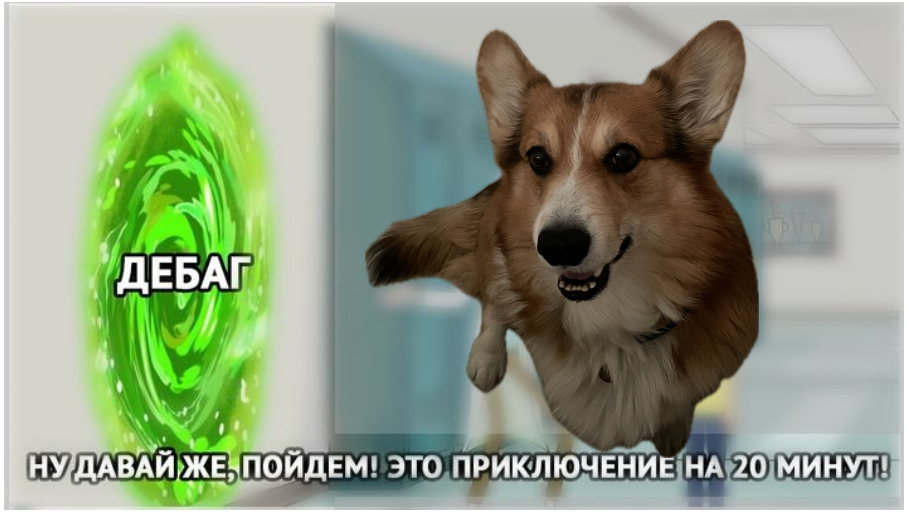


Supported media formats. Video

Format	Encoder	Decoder	Details	File Types Container Formats
H.263	YES	YES	Support for H.263 is optional in Android 7.0+	• 3GPP (.3gp) • MPEG-4 (.mp4) • Matroska (.mkv)
H.264 AVC Baseline Profile (BP)	Android 3.0+	YES		• 3GPP (.3gp) • MPEG-4 (.mp4) • MPEG-TS (.ts, AAC audio only, not seekable, Android 3.0+) • Matroska (.mkv)
H.264 AVC Main Profile (MP)	Android 6.0+	YES	The decoder is required, the encoder is recommended.	
H.265 HEVC		Android 5.0+	Main Profile Level 3 for mobile devices and Main Profile Level 4.1 for Android TV	• MPEG-4 (.mp4) • Matroska (.mkv)
MPEG-4 SP		YES		3GPP (.3gp)
VP8	Android 4.3+	Android 2.3.3+	Streamable only in Android 4.0 and above	• WebM (.webm) • Matroska (.mkv, Android 4.0+)
VP9		Android 4.4+		• WebM (.webm) • Matroska (.mkv)
AV1	Android 14+	Android 10+	Encoder and decoder are mandatory beginning with Android 14.	• MPEG-4 (.mp4) • Matroska (.mkv)

ExoPlayer


Files
1.3.1
Go to file
extractor
src
androidTest
main
java/androidx/media3/extract...
amr
avi
bmp
flac
flv
heif
jpeg
metadata
mkv
mp3
mp4
ogg
png
text
ts
wav
webp
AacUtil.java
Ac3Util.java
Ac4Util.java
AvcConfig.java
Name
Last commit message
Last commit date
..
amr Do not trim audio samples by changing their timestamp last year
avi Plumb SubtitleParser.Factory into AviExtractor 4 months ago
bmp Support image track extraction for JPEG 8 months ago
flac Do not trim audio samples by changing their timestamp last year
flv Move NAL unit related methods from common to container module last year
heif Support image track extraction for JPEG 8 months ago
jpeg JpegMotionPhotoExtractor: Don't emit an image track with no metadata 3 months ago
metadata Replace or suppress deprecated usages 3 months ago
mkv Replace or suppress deprecated usages 3 months ago
mp3 Plumb MP3 average bitrate from metadata frames into Format 2 months ago
mp4 Set Format.frameRate for single-frame MP4 tracks 2 months ago
ogg Do not trim audio samples by changing their timestamp last year
png Support image track extraction for JPEG 8 months ago
text Don't emit a CuesWithTiming for zero-duration Subtitle events 2 months ago
ts Rollback of e665e2a 2 months ago
wav Merge pull request #1117 from loliball:dev_wav_align_fix 2 months ago
webp Support image track extraction for JPEG 8 months ago
AacUtil.java Do not trim audio samples by changing their timestamp last year
Ac3Util.java Replace BitArray.skipBytes() with BitArray.skipBits() 10 months ago
Ac4Util.java Do not trim audio samples by changing their timestamp last year
AvcConfig.java Merge pull request #491 from v-novaltd:dsparano-exo128 7 months ago
BinarySearchSeeker.java Do not trim audio samples by changing their timestamp last year


ДЕБАГ
НУ ДАВАЙ ЖЕ, ПОЙДЕМ! ЭТО ПРИКЛЮЧЕНИЕ НА 20 МИНУТ!

А-А-А-А-А!!!
Что тут происходит?!

	A	B	C	D	E	F	G	H
1								
2	x32			x64			EXO	
3								
4	[1] size 722			[1] size 738			Size 1585152	
5	inputBuff[0]		0	inputBuff[0]		0	inputBuff[0]	0
6	inputBuff[1]		0	inputBuff[1]		0	inputBuff[1]	0
7	inputBuff[2]		0	inputBuff[2]		0	inputBuff[2]	0
8	inputBuff[3]		1	inputBuff[3]		1	inputBuff[3]	1
9				inputBuff[4]		0		
10				inputBuff[5]		0		
11				inputBuff[6]		0		
12				inputBuff[7]		0		
13	inputBuff[4]		6	inputBuff[8]		6	inputBuff[4]	6
14	inputBuff[5]		5	inputBuff[9]		5	inputBuff[5]	5
15	inputBuff[6]		-1	inputBuff[10]		-1	inputBuff[6]	-1
16	inputBuff[7]		-1	inputBuff[11]		-1	inputBuff[7]	-1
17	inputBuff[8]		64	inputBuff[12]		64	inputBuff[8]	64
18	inputBuff[9]		-36	inputBuff[13]		-36	inputBuff[9]	-36
19	inputBuff[10]		69	inputBuff[14]		69	inputBuff[10]	69
20	inputBuff[11]		-23	inputBuff[15]		-23	inputBuff[11]	-23
21	inputBuff[12]		-67	inputBuff[16]		-67	inputBuff[12]	-67
22	inputBuff[13]		-26	inputBuff[17]		-26	inputBuff[13]	-26
23	inputBuff[14]		-39	inputBuff[18]		-39	inputBuff[14]	-39
24	inputBuff[15]		72	inputBuff[19]		72	inputBuff[15]	72
25	inputBuff[16]		-73	inputBuff[20]		-73	inputBuff[16]	-73
26	inputBuff[17]		-106	inputBuff[21]		-106	inputBuff[17]	-106
27	inputBuff[18]		44	inputBuff[22]		44	inputBuff[18]	44
28	inputBuff[19]		-40	inputBuff[23]		-40	inputBuff[19]	-40
29	inputBuff[20]		32	inputBuff[24]		32	inputBuff[20]	32
30	inputBuff[21]		-39	inputBuff[25]		-39	inputBuff[21]	-39
31	inputBuff[22]		35	inputBuff[26]		35	inputBuff[22]	35
32	inputBuff[23]		-18	inputBuff[27]		-18	inputBuff[23]	-18
33	inputBuff[24]		-17	inputBuff[28]		-17	inputBuff[24]	-17

Android SDK. MediaExtractor

```
/**
 * Retrieve the current encoded sample and store it in the byte buffer
 * starting at the given offset.
 * <p>
 * <b>Note:</b>As of API 21, on success the position and limit of
 * {@code byteBuf} is updated to point to the data just read.
 * @param byteBuf the destination byte buffer
 * @return the sample size (or -1 if no more samples are available).
 */
public native int readSampleData(@NonNull ByteBuffer byteBuf, int offset);
```

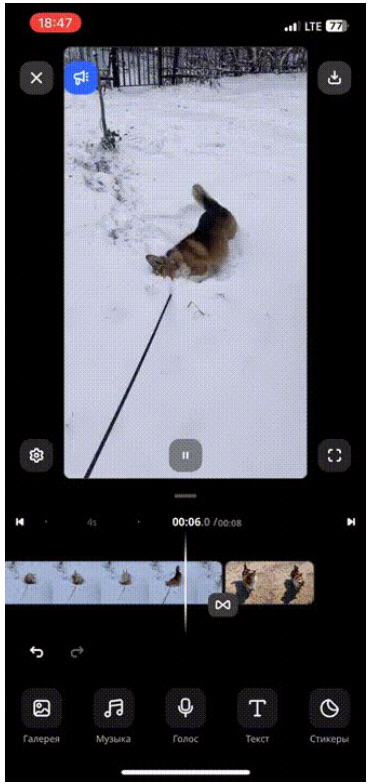
Выводы

- Кроссплатформенные решения, которые, например, реализуются на основе JVM, всё равно могут использовать код, специфичный для платформы, архитектуры и т.п. В этот момент может происходить что угодно и влиять на поведение системы в целом. Последствия могут не иметь отражения в документации, т.к. о них никто не задумывался и/или с ними не сталкивались.
- Используйте простые реализации (прототипирование) для проверки работоспособности решений.
- Обкатайте решение отдельно от общей системы, убедитесь в его стабильности и уже потом переносите в приложение.
- Используйте эталонные решения как часть тест-системы, должна быть возможность сравнить реализации между собой при внесении изменений.
- Необходимо иметь базовые представления о работе ОС, VM и всего того, что обеспечивает работоспособность вашего кода.

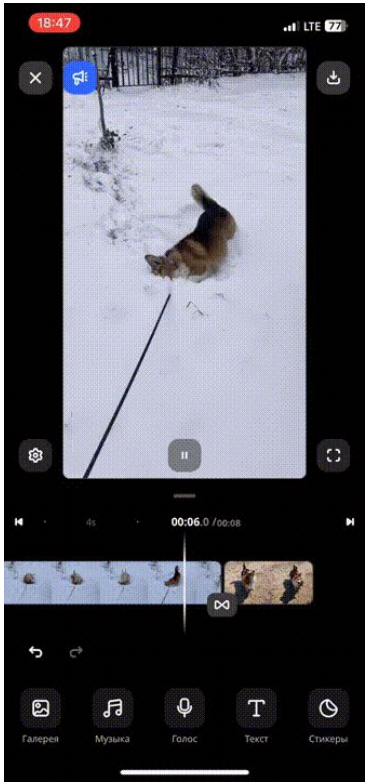
Баг мимикрирующий

Одновременная работа с несколькими видео

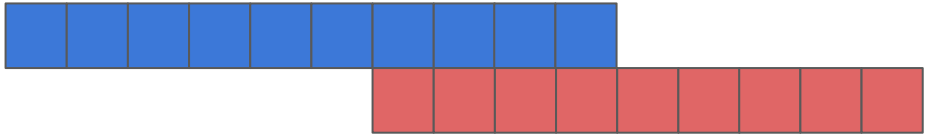
- применение переходов
- picture in picture
- анимированные стикеры

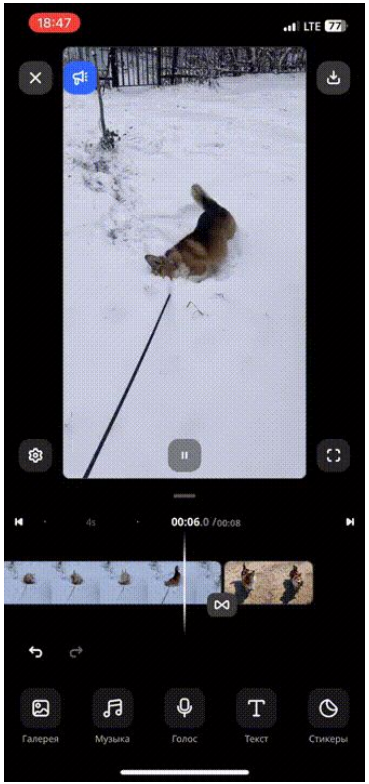


Видеопереход

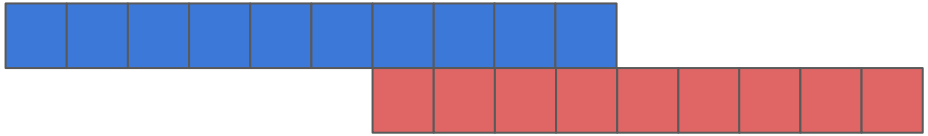


Видеопереход

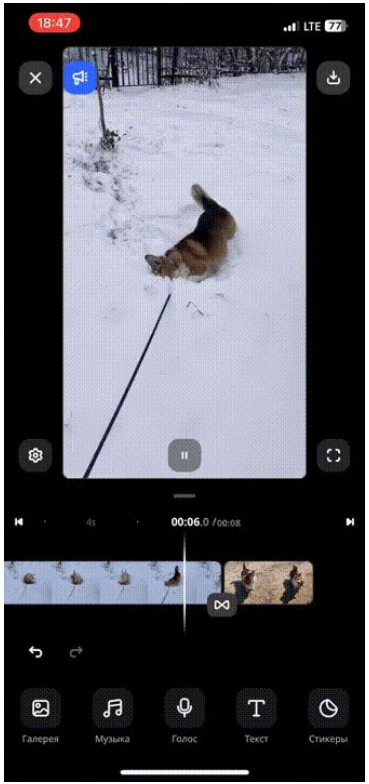




Видеопереход

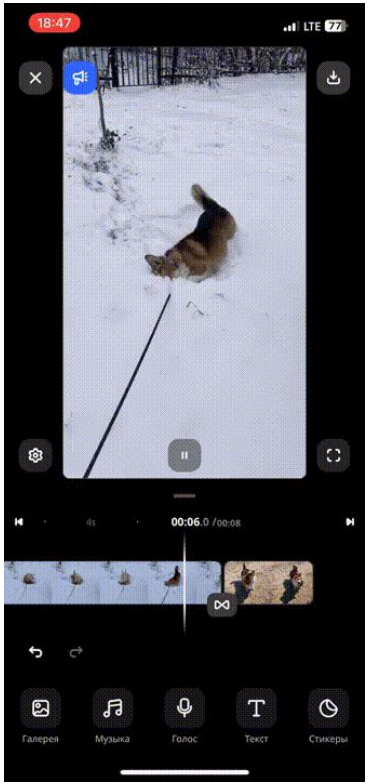


- понизить разрешение

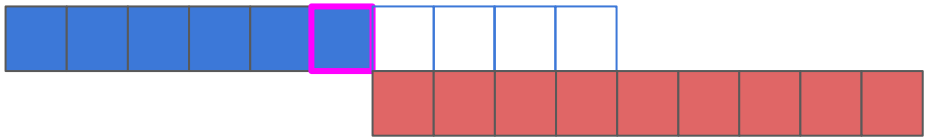


Видеопереход

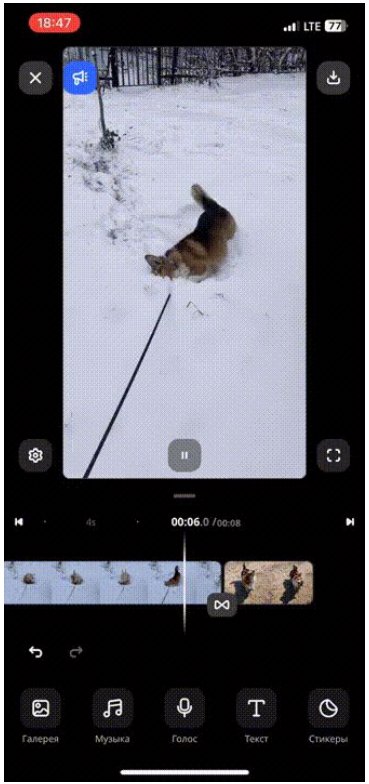
- понизить разрешение
- поменять реализацию декодера



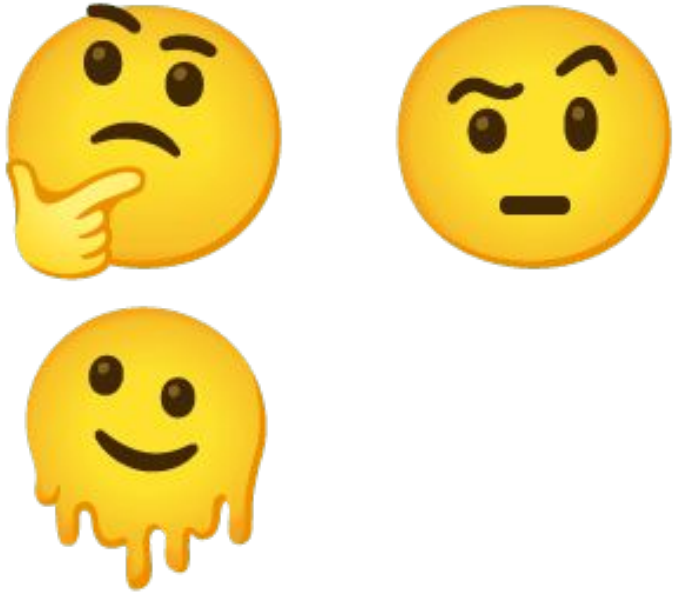
Видеопереход



- понизить разрешение
- поменять реализацию декодера
- “стоп”-кадр



Поиск причин



Поиск причин

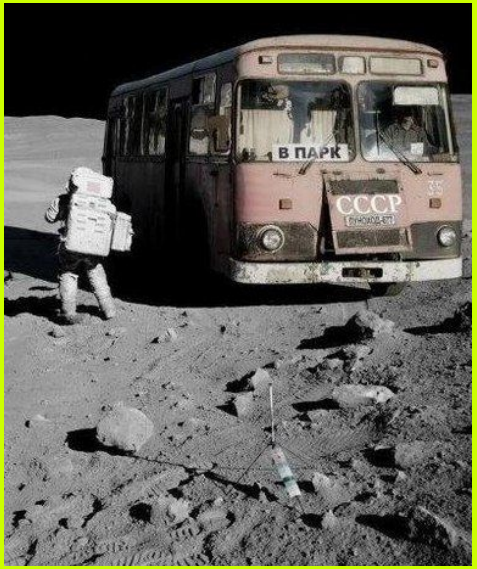
en failed: library "libiconv.so" not found

Выводы

- Знания о внутреннем устройстве системы помогут вам сформулировать гипотезы для проверки.
- Когда больше нет вариантов - не отчаивайтесь, начинайте снова, ищите там, где раньше думали, что всё в порядке.
- Информативные логи с правильным уровнем (INFO, DEBUG, WARNING, ERROR) помогут найти аномалии в системе и локализовать их.
- Автоматизируйте проверки, чтобы в будущем не делать это вручную.

Баг красный ...

Баг красный ...

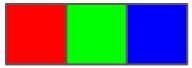


Баг красный песец

Pixel format. RGB

Pixel format. RGB

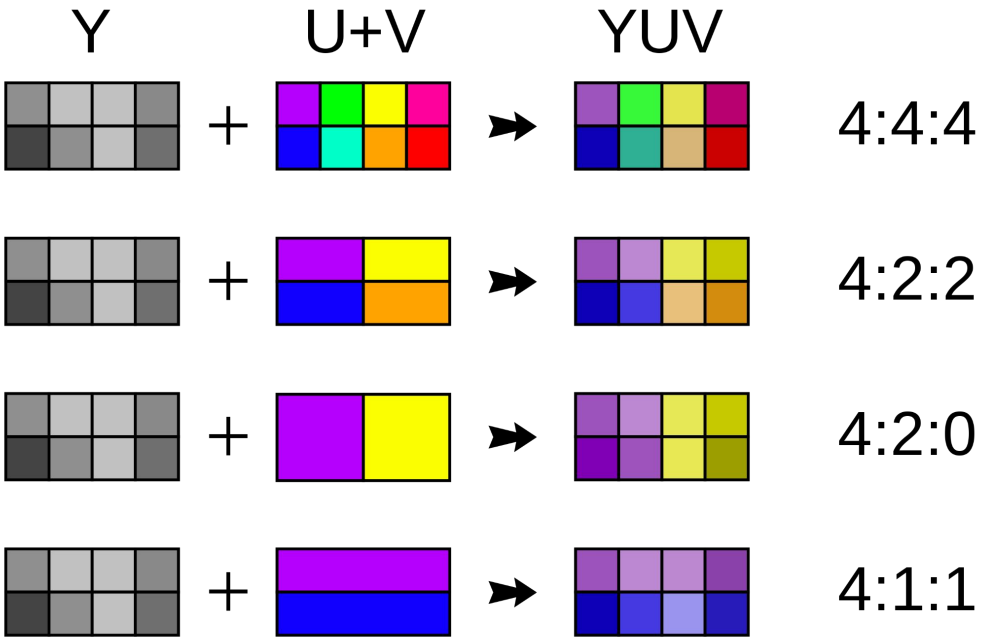
HD, 720 x 1280 = 921600 px

24bit RGB 

≈ 2,6 МБ (1 кадр)

Pixel format. YUV

Pixel format. YUV



Pixel format

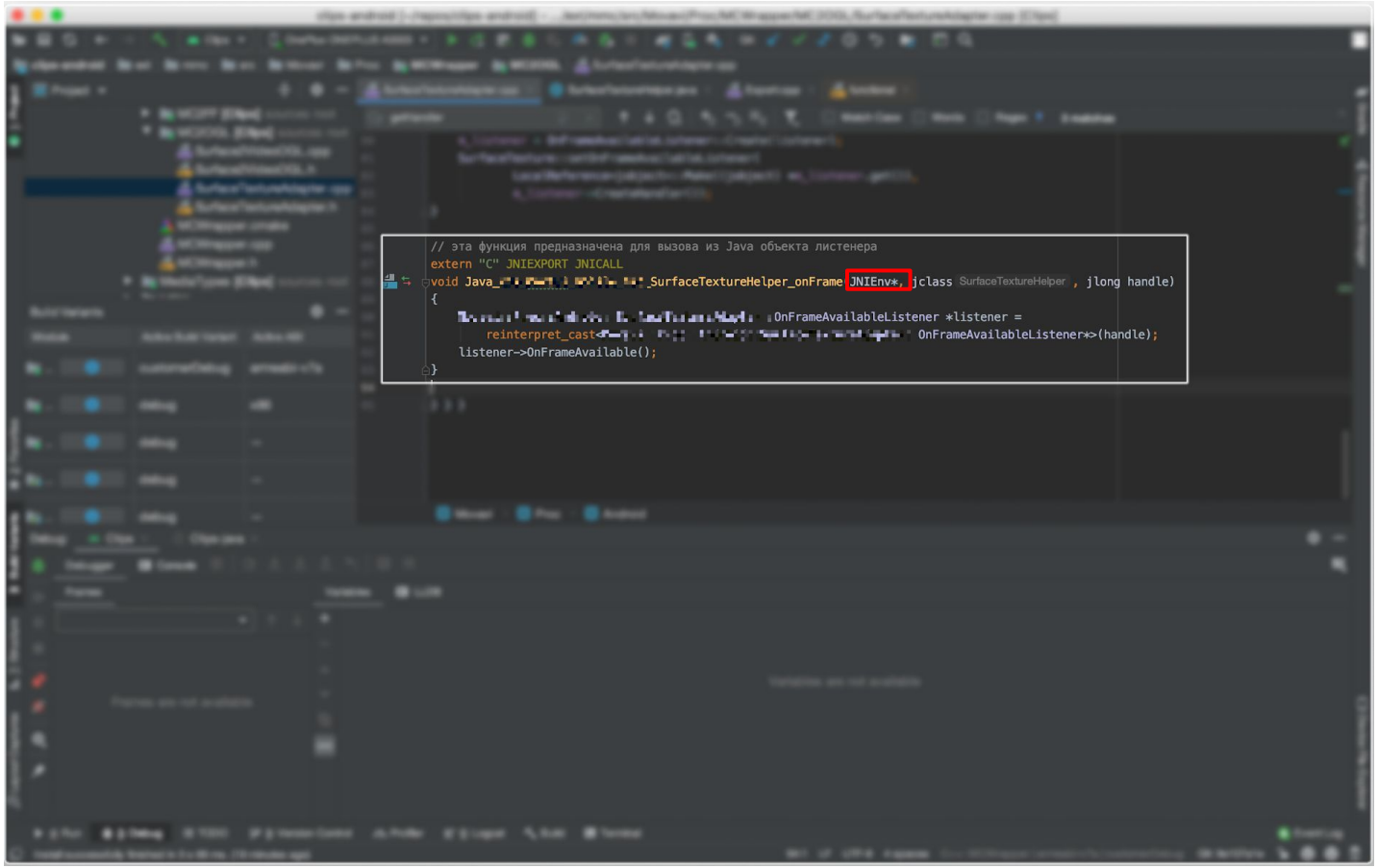
```
// Эффект ожидает цветовое пространство RGBA
const auto rgbaFrame = srcFrame->ConvertToRGBA(context);
```

Выводы

- Не смешивайте несколько задач в одну.
- Невозможно проверить все возможные комбинации настроек в приложении, тем более вручную.
- Всегда проверяйте граничные условия!
- Проектируя систему, которая имеет ограничения на формат данных с которыми она работает обеспечьте их предварительную подготовку, либо проверку на соответствие требованиям, а лучше и то, и другое.
- Проверяйте решения. Делайте это постоянно. Автоматизируйте процесс.

Жадный баг

JNI ERROR (app bug): weak global reference table overflow (max=51200)


// эта функция предназначена для вызова из Java объекта листенера
extern "C" JNIEXPORT JNICALL
void Java_ _SurfaceTextureHelper_onFrame(JNIEnv*, jclass SurfaceTextureHelper , jlong handle)
{
OnFrameAvailableListener *listener =
reinterpret_cast OnFrameAvailableListener*>(handle);
listener->OnFrameAvailable();
}

JavaVM

jni.h

```
/*
 * Prototypes for functions exported by loadable shared libs.  These are
 * called by JNI, not provided by JNI.
 */
JNIEXPORT jint JNI_OnLoad(JavaVM* vm, void* reserved);
JNIEXPORT void JNI_OnUnload(JavaVM* vm, void* reserved);
```

JavaVM → JNIEnv

jni.h

```
struct _JavaVM {
    const struct JNIInvokeInterface* functions;

#if defined(__cplusplus)
    jint DestroyJavaVM()
    { return functions->DestroyJavaVM(this); }
    jint AttachCurrentThread(JNIEnv** p_env, void* thr_args)
    { return functions->AttachCurrentThread(this, p_env, thr_args); }
    jint DetachCurrentThread()
    { return functions->DetachCurrentThread(this); }
    jint GetEnv(void** env, jint version)
    { return functions->GetEnv(this, env, version); }
    jint AttachCurrentThreadAsDaemon(JNIEnv** p_env, void* thr_args)
    { return functions->AttachCurrentThreadAsDaemon(this, p_env, thr_args); }
#endif /*__cplusplus*/
};
```

JavaVM → JNIEnv

jni.h

```
struct _JavaVM {
    const struct JNIInvokeInterface* functions;

#if defined(__cplusplus)
    jint DestroyJavaVM()
    { return functions->DestroyJavaVM(this); }
    jint AttachCurrentThread(JNIEnv** p_env, void* thr_args)
    { return functions->AttachCurrentThread(this, p_env, thr_args); }
    jint DetachCurrentThread()
    { return functions->DetachCurrentThread(this); }
    jint GetEnv(void** env, jint version)
    { return functions->GetEnv(this, env, version); }
    jint AttachCurrentThreadAsDaemon(JNIEnv** p_env, void* thr_args)
    { return functions->AttachCurrentThreadAsDaemon(this, p_env, thr_args); }
#endif /*__cplusplus*/
};
```

JNI ERROR

```
E  JNI ERROR (app bug): weak global reference table overflow (max=51200)
E  Failed adding to JNI weak global ref table (51200 entries)
E  VM aborting
```

JavaVM (references)

jni.h

```
jweak NewWeakGlobalRef(jobject obj)
{ return functions->NewWeakGlobalRef(this, obj); }
```

JavaVM (references)

```
java_vm_ext.cc:740] JNI ERROR (app bug): weak global reference table overflow (max=51200)weak global reference table dump:
java_vm_ext.cc:740]   Last 10 entries (of 51200):
java_vm_ext.cc:740]     51199: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51198: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51197: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51196: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51195: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51194: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51193: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51192: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51191: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]     51190: 0x12f57020 com.[illegible]
java_vm_ext.cc:740]   Summary:
java_vm_ext.cc:740]     51148 of com.[illegible] (1 unique instances)
java_vm_ext.cc:740]        35 of java.lang.DexCache (35 unique instances)
java_vm_ext.cc:740]         9 of java.lang.Class (9 unique instances)
java_vm_ext.cc:740]         7 of dalvik.system.PathClassLoader (5 unique instances)
java_vm_ext.cc:740]         1 of java.lang.BootClassLoader
```

JavaVM (references)

java_vm_ext.cc

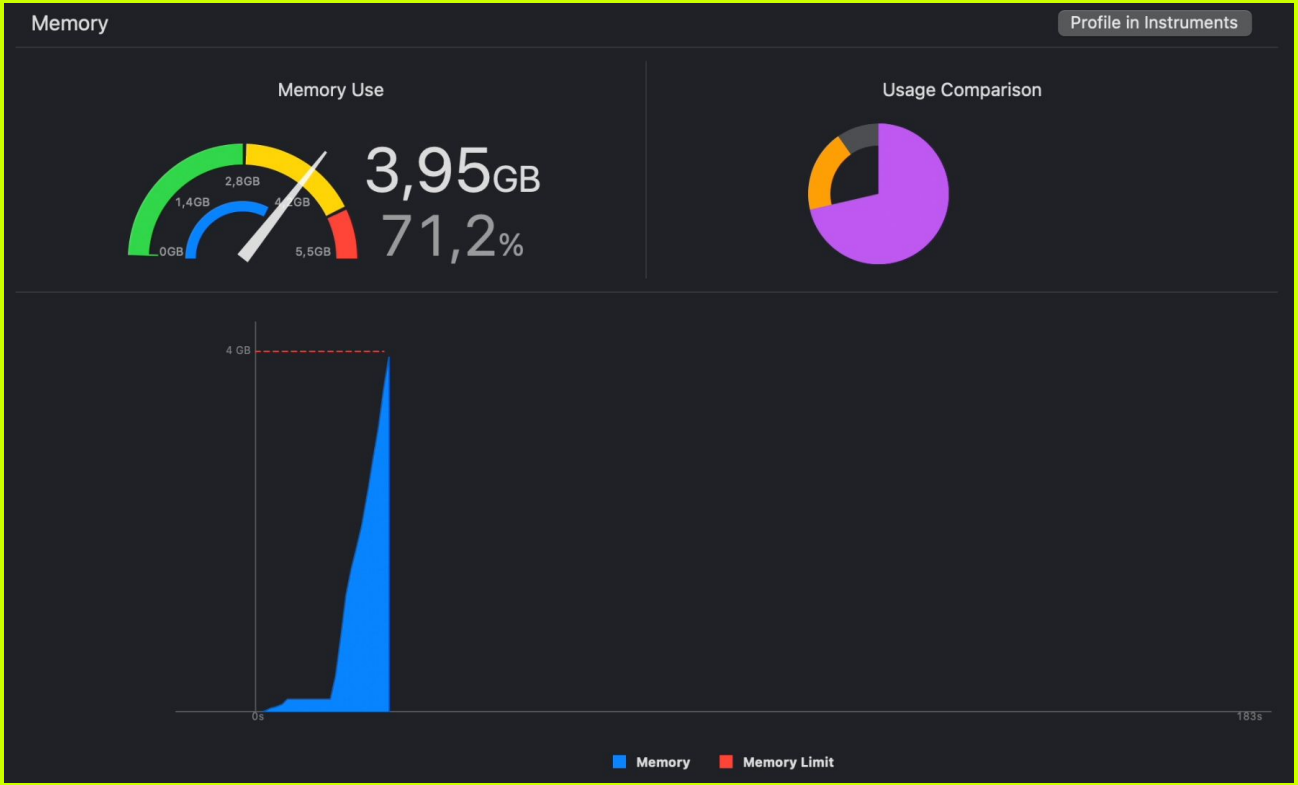
```
62  // Maximum number of global references (must fit in 16 bits).
63  static constexpr size_t kGlobalsMax = 51200;
64
65  // Maximum number of weak global references (must fit in 16 bits).
66  static constexpr size_t kWeakGlobalsMax = 51200;
```

Выводы

- JNI и Android NDK - отдельный мир на стыке C++ и Android разработки.

Баг беспамятный

Баг беспамятный

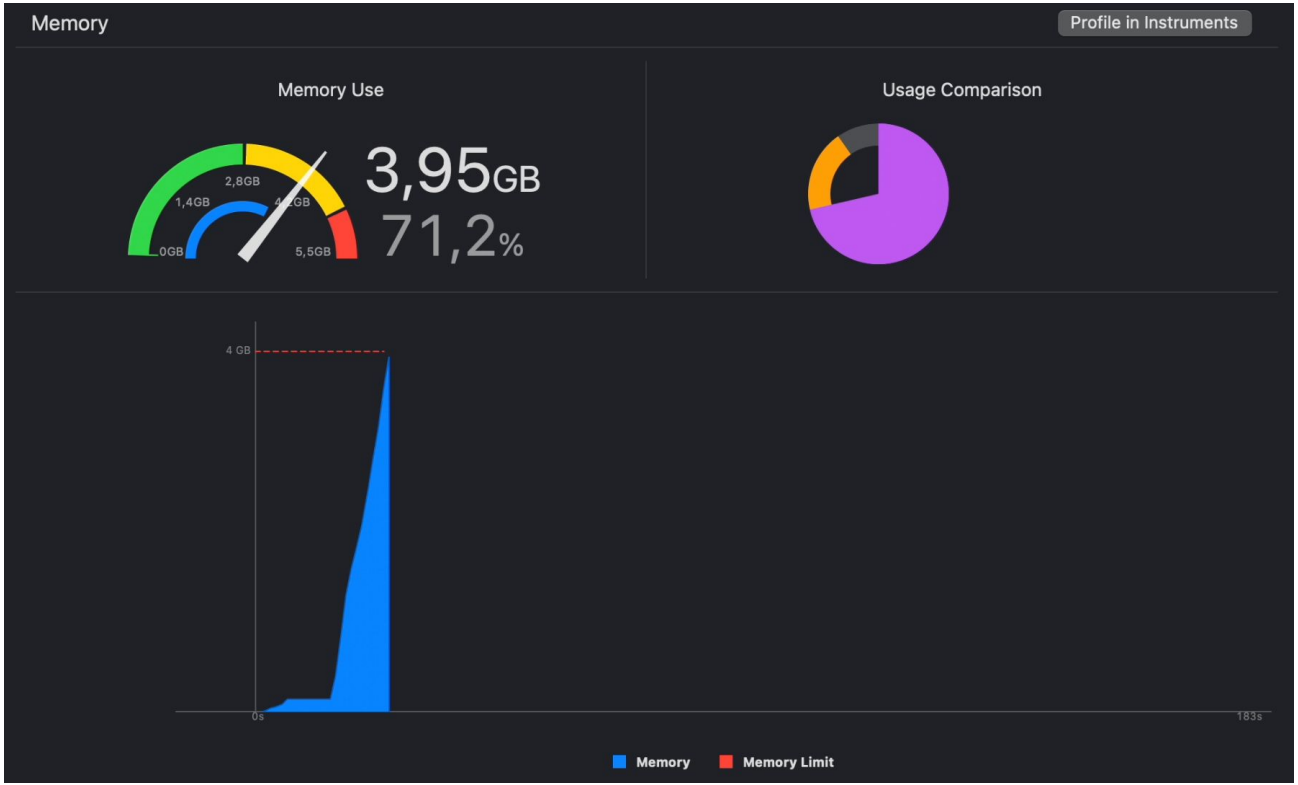


Системы управления памятью

Системы управления памятью

Системы управления памятью

Системы управления памятью

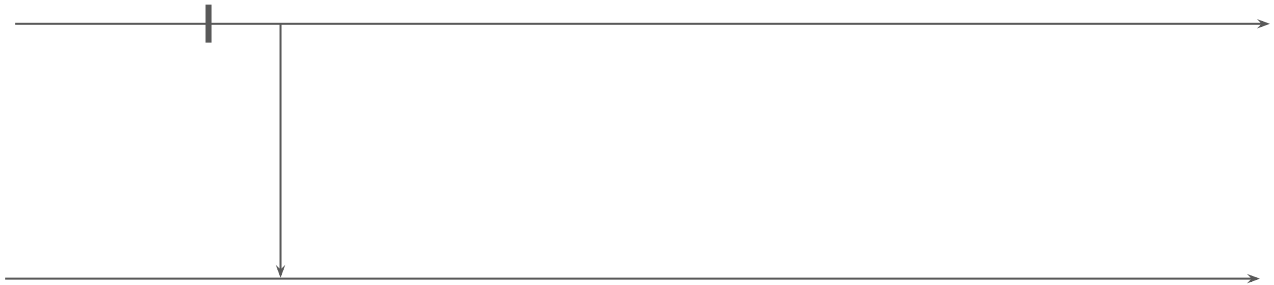


Модификация кадров

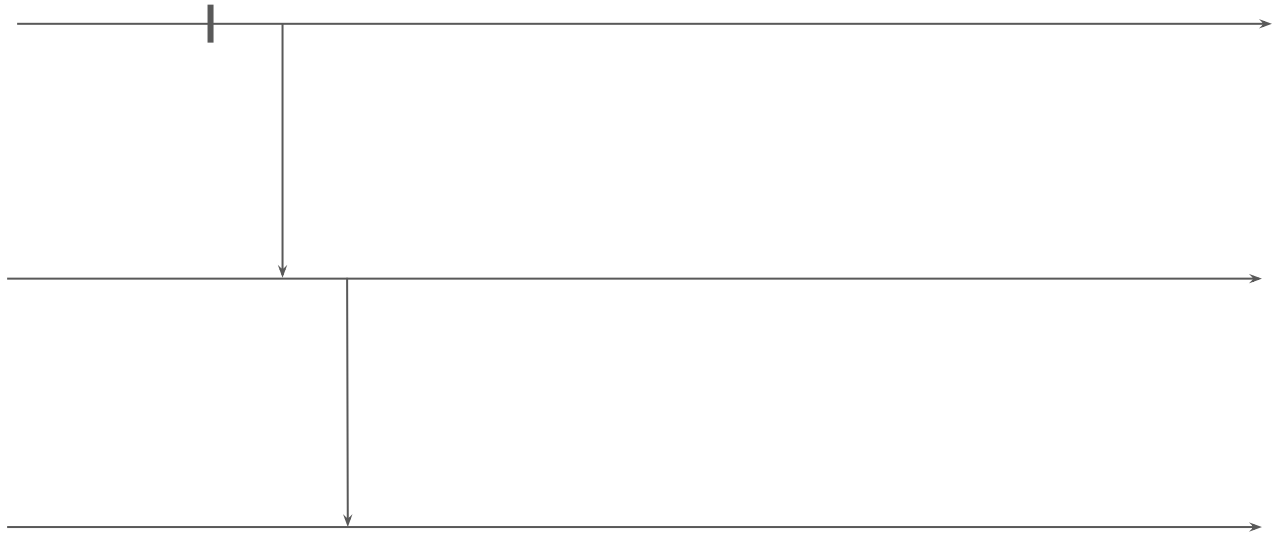
Модификация кадров

Модификация кадров

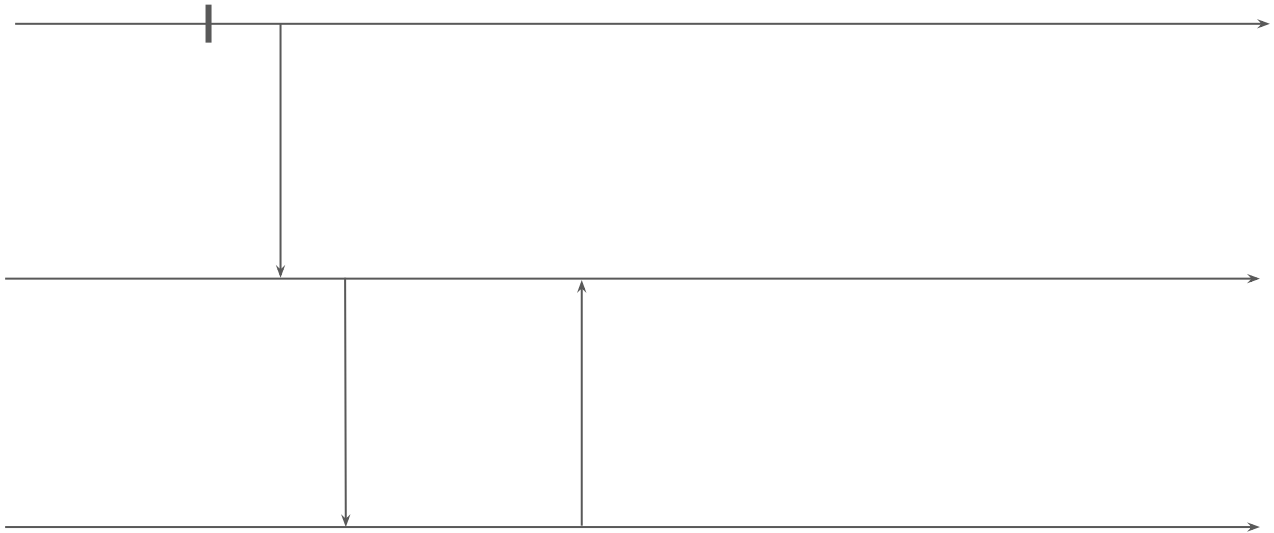
Модификация кадров



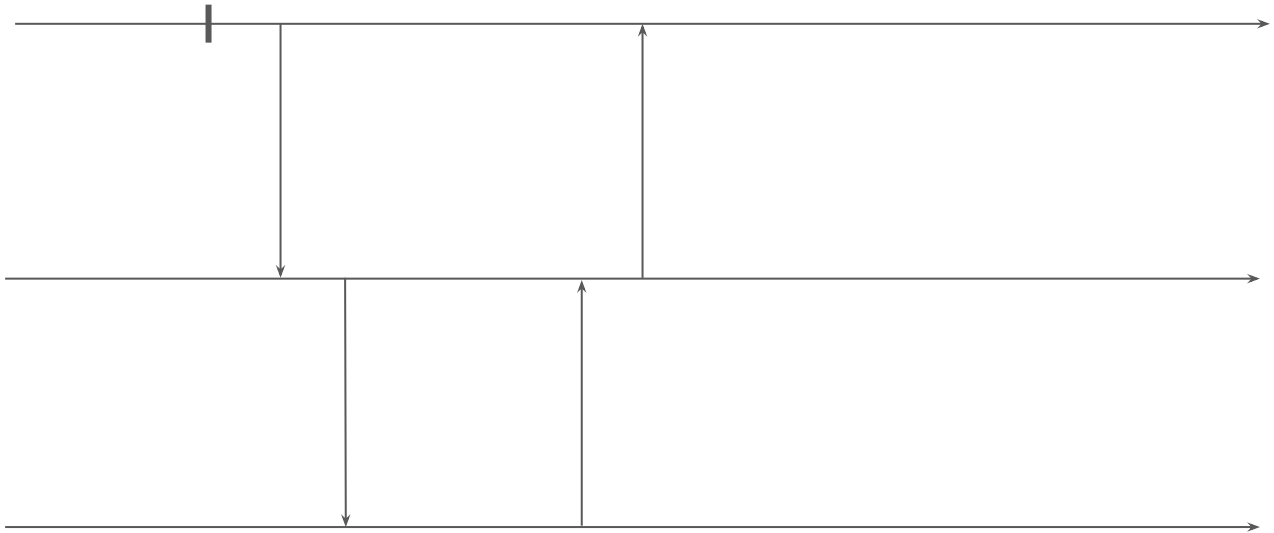
Модификация кадров



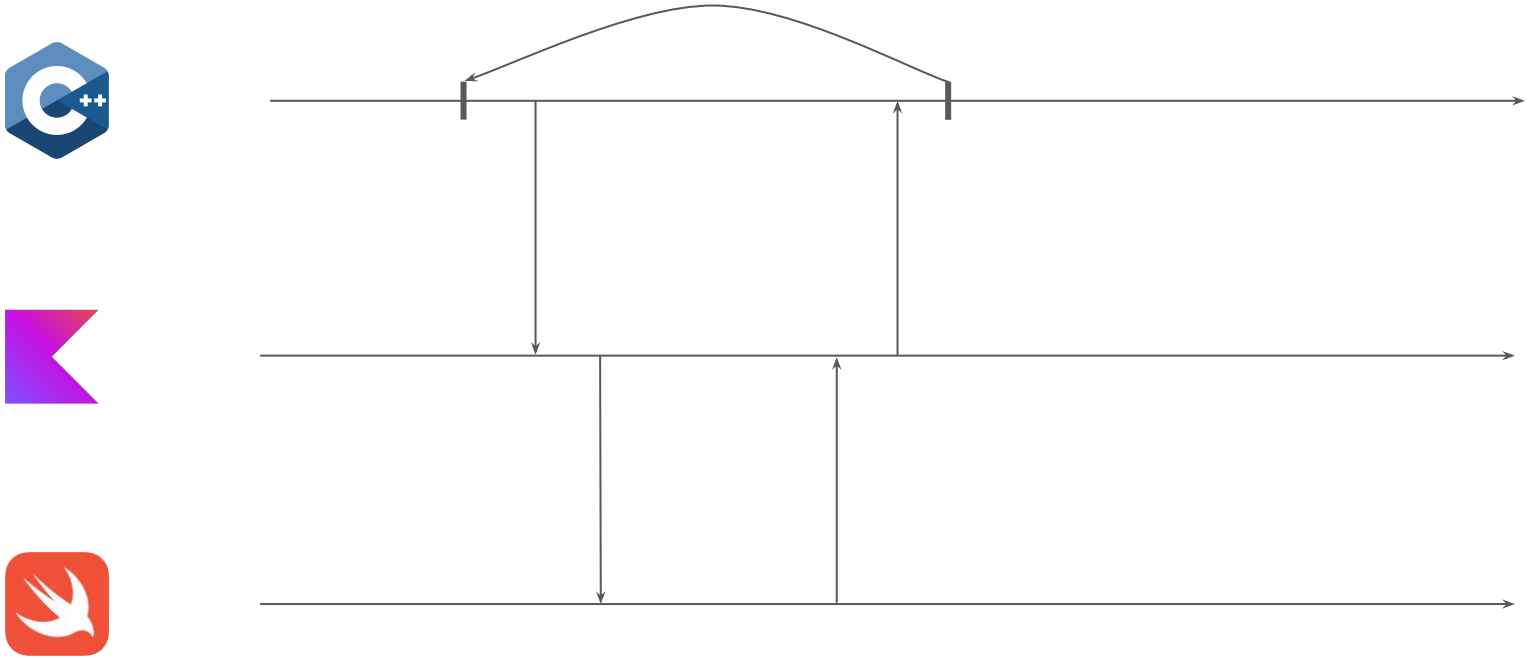
Модификация кадров



Модификация кадров



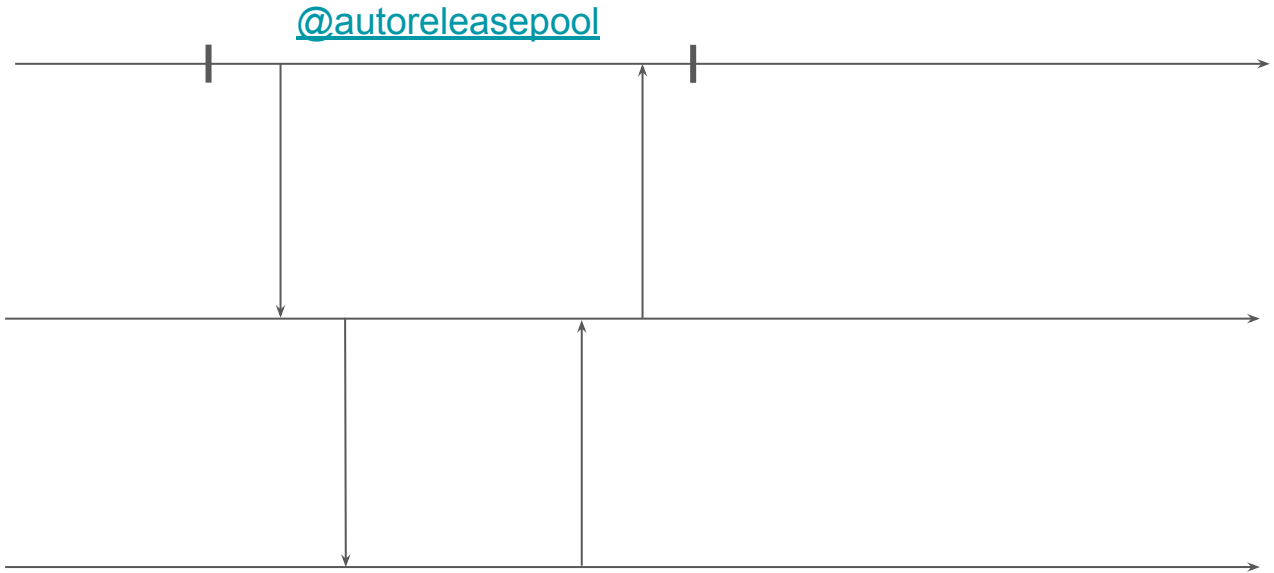
Модификация кадров



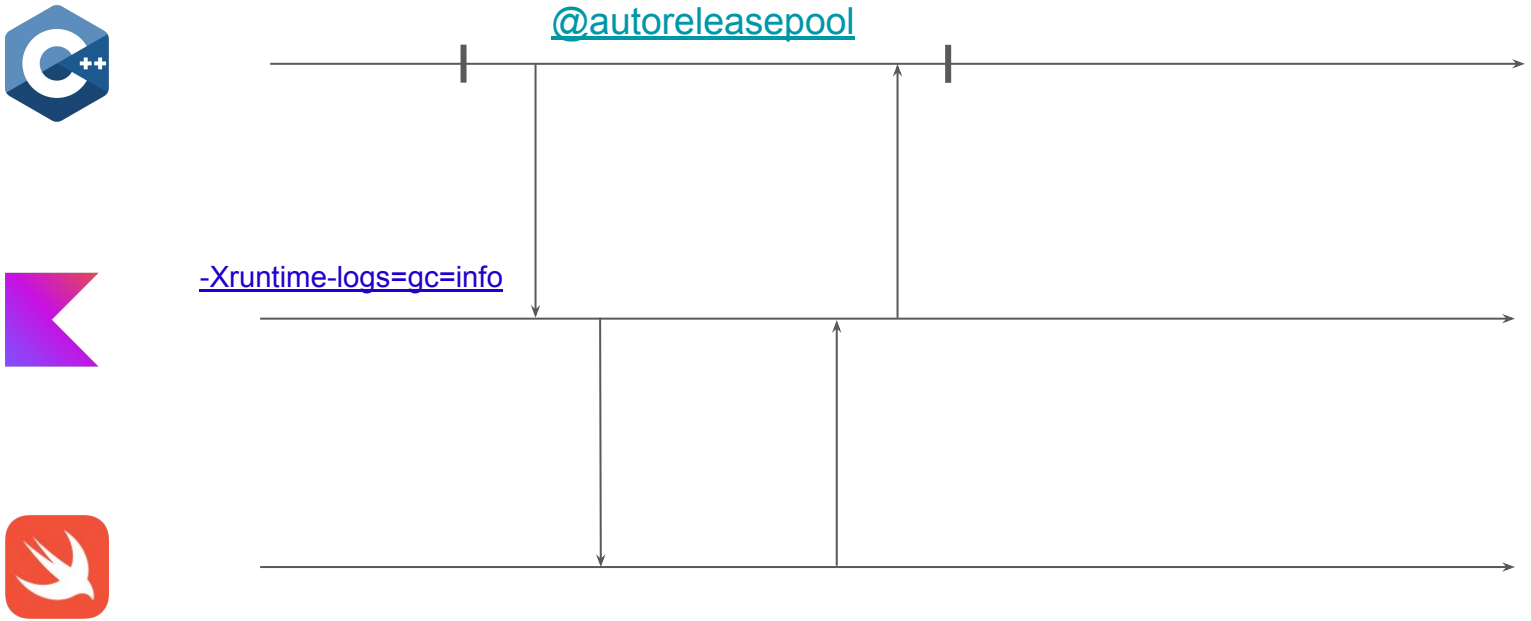
Profiler

Size	Responsible Library	Responsible Caller
8,09 MiB	IOGPU	-[IOGPUMetalResource initWithDevice:options:args:argsSize:]
8,09 MiB	IOGPU	-[IOGPUMetalResource initWithDevice:options:args:argsSize:]
8,09 MiB	IOGPU	-[IOGPUMetalResource initWithDevice:options:args:argsSize:]
7,98 MiB	CoreVideo	CVPixelBufferBacking::performStandardMemoryLayoutAndCopyIOSurfaceCreationProperties(void*, unsig…
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreVideo	CVPixelBufferBacking::performStandardMemoryLayoutAndCopyIOSurfaceCreationProperties(void*, unsig…
7,92 MiB	CoreVideo	CVPixelBufferBacking::performStandardMemoryLayoutAndCopyIOSurfaceCreationProperties(void*, unsig…
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreVideo	CVPixelBufferBacking::performStandardMemoryLayoutAndCopyIOSurfaceCreationProperties(void*, unsig…
7,92 MiB	CoreVideo	CVPixelBufferBacking::performStandardMemoryLayoutAndCopyIOSurfaceCreationProperties(void*, unsig…
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy
7,92 MiB	CoreGraphics	create_protected_copy

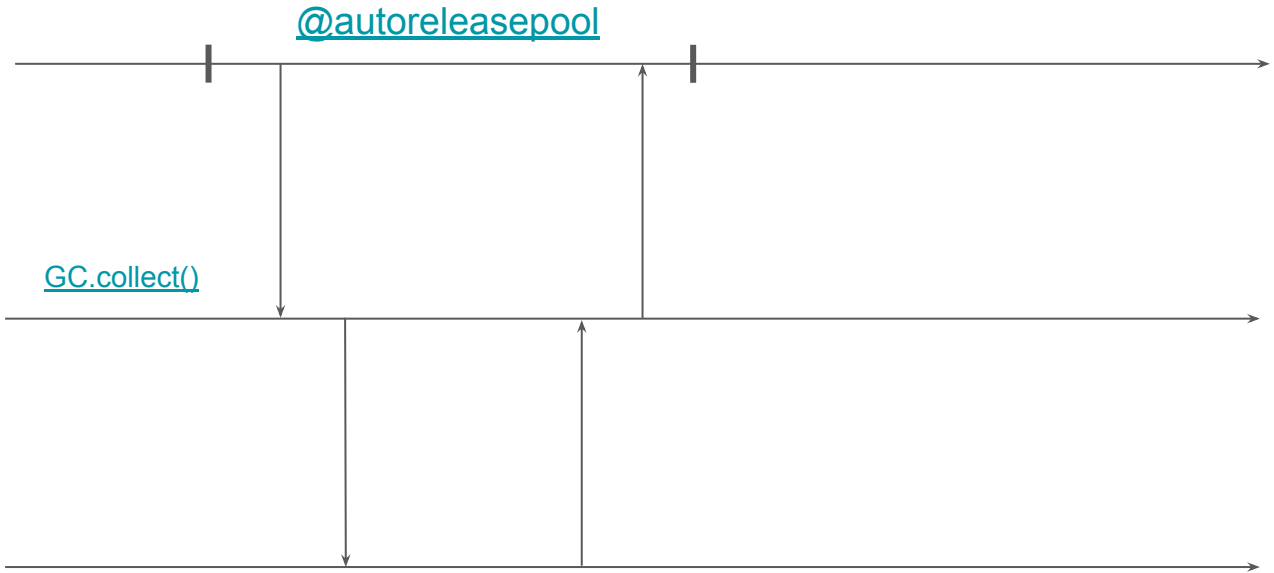
Системы управления памятью



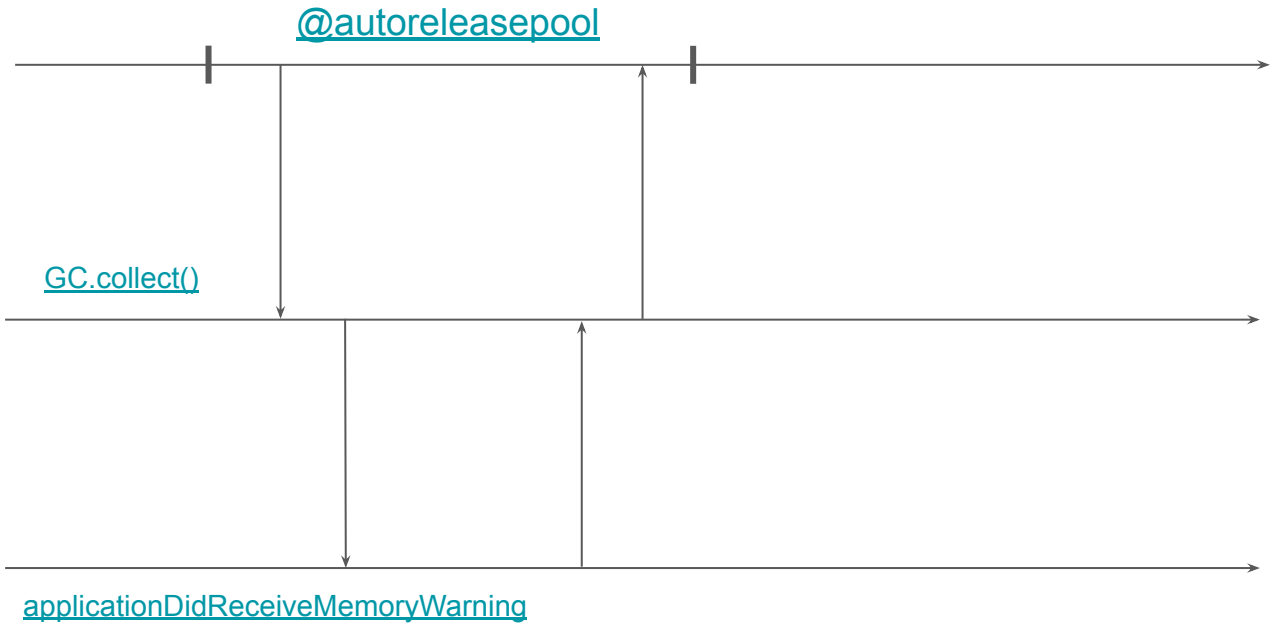
Системы управления памятью



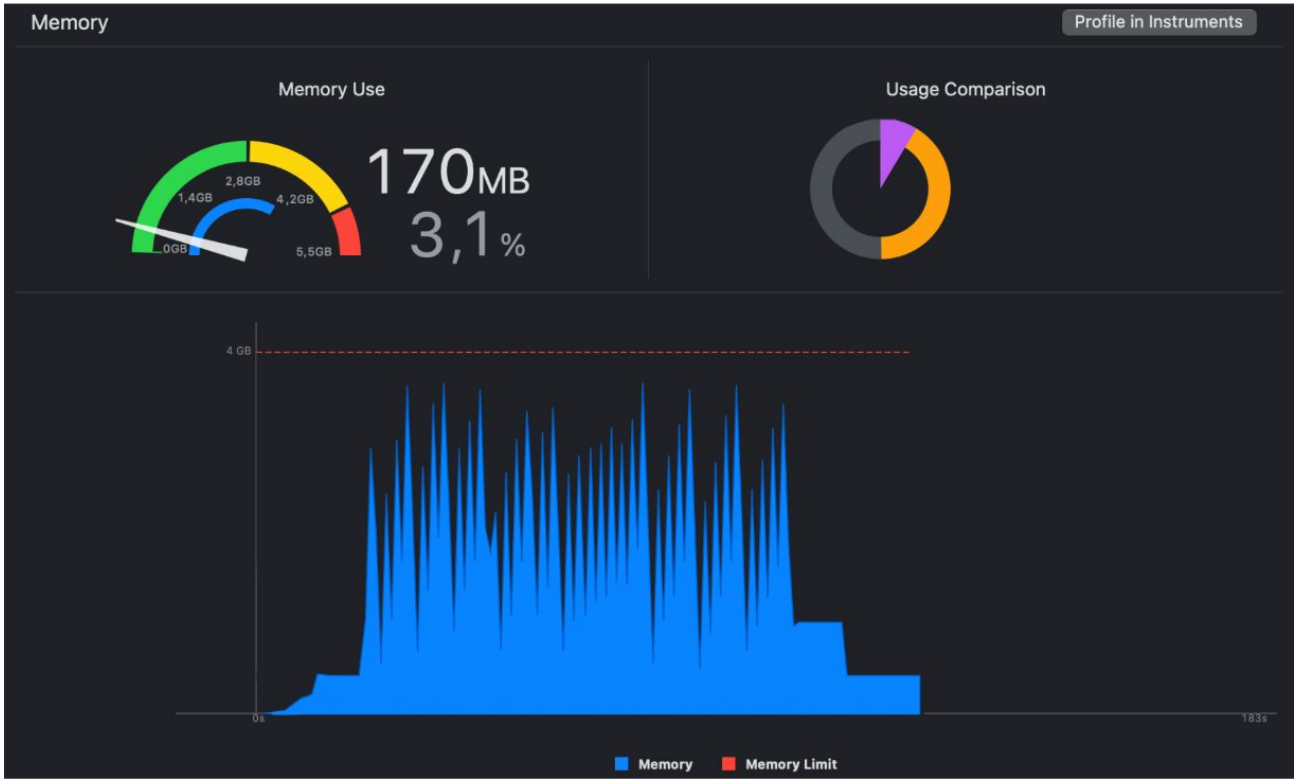
Системы управления памятью



Системы управления памятью



Системы управления памятью



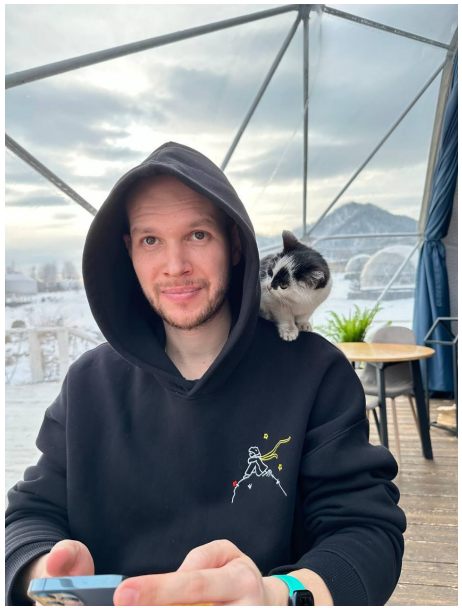
Выводы

- Необходимы знания о внутреннем устройстве системы
- Эксперименты
- Комбинация разных источников данных
- Фиксация результата

Итоги

- Code review
- Статический анализ кода
- Использование актуальных версий IDE
- Автоматизация рутины
- Прототипирование
- Атомарные изменения
- Эталонные решения - часть тест-системы
- Информативное логирование
- Автоматизация тестирования
- Изучайте новое

Спасибо за внимание



Kotlin