

## МЕТОДИ ОПТИМІЗАЦІЇ МОБІЛЬНИХ ДОДАТКІВ, ПРАЦЮЮЧИХ ПІД КЕРУВАННЯМ ОС ANDROID

Гречмак Д.В.

Науковий керівник – к.т.н., ст. викл. Ткачов В.М.

Харківський національний університет радіоелектроніки  
(61166, Харків, просп. Науки, 14, каф. ЕОМ, тел. (057) 702-13-54)

e-mail: [d\\_ec@nure.ua](mailto:d_ec@nure.ua)

Users often avoid downloading apps that seem too large, particularly in emerging markets where devices connect to often-spotty 2G and 3G networks or work on pay-by-the-byte plans. In addition to these issues, devices may have only limited storage available, making it even more important to be mindful about your app's size.

На ринках з дорогим інтернетом і малопотужними пристроями боротьба йде буквально за мегабайти. Завантажать ваш додаток чи ні - залежить від розміру. Наприклад, розробники Opera дуже пишаються тим, що їх додаток займає всього 8 Мб. Необхідно мінімізувати кількість бібліотек. Спочатку треба прибрати зайві бібліотеки, а потім прийти до відмови від сторонніх бібліотек в цілому. Вони спрощують роботу розробників, але заради пари зручних функцій підключати цілу бібліотеку нерозумно. Тому гарним тоном буде самостійно описувати логіку під потреби кожного проекту. Звичайно, це займає більше часу, більше коду, зате значно скорочує розмір програми.

Не тримайте у проекті бібліотеки зі схожим функціоналом. Коли використання бібліотек неминуче - створення реклами, наприклад – треба подивитися які бібліотеки вони містять всередині. Можливо, варто відмовитися від улюбленої на користь тієї, що вже лежить в іншій. Використовуйте легкі зображення. Ресурси модуля програми - картинки та інше - часто займають найбільше місця. Зменшуйте їх. Це простіше, ніж придумати, де зменшувати розмір надалі. Для цього гарно підходять векторні зображення - вони складаються з простих геометричних фігур, тому і зовні зрозумілі, і займають мало місця. Замість стандартної упаковки з п'яти картинок під різні формати дисплеїв, ви додаєте одну векторну. Вона змінює розмір, не втрачаючи якості. З мінусів - зображення повинне бути простим. Складну картинку намалювати в векторі можна, але це не оптимальний варіант, адже вона все одно буде багато важити і довго грузиться. Якщо картинка складна, можна використовувати WebP. Це гугловський формат стиснення, який дозволяє зменшити вагу зображення мало не в 10 разів. Необхідно користуватися утилітами для перетворення коду. Існує утиліта Proguard, але є й інші варіанти. Крім перетворення коду до стану нечитабельності, вона виявляє і видаляє невикористані ресурси, тим самим оптимізуючи його.

Користуйтеся ресурсами платформи. Наприклад, у Android є внутрішній клас `drawable objects`, який дозволяє малювати прості зображення всередині системи, не додаючи нічого в ресурси. Задайте логіку створення елемента і він відмалюється самостійно в момент завантаження графічного інтерфейсу. Використовуйте ресурси декілька раз. Наприклад, картинки які дублюються в різних кольорах можна зберігати не як дві копії, а як одну та колірний фільтр для неї. У пошуках таких оптимізаційних варіантів ви витратите час, але воно того варто заради збереження мінімального обсягу програми. Можливе використання App Bundle - тип збірки додатку, коли елементи для кожного користувача збираються індивідуально не розробником, а в PlayMarket. Це оптимізує розмір, PlayMarket повідомляє, що це допомагає зменшити розмір програми на 30%. Оптимізуйте швидкість відображення інтерфейсу - це актуально для користувачів зі «слабким залізом». Найчастіше призначений для користувача інтерфейс верстається в xml файлі - це зручно і швидко в порівнянні з версткою в коді. Це не завжди добре позначається на часі обробки UI, особливо на слабких пристроях. Можна трохи прискорити цей процес, якщо задати верстку в коді. Так ми заощадимо час на процесі перетворення xml верстки в код. Або ж можна використовувати клас `AsyncLayoutInflater`, щоб зробити цей процес асинхронним. Можливо, загальна швидкість обробки не збільшиться, але хоча б зробить користувальницький інтерфейс більш чутливим.

Список використаної літератури:

1. Vitalii Tkachov, Anna Budko, Kateryna Hvozdetska and Daryna Hrebenuk. Method of Building Dynamic Multi-hop VPN Chains for Ensuring Security of Terminal Access Systems // IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T): Kharkiv 06-09 oct. 2020, Kharkiv.

2. Tkachov, V., Hunko, M., Volotka, V.: Scenarios for Implementation of Nested Virtualization Technology in Task of Improving Cloud Firewall Fault Tolerance. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), pp. 759-763. IEEE (2019).

3. Hunko M.A., Tkachov V.M. Development of a module for sorting the ipaddresses of user nodes in cloud firewall protection of web resources. Дев'ята міжнародна науково-технічна конференція «Сучасні напрями розвитку інформаційнокомунікаційних технологій та засобів управління». 2019. С. 30.