

ЛЕБЕДЕВ М. А.

**МИНИМИЗАЦИЯ ЭНЕРГОПОТРЕБЛЕНИЯ, СВЯЗАННОГО С ОПРЕДЕЛЕНИЕМ
МЕСТОПОЛОЖЕНИЯ В МОБИЛЬНОМ ПРИЛОЖЕНИИ
ДЛЯ УДАЛЕННОГО МОНИТОРИНГА СОБЫТИЙ УСТРОЙСТВ
НА ОПЕРАЦИОННОЙ СИСТЕМЕ ANDROID**

Аннотация. В статье представлен анализ мобильных приложений для удаленного мониторинга событий устройств на операционной системе Android. Выявляется их главный недостаток, связанный с энергопотреблением. Предлагается решение, позволяющее минимизировать энергопотребление, связанное с определением местоположения в мобильном приложении для удаленного мониторинга событий устройств на операционной системе Android.

Ключевые слова: Android, мобильное приложение, местоположение, энергопотребление, мониторинг событий.

LEBEDEV M. A.

**MINIMIZATION OF POWER CONSUMPTION RELATED TO LOCATION TRACKING
IN MOBILE APPLICATION FOR REMOTE MONITORING
OF DEVICE EVENTS ON OPERATING SYSTEM ANDROID**

Abstract. The article presents an analysis of mobile applications for remote monitoring of device events on the operating system Android. Their main drawback associated with power consumption is revealed. A solution is proposed that minimizes the power consumption related to location tracking in a mobile application for remote monitoring of device events on the operating system Android.

Keywords: Android, mobile application, location, power consumption, event monitoring.

Одна из полезных функций мобильного устройства – это возможность определять местоположение его владельца. Приложения, которые используют эту функцию, нередко используются для контроля сотрудников служб, связанных с доставкой. Также существует много программ нацеленных на так называемый «родительский контроль», который позволяет членам семьи поддерживать контакт, узнавать местоположение друг друга и осуществлять контроль за своими детьми.

В сравнении с другими системами слежения, приложения мобильного мониторинга имеют ряд заметных преимуществ:

- низкая стоимость внедрения по сравнению с другими GPS-системами;
- высокая информативность;

- использование большего числа источников для определения местоположения (вышки сотовой связи, Wi-Fi точки, навигационные спутники);
- широкий спектр дополнительных возможностей.

В настоящее время существует немало мобильных приложений, которые дают возможность получать информацию о положении целевого устройства в заданный момент времени и решают задачи, связанные с контролем местоположения сотрудников. В ходе исследования приложений для операционной системы Android, реализующих такой функционал, были выявлены схожие проблемы. Главная проблема – серьезный расход заряда батареи смартфона в процессе работы приложения.

Проблема с сильным расходом батареи обусловлена, в первую очередь, непрерывным использованием соответствующих датчиков, нужных для определения геопозиции устройства, а также перманентной отправкой этих данных на сервер.

В данной статье будет разработан и описан один из возможных алгоритмов, позволяющих минимизировать энергопотребление, связанное с определением местоположения в мобильном приложении для удаленного мониторинга событий устройств на операционной системе Android.

Для начала определимся с возможностями системы и составим некоторые требования, предъявляемые к ней. Система будет ориентирована на работу с выездными сотрудниками какого-либо предприятия, т.е. взаимодействие будет происходить между диспетчером и рабочим данного предприятия. Несмотря на это, пользоваться обе взаимодействующие стороны будут одним приложением, функционал которого будет меняться в зависимости от типа аккаунта пользователя. Аккаунт диспетчера будет выступать в роли агрегатора аккаунтов сотрудников, т.е. существование сотрудника в системе независимо от диспетчера невозможно. Таким образом, система должна состоять из следующих частей: клиентское приложение с автоматической сменой возможностей в зависимости от типа аккаунта пользователя (диспетчер или сотрудник), бекенд системы.

На основе проведенного анализа и предполагаемого сценария использования приложения были составлены следующие требования к разрабатываемой системе.

1. Мобильное приложение должно позволять получать координаты, с помощью GPS, GSM-сети и Wi-Fi.
2. Мобильное приложение должно иметь режим экономии энергии.
3. Система должна позволять синхронизировать данные между мобильным приложением и сервером.
4. Система должна иметь функционал для работы оффлайн.

5. Система должна поддерживать возможность обмена сообщениями между сотрудником и диспетчером.
6. В системе должна быть поддержка PUSH-уведомлений на различные действия.
7. В системе должен присутствовать функционал создания и просмотра заданий.
8. В системе должен присутствовать функционал смены статуса сотрудника.
9. Мобильное приложение должно автоматически запускаться после старта системы.
10. Функционал мобильного приложения должен зависеть от типа авторизованного аккаунта пользователя (сотрудник или диспетчер)
11. Мобильное приложение должно иметь адаптированный для смартфонов интерфейс.

Для решения проблемы энергосбережения стоит разобраться, какие способы получения местоположения доступны в операционной системе Android.

В Android есть 3 провайдера (поставщика, источника) данных о местоположении.

GPS (GPS, AGPS): провайдер местоположения посредством GPS. Этот провайдер определяет местоположение с использованием одноименных спутников. В зависимости от условий ему может потребоваться некоторое время, чтобы вернуть обновленное местоположение. Оказывает сильное влияние на заряд батареи устройства.

Network (AGPS, CellID, WiFi MACID): поставщик местоположения, который использует сеть. Он определяет местоположение на основе информации, полученной от доступных точек сотовой связи и точек доступа Wi-Fi. Имеет среднее воздействие на заряд батареи устройства.

Passive (CellID, WiFi MACID): специальный провайдер местоположения для приема местоположений без фактического инициирования его обновления. Этот провайдер может использоваться для пассивного получения обновлений данных о локации. То есть информация, которую он использует, будет обновляться в момент, когда другие приложения или службы запрашивают их. Таким образом с помощью него мы получаем только закешированные системой обращения к GEO-данным от других приложений. Это могут быть как обращения Network (Wi-Fi) в активном и пассивном режиме, мониторинг вышек сотовой связи, так и использование GPS. Оказывает минимальное воздействие на заряд батареи устройства [2].

Почти все приложения, использующие в своей работе данные о локации, используют один из описанных выше провайдеров локации для определения местоположения, другие используют свои закрытые реализации. В доступных режимах экономии энергии, которые предоставляют исследованные приложения, суть заключается в выборе одного из

поставщиков локаций, что позволяет добиться либо максимальной точности, либо минимально возможной нагрузки на батарею. В последнем случае экономия энергии происходит за счет снижения точности. Поэтому решение проблемы с высоким энергопотреблением кроется не в выборе оптимального поставщика местоположения, а в том, как с ним построено взаимодействие нашего приложения. Но, несмотря на это, в нашем приложении будет присутствовать и стандартная настройка выбора поставщика локации.

Заметим, что обновление информации о географических координатах в базе данных будет происходить в моменты получения этих данных от датчиков устройства. Поэтому далее мы будем говорить только об обновлении данных поставщиками локации, а отправка этих данных на сервер будет подразумеваться.

Первый способ уменьшения энергозатрат, который напрашивается для реализации, это установка некоторого интервала обновления данных о локации устройства. Скорее всего, в проанализированных приложениях такой интервал равен некой константе. Соответственно, чем длиннее будет такой интервал, тем меньшее потребление заряда аккумулятора мы получим. Точность при любых интервалах останется на прежнем уровне и будет зависеть от выбранного поставщика. Минус заключается в том, что, исходя из специфики нашего приложения, диспетчеру может потребоваться увидеть местоположение сотрудника в случайный момент времени, а не дожидаться, когда данные придут на сервер.

Отсюда вытекает идея сделать такой режим работы приложения, при котором бы диспетчер получал информацию о местонахождении сотрудника только в моменты, когда такая информация ему потребуется, а именно, когда он будет наблюдать на экране своего смартфона за передвижением конкретного сотрудника. Соответственно, данные о местоположении сотрудника будут обновляться только в такие моменты времени, что позволит минимизировать энергозатраты.

Для того, чтобы диспетчер в режиме реального времени получал информацию о том, что сотрудник прибыл на точку выполнения задания, будем в этот момент времени обновлять географические координаты сотрудника, а также посылать уведомление диспетчеру.

Эти особенности будут реализованы только для режима энергосбережения, всегда будет возможность переключиться в обычный режим работы приложения, в котором обновление информации о местоположении будет происходить непрерывно. К тому же, логично, что данные будут обновляться и те моменты, когда сотрудник наблюдает свое местоположение на экране своего смартфона.

Исходя из сказанного, в приложении будет реализовано два режима работы.

1. Обычный режим, в котором данные о местоположении будут обновляться непрерывно.

2. Режим экономии энергии, который будет реализовывать описанные возможности.

Первым шагом реализации алгоритма, решающего проблемы энергосбережения, будет подбор подходящей под наши нужды базы данных, которая бы могла принимать множественные запросы на запись и чтение геоданных. Также требуется реализовать бекенд системы, который бы связывал мобильные клиенты диспетчера и сотрудника с базой данных. Системой, решающей обе задачи, была выбрана NoSQL база данных Firebase от компании Google. Вместо обычных HTTP-запросов она использует синхронизацию данных. Каждый раз, когда данные изменяются, любое подключенное устройство получает это обновление в течении миллисекунд. Данные в базе хранятся в формате JSON. Общение между базой данных Firebase и мобильным клиентом производится путем подключения пользователей к узлам баз данных. Таким образом, любое изменение в некотором узле базы, связанным с данными о местоположении сотрудника, будет немедленно отправлено на все устройства, которые в этот момент держат с ним (узлом) соединение.

Для того, чтобы мобильное приложение всегда оставалось отзывчивым и смогло бы получать и отправлять данные о локации требуется создание некоторой службы, которая будет существовать независимо от того, запущено приложение или нет. Для этого идеально подойдет Service, один из стандартных компонентов, предоставляемый SDK Android. Service – компонент приложения, который может выполнять длительные операции в фоновом режиме и не содержит пользовательского интерфейса. Именно в нем и будут реализованы методы взаимодействия с нашей базой данных.

Для оповещения об изменении данных в базе Firebase использует механизм обратных вызовов. Большие объемы данных и возможные тяжелые операции, связанные с ними, не рекомендуется выполнять в основном потоке Android-приложения, так как этот поток отвечает в том числе за отрисовку пользовательского интерфейса, который может зависнуть в ходе долгих операций на нем. Для корректной работы с потоками приложения будем использовать библиотеку RxJava 2, которая реализует принципы реактивного программирования для JVM. События Firebase, поступающие в методы обратных вызовов, будем транслировать в потоки данных и обрабатывать с помощью реализации паттерна «Наблюдатель». В нем есть объект Observable, который генерирует события. В нашем случае база данных будет генерировать события изменения своих полей, и Observer, который подписывается и получает эти события, вызывая методы для дальнейшей обработки данных, в том числе для вывода в реальном времени геоположения сотрудника на экране устройства диспетчера.

Подписка на события обновления локации сотрудника будет полностью связана с жизненным циклом приложения для операционной системы Android и активна только в тот период времени, когда диспетчер наблюдает за этим сотрудником на экране своего смартфона. Соответственно, получение информации от поставщиков местоположения на устройстве сотрудника будет происходить только в эти моменты, что и будет являться главным фактором уменьшения времени интенсивной нагрузки на батарею устройства-источника локации. В остальные моменты времени подписка будет неактивна, и сервис, запущенный в приложении сотрудника, будет находиться в режиме ожидания. Для определения моментов наблюдения диспетчером конкретных сотрудников, в узле базы данных, связанным с учетной записью диспетчера будет существовать специальное поле, хранящее уникальный идентификатор этого сотрудника. Сигналом к подписке на обновления определенного устройства будет являться изменение именно этого поля.

Таким образом, мы уменьшили расход энергии мобильного устройства, сведя к минимуму временной интервал взаимодействия нашего приложения с поставщиками локации, необходимый для нормального функционирования системы для удаленного мониторинга событий устройств на операционной системе Android.

ЛИТЕРАТУРА

1. Харди Б., Филлипс Б. Android. Программирование для профессионалов. – 2-е изд. – СПб.: Питер, 2016. – 640 с.
2. Android Location Providers – gps, network, passive – Tutorial [Электронный ресурс]. – Режим доступа: <https://developerlife.com/2010/10/20/gps>.