

## **ОПТИМІЗАЦІЯ ВИКОРИСТАННЯ РЕСУРСІВ АКУМУЛЯТОРА ПРИ ВІДСТЕЖЕННЯ ГЕОЛОКАЦІЇ В ОПЕРАЦІЙНІЙ СИСТЕМІ ANDROID**

Вінницький національний технічний університет;

### **Анотація**

*В матеріалах розглядається спосіб покращення продуктивності акумулятора при розробці додатків для відстеження геолокації в операційній системі .*

**Ключові слова:** акумулятор, ресурси, геолокація, місцезнаходження, Android.

### **Abstract**

*The materials consider a way to improve battery performance in the development of applications for tracking geolocation in the Android operating system.*

**Keywords:** battery, resources, location, geolocation, Android.

### **Вступ**

Фонові обмеження місцеположення, введені в Android 8.0 (рівень API 26), привернули увагу до теми, як використання служб локації впливає на розрядку акумулятора. Далі буде розглянути найкращі практики служби локації, щоб зробити додатки більш ефективними від заряду акумулятора. Застосування цих методів сприятливо відобразиться для програми незалежно від версії платформи, на якій вона працює.

### **Результати дослідження**

Спочатку, визначимо зміни, які були внесені у обмеження фонового розташування в Android 8.0:

1.Збір місцевого розташування зменшується, а місце розташування обчислюється та доставляється лише кілька разів на годину.

2.Геофонічна чутливість змінюється від десятків секунд до приблизно двох хвилин. Ця зміна помітно покращує продуктивність акумулятора - до 10 разів краще на деяких пристроях [1].

Далі описано перелік найкращих практик для покращення продуктивності акумулятора на пристрої при розробці додатків для відстеження геолокації.

#### **Видалення оновлення місцеположення**

Поширеним джерелом непотрібного розряду акумулятора є неможливість видалення оновлень місцеположення, коли вони більше не потрібні. Це може статися, наприклад, коли методи життєвого циклу активності `onStart()` або `onResume()` містять виклик `requestLocationUpdates()` без відповідного виклику для видалення `LocationUpdates()` у методах життєвого циклу `onPause()` або `onStop()`.

#### **Встановлення тайм-аутів**

Для захисту від розряду акумулятора можливе використання тайм-ауту, коли оновлення місцеположення повинні припинитися. Час очікування гарантує, що оновлення не триватиме нескінченно довго, і він захищає додаток у сценаріях, коли оновлення запитуються, але не видаляються (наприклад, через помилку в коді).

Для запиту в `fused location provider` кращим рішенням буде додавання таймаут, викликавши `setExpirationDuration()`, який отримує параметр, який представляє час у мілісекундах з моменту останнього виклику методу. Ви також можете додати тайм-аут, викликавши `setExpirationTime()`, який отримує параметр, який представляє час закінчення в мілісекундах з моменту останнього завантаження системи.

#### **Пакетні запити**

Для всіх випадків використання додатку не в передньому плані(фоновий режим) кращою практикою є виконання декількох запитів одночасно. Можна використовувати метод `setInterval()`, щоб вказати інтервал, через який буде обчислено розташування. Потім методом `setMaxWaitTime()`, щоб встановити інтервал, в якому розташування отримається додатком. Значення, передане методу `setMaxWaitTime()`, має бути кратним значенням, переданим методу `setInterval()`. Нижче представлений приклад на отримання місцезнаходження на мові програмування Java:

```
LocationRequest запит = новий LocationRequest ();  
request.setInterval (10 * 60 * 1000);  
request.setMaxWaitTime (60 * 60 * 1000);
```

У цьому випадку розташування обчислюється приблизно кожні десять хвилин, і приблизно шість точок даних про місцезнаходження доставляються в партії приблизно кожну годину. Хоч все ще отримуються оновлення локації кожні десять хвилин, заощаджуються ресурси акумулятора, оскільки пристрій прокидається лише кожну годину.

Використання пасивного оновлення місцезнаходження

У фонових випадках використання корисно зменшити оновлення місцезнаходження. Ліміти Android 8.0 застосовують цю практику, але програми, що працюють на старих пристроях, повинні намагатися максимально обмежити розташування фону.

Цілком імовірно, що додаток перебуває у фоновому режимі, інший додаток може часто запитувати оновлення місцезнаходження на передньому плані. Служби локації надають ці оновлення для програми. Приклад запиту про місцезнаходження, який умовно споживає дані про місцезнаходження на мові програмування Java:

```
LocationRequest запит = newLocationRequest ();  
request.setInterval (15 * 60 * 1000);  
request.setFastestInterval (2 * 60 * 1000);
```

У попередньому прикладі місце розташування обчислюється для вашої програми приблизно кожні 15 хвилин. Якщо інші додатки вимагають місцезнаходження, дані стають доступними для програми через максимум дві хвилини.

Під час пасивного споживання місцезнаходження не виникає витрата акумулятора, але, потрібно бути обережним у випадках, коли отримання даних про місцезнаходження викликає дорогі операції з процесором або введенням / виведенням. Щоб мінімізувати витрати акумулятора, інтервал, визначений у `setFastestInterval()`, не повинен бути занадто малим [2].

Дані практики оптимізації використання ресурсів акумулятора передбачають використання API локальних служб Google, які пропонують більш високу точність і накладають легше навантаження на акумулятор, ніж API-файли розташування.

## Висновки

Встановлено, перелік рекомендацій для покращення продуктивності акумулятора на пристрої при розробці додатків для відстеження геолокації.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1.developer.android: [Електронний ресурс] - режим доступу:  
<https://developer.android.com/guide/topics/location/battery>.

2. Muthumurugesan D., Nalini S., Vinodini R. Smart Way to Track the Location in Android Operating System/ Muthumurugesan D., Nalini S., Vinodini R. – IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661

**Очеретний Анатолій Богданович** — студент групи 2кн-16б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: [anatoliy.ocheretnyi@gmail.com](mailto:anatoliy.ocheretnyi@gmail.com)

Науковий керівник: **Озеранський Володимир Сергійович** — д-р техн. наук, доц. кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця