

УДК 004.9

ПРОЕКТИРОВАНИЕ ОСНОВНЫХ КОМПОНЕНТОВ ANDROID-СИСТЕМЫ ПЕРЕДАЧИ ФРАГМЕНТИРОВАННОЙ ВИДЕОЗАПИСИ НА УДАЛЁННЫЕ СЕРВЕРЫ

Марченко К. В.

студент,

Новосибирский государственный университет,

Новосибирск, Россия

Аннотация

В статье рассматривается схема функционирования компонентов съёмки и видеотрансляции Android-системы, осуществляющей передачу на удалённые серверы видеоданных по фрагментам в процессе их получения с камеры. Рассматривается архитектура компонента передачи видеозаписи. Описанные компоненты являются переиспользуемыми и открыты для использования различными приложениями.

Ключевые слова: видеосъёмка, передача данных, фрагментированная запись, архитектура системы, Android.

DESIGNING THE MAIN COMPONENTS OF AN ANDROID-SYSTEM FOR TRANSMISSION OF FRAGMENTED VIDEO RECORDING TO REMOTE SERVERS

Marchenko K. V.

student,

Novosibirsk State University,

Novosibirsk, Russia

Abstract

The paper overviews the functions of the recoding and transmitting components of an Android-based system designed to upload video data to remote servers in fragments as they are received from the camera. The transmitting component architecture is analyzed. The described componens are reusable and are open to be used in various applications

Keywords: video, data transmission, fragmented recording, system architecture, Android.

В настоящее время в самых разнообразных видах ситуаций люди фиксируют события с помощью видеосъёмки. При этом зачастую съёмка осуществляется с помощью мобильных устройств, подавляющее большинство которых имеют операционную систему Android [1]. Видеозаписи могут сохраняться не только на самом устройстве, но и передаваться с помощью интернета на удалённые серверы. Однако отнюдь не все сервисы поддерживают так называемый стриминг [2], для реализации которого необходима поддержка соответствующих протоколов. В таких ситуациях появляется необходимость передавать видеозапись по фрагментам малой длины, чтобы затем собирать их в один файл.

Рассмотрим схему функционирования компонентов съёмки и передачи получаемой видеозаписи на удалённые серверы и архитектуру последнего.

Съёмка

Запись начинается при получении сигнала от пользователя. Вначале происходит добавление видеодорожки, после чего устанавливаются следующие характеристики создаваемой видеозаписи:

- Формат – MP4
- FPS (Frames per Second) – 30
- Временной интервал между двумя ключевыми кадрами – 1 секунда

Таким образом, каждый тридцатый кадр – ключевой.

Далее происходит начало записи. После получения очередного несжатого кадра видеозаписи делается проверка, является ли он ключевым.

Если является, то данный ключевой кадр сохраняется в буфер, а экземпляр класса, отвечающего за создание видеозаписи, останавливается, сохраняя при этом созданный фрагмент видеозаписи локально на мобильное устройство. Об остановке видеозаписи сообщается слушателю, производящему трансляцию фрагментов. После этого происходит перезапуск экземпляра, отвечающего за создание видеозаписи класса.

Если кадр не был ключевым, то происходит его сжатие с помощью стандарта сжатия видео AVC (MPEG-4 Part 10 Advanced Video Coding, H.264) и добавление к записи.

В любом случае, далее начинается ожидание нового кадра.

При получении сигнала о прекращении записи созданный на данный момент фрагмент сохраняется на мобильном устройстве, компонент, занимающийся трансляцией фрагментов, оповещается о завершении записи, и после этого происходит непосредственно остановка записи.

Трансляция

Получив от компонента съёмки сигнал о её начале, компонент трансляции начинает свою работу. На первом шаге происходит создание пула потоков, в котором находятся 4 потока. Распределением задач по потокам занимается планировщик. Все операции внутри компонента трансляции выполняются не в главном потоке. Главный поток отвечает за пользовательский интерфейс (UI) и обработку взаимодействия с пользователем, поэтому его нельзя нагружать операциями ввода/вывода и долгими сетевыми запросами. Иначе можно наблюдать задержки в работе UI, а в худшем случае приложение может перейти в состояние, когда оно не отвечает на действия пользователя (так называемые ошибки ANR, “Приложение не отвечает” [3]).

После создания пула потоков и передачи его планировщику происходит считывание фрагмента видеозаписи из локального хранилища. По завершении считывания фрагмент отправляется на удалённый сервер. Обе операции выполняются в одном “фоновом” потоке, после чего он возвращается в пул и используется для отправки следующих фрагментов.

Когда пользователь останавливает запись, компонент трансляции фрагментов продолжает свою работу до тех пор, пока не будет отправлен последний фрагмент. После этого компонент прекращает работу.

Перейдём к обзору реализации компонента трансляции (рисунок 1).

Когда начинается запись, при наличии доступа в интернет создаётся экземпляр класса *PeriodicRecordRemoteTransmitter*, который отвечает за

поочерёдную передачу видеофрагментов и завершает свою работу, когда передан последний фрагмент.

При создании объекту класса *PeriodicRecordRemoteTransmitter* передаются реализации интерфейсов *LocalStorage* и *RemoteReposControllerProviding*, позволяющие работать с локальным хранилищем записей и репозиторием с удалённого сервера соответственно. Оба интерфейса являются наследниками более общего интерфейса *RecordRepository*, представляющего собой любое хранилище видеозаписей, без указания его конкретного местоположения.

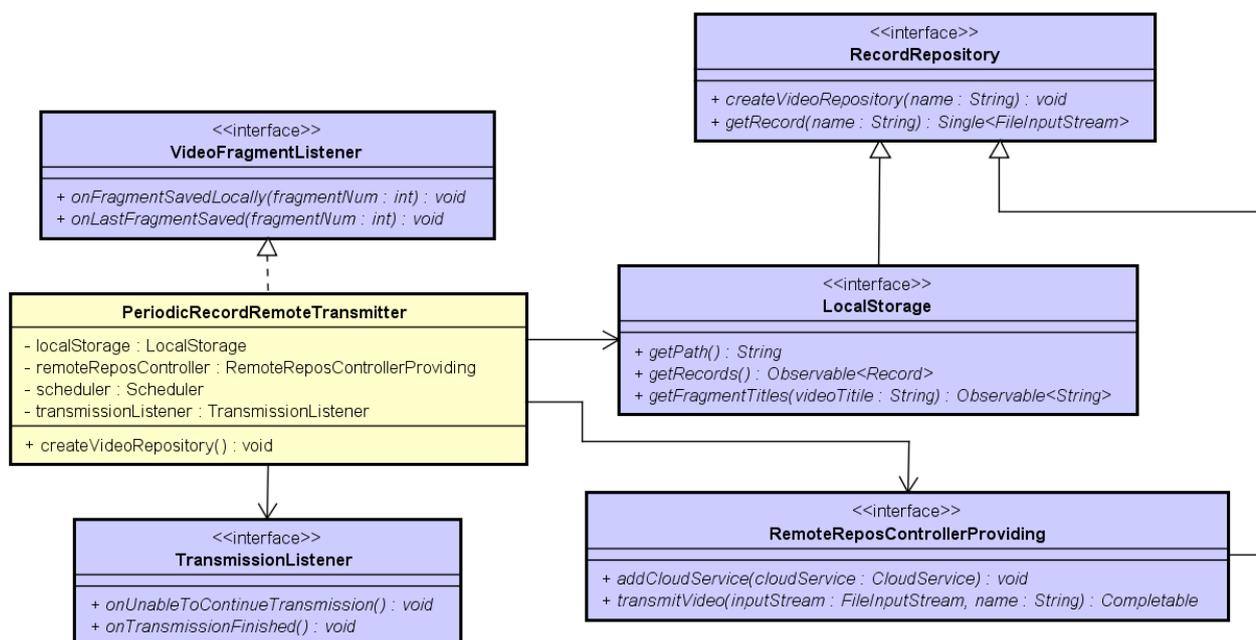


Рис. 1. Диаграмма классов компонента трансляции

При первом запуске трансляции создаётся пул потоков, содержащий в себе 4 потока. Он передаётся для управления планировщику *Scheduler*, который определяет, в каком потоке происходит считывание и отправка того или иного видеофрагмента.

При запуске трансляции на удалённом сервере создаётся репозиторий, в который будут загружаться все фрагменты текущей видеозаписи. Имя удалённого репозитория совпадает с именем локального.

PeriodicRecordRemoteTransmitter реализует интерфейс *VideoFragmentListener*, в котором определены 2 метода:

- *onFragmentSavedLocally* – вызывается, когда очередной фрагмент был сохранён локально. В качестве параметра передаётся номер этого фрагмента.
- *onLastFragmentSaved* – вызывается после сохранения последнего фрагмента видеозаписи. Параметром так же передаётся номер фрагмента.

В методе *onFragmentSavedLocally* вначале происходит считывание записи с мобильного устройства с помощью вызова, объявленного в интерфейсе *RecordRepository* метода *getRecord*. Параметром ему передаётся название записи, возвращаемое значение – *Single<FileInputStream>*, где *Single<>* – источник данных из RxJava 2, на который подписывается *PeriodicRecordRemoteTransmitter*. При успешном получении *FileInputStream* с помощью вызова метода *transmitVideo* у объекта, реализующего интерфейс *RemoteReposControllerProviding*, осуществляется передача фрагмента на удалённый сервер.

Параметрами являются поток видеоданных и название для создаваемого на удалённом сервере файла. Если при передаче видеофрагмента произошла ошибка, об этом оповещается слушатель *TransmissionListener*. После завершения передачи фрагмента поток, использовавшийся для этого, возвращается в пул потоков.

Когда был вызван метод *onLastFragmentSaved*, *PeriodicRecordRemoteTransmitter* передаёт последний фрагмент и оповещает слушателя о завершении трансляции с помощью обратного вызова *onTransmissionFinished*.

Таким образом, были рассмотрены спроектированные согласно основным принципам объектно-ориентированного программирования компоненты съёмки и передачи фрагментированной видеозаписи, позволяющие передавать запись на удалённые серверы по фрагментам малой длины в процессе её создания. Полученные компоненты переиспользуемы и могут быть встроены в системы разнообразных назначений.

Библиографический список

1. Mobile OS Market Share 2018 // Statista: The Statistics Portal. URL: <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems> (last access 14.05.2018).
2. Ozer J. What is streaming? Streaming Media Magazine, 2011, 26 February. URL: <http://www.streamingmedia.com/Articles/ReadArticle.aspx?ArticleID=74052> (last access 14.05.2018).
3. ANRs. Android Developers. URL: <https://developer.android.com/topic/performance/vitals/anr> (last access 14.05.2018).