

## **ДОСЛІДЖЕННЯ МЕХАНІЗМІВ УПРАВЛІННЯ РЕСУРСАМИ В ОПЕРАЦІЙНИХ СИСТЕМАХ ANDROID ТА IOS ДЛЯ РОЗРОБКИ МОБІЛЬНИХ ДОДАТКІВ**

В.П. Козлов, А.В. Козир  
(Україна, Дніпро, НТУ «Дніпровська політехніка»)

Добре відома головна відмінність двох операційних систем: Apple, на відміну від Google, повністю контролює вихідний код своєї операційної системи iOS, відкриваючи стороннім розробникам доступ тільки до окремих елементів. При цьому Стів Джобс неодноразово критикував Android за відкритість і, як наслідок, надмірну ускладненість, але історія показала, що будь-який підхід може обернутися як благом, так і прокляттям.

iOS вперше була представлена як iPhone OS 1, «мобільна версія Mac OS», разом з самим першим смартфоном компанії 9 січня 2007 року. Як і «комп'ютерна» операційна система Apple, iOS побудована на двох основних компонентах, розроблених Apple і NeXT - це ядро XNU і система Darwin, яка формально відповідає специфікації SUSv3 «Юнікса», не будучи їм як таким. Single UNIX Specification - загальна назва для сімейства стандартів, яким повинна задовольняти операційна система, щоб називатися «UNIX». При цьому XNU і Darwin, на відміну від iOS і macOS - системи з відкритим початковим кодом. Початковий код таких програм доступний для перегляду, вивчення та зміни, що дозволяє користувачеві взяти участь в доопрацюванні цієї відкритої програми, використовувати код для створення нових програм і виправлення в них помилок - через запозичення початковоого коду.

Закритість iOS дозволяє Apple контролювати не тільки власний софт, а й сторонніх розробників. Наприклад, незважаючи на безкоштовність iOS SDK і середовища розробки Xcode, кожен розробник змушений платити 99 доларів у рік за можливість використовувати емулятор Apple-пристроїв і розміщувати свої додатки в App Store. До березня 2008 року, коли SDK був викладений у відкритий доступ, сторонні розробники не могли писати програми під iPhone OS в принципі, магазин додатків з'явився тільки в другій версії ОС.

Шлях Android на мобільний ринок був куди більш тернистий. Спочатку придбана компанією Google в 2005 році Android Inc. працювала над ОС для кнопочкових телефонів з GPS. Однак презентація iPhone зламала команді на чолі з Енді Рубінім всі плани. Систему довелося терміново переключити, і тільки у вересні 2008-го разом зі смартфоном HTC Dream (він же T-Mobile G1) була представлена перша повністю робоча версія Android. На виправлення найбільш явних помилок і ключові доопрацювання пішов ще цілий рік - тільки до кінця 2009 року можна було говорити про можливість серйозно конкурувати з iOS.

Важливу роль в цьому зіграли Motorola і Samsung, які підтримали Google і почали в тому році випуск комплектуючих з Android на борту.

В основі Android лежить ядро Linux і філософія відкритого ПЗ. Це означає можливість всім бажаючим взяти рішення від Google за основу і доопрацювати відповідно до своїх уявлень, заплативши всього 25 доларів одноразово. Саме звідси з'явився ціла купа сторонніх прошивок і навіть «власних ОС» на базі Android. Ось тільки в момент появи Android на ринку в нього мало хто вірив, і навіть HTC випустила свій знаменитий смартфон тільки тому, що Google їй заплатила.

Що до оптимізації цих операційних систем? Користувачам, неважливо, як розробник оптимізує своє «залізо». Їм важливо, щоб все працювало, а ще краще - «літало». Google і Apple сьогодні справляються з цим завданням на одному рівні - не ідеально. Відзначимо, що перед пошуковим гігантом в даному випадку стоїть куди більш серйозне завдання: забезпечити роботу ОС не тільки на топових пристроях, а й на бюджетних варіантах, які Apple не випускає в принципі.

Але с точки зору розробників визначимо відмінності у принципах роботи багатозадачності у цих операційних системах.

Як працює багатозадачність в iOS? Всього у додатків в iOS є п'ять типів станів:

Неробочий - робота додатка була припинена або воно не було запущено.

Неактивний - додаток в активному режимі, але не отримує подій (наприклад, користувач заблокував екран).

Активний - нормальний стан додатку в режимі використання.

Фоновий - додаток прихований, але виконує код.

Припинений - додаток знаходиться в пам'яті, але не виконує код.

Коли ви натискаєте кнопку «Додому», програма перемикається з активної в фонову. Більшість додатків через декілька секунд після цього переходять в призупинений стан. Перший технічний нюанс полягає в тому, що припинене додаток залишається в пам'яті пристрою. Це зроблено для того, щоб воно могло продовжити роботу швидше, якщо користувач вирішить до нього повернутися. При цьому вона не використовує ресурси процесора і батарею.

Можна припустити, що якщо ПЗ знаходиться в пам'яті, то його потрібно видалити звідти для звільнення ресурсів. Насправді, iOS робить це самостійно. Якщо у користувача є припинений додаток і він запускає ресурсомісткий софт, наприклад складну тривимірну гру, iOS сама вивантажує з пам'яті припинену програму і переміщує її у неробочий стан. Іншими словами вона буде повністю видалена з ОЗУ і запущена з нуля при наступному запуску.

З цього можна зробити висновок, що головне правило багатозадачності iOS звучить наступним чином: панель багатозадачності є лише списком додатків, які нещодавно використовувалися, незалежно від того, в якому стані вони знаходяться. Операційна система самостійно вивантажує задачі - тоді, коли це необхідно.

За цим стає зрозуміло, що політика компанії Apple не дозволяє користувачу якимось чином впливати на виконання активних мобільних додатків у реальному часі. Ні за допомогою самої iOS, ні за допомогою спеціальних додатків. На такі заходи змушує йти політика безпеки Apple.

Існують додатки для iOS, які дозволяють системно відстежувати в реальному часі стан оперативної пам'яті, навантаження центрального процесора і сховища, але ніяк не впливати на ці показники.

Але що ж тоді робити з оптимізацією додатків для iOS? Насправді шляхів по оптимізації швидкості роботи мобільних додатків досить багато. Але їх потрібно проводити під час розробки додатків, а не під час їх виконання в системі. Розглянемо кілька способів оптимізації додатків в iOS, які можуть бути впроваджені під час розробки додатку:

- оптимізація часу запуску iOS-додатків;
- оптимізація ресурсів iOS додатків;
- оптимізація відзивчивості додатку.

У той же час, на Android системах ситуація дещо інша. Розробник також як і в iOS має різноманітні можливості для використання ресурсів пристрою при розробці свого додатку. Але, відкритий початковий код системи надає розробнику з "root" правами практично повний доступ до працюючої системи. Також, є спеціальні додатки, які дозволяють користувачеві безпосередньо змінювати такі показники як частота центрального процесора і об'єм виділеної оперативної пам'яті для конкретного додатку.

Такі можливості дозволяють оптимізувати ресурси системи прямо під час її роботи, якщо це необхідно користувачу. Такий підхід дозволяє явно управляти всіма аспектами операційної системи, на відміну від iOS системи, яка не надає користувачу та навіть розробнику ніякої можливості контролю над ресурсами системи в реальному часі.

#### ПЕРЕЛІК ПОСИЛАНЬ:

1. Дэйв Марк и др. iOS 6 SDK. Разработка приложений для iPhone, iPad и iPod touch - Beginning iOS 6 Development Exploring the iOS SDK. — М.: [«Вильямс»](#), 2013. — с. 672
2. Голощапов А. Google Android: программирование для мобильных устройств. — СПб.: [БХВ-Петербург](#), 2010. — с. 448