**Тема 4 Архітектура ОС Android. Інструменти розробника. Емулятори**

Розглянемо основні компоненти операційної системи Android (рис. 2.1):



Рисунок 2.1 Архітектура ОС Android

З точки зору архітектури, система Android представляє собою повний програмний стек, в якому можна виділити такі рівні:

**Рівень додатків (Applications)** – набір встановленого прикладного програмного забезпечення;

**Рівень каркасу додатків (Application Framework)** забезпечує розробникам доступ до API, наданих компонентами системи рівня бібліотек;

**Набір бібліотек і середовище виконання (Libraries & Android Runtime)** забезпечує найважливіший базовий функціонал для додатків, містить віртуальну машину Dalvik і базові бібліотеки Java необхідні для запуску Android додатків;

**Базовий рівень (Linux Kernel)** – рівень абстракції між апаратним рівнем і програмним стеком.

Розглянемо детальніше ці рівні.

**Applications**. Android постачається з набором основних додатків, що включає в себе календар, карти, браузер, менеджер контактів і ін. Всі перераховані програми написані на Java.

**Application Framework.** Будучи відкритою платформою, Android дає розробникам можливість створювати гнучкі та інноваційні програми. Розробники можуть використовувати апаратні можливості пристрою, отримувати інформацію про місцезнаходження, виконувати завдання у фоновому режимі, встановлювати оповіщення та багато іншого. Розробники мають повний доступ до тих же API, що використовуються в основних додатках.

Архітектура додатків розроблена з метою спрощення повторного використання компонентів; будь-який додаток може «публікувати» свої можливості і будь-яке інший додаток може потім використовувати ці можливості (з урахуванням обмежень безпеки). Цей же механізм дозволяє замінювати стандартні компоненти на призначені користувацькі.

**Libraries**. Android включає в себе набір C/C++ бібліотек, які використовуються різними компонентами системи. Ці можливості доступні розробникам в контексті застосування Android Aplication Framework. Деякі основні бібліотеки, перераховані нижче:

 Mедіа бібліотеки – ці бібліотеки надають підтримку відтворення і запису багатьох популярних аудіо, відео форматів і форматів зображень, в тому числі MPEG4, MP3, AAC, AMR, JPG, PNG і інших;

 Surface Manager – управляє доступом до підсистеми відображення 2D і 3D

графічних шарів;

 LibWebCore – сучасне веб-ядро, на якому побудований браузер Android;

 SGL – основне графічне ядро 2D;

 3D бібліотеки – реалізовані на основі OpenGL; бібліотеки використовують або апаратне 3D-прискорення (при його наявності), або вмикаються програмно;

 FreeType – підтримка растрових і векторних шрифтів;

 SQLite – механізм бази даних, доступний для всіх додатків.

**Android Runtime**. Android включає в себе набір основних бібліотек, які забезпечують більшість функцій, доступних в бібліотеках Java. Кожна програма Android працює в своєму власному процесі, зі своїм власним екземпляром віртуальної

машини Dalvik. Dalvik була написана так, що пристрій може працювати ефективно з декількома віртуальними машинами одночасно.

Dalvik проектувалася спеціально під платформу Android. Віртуальна машина оптимізована для низького споживання пам’яті та роботи на мобільному апаратному забезпеченні. Dalvik використовує власний байт-код. Android-додатки перекладаються компілятором в спеціальний машинно-незалежний низькорівневий код. І саме Dalvik інтерпретує і виконує таку програму при виконанні на платформі. Крім того, за допомогою спеціальної утиліти, що входить до складу Android SDK, Dalvik здатна перекладати байт-коди Java в коди власного формату і також виконувати їх у своїй віртуальному середовищі.

**Linux Kernel**. Android базується на Linux 2.6 з основними системними службами – безпека, управління пам’яттю, управління процесами і модель драйверів. Розробники Android модифікували ядро Linux, додавши підтримку апаратного забезпечення, використовуваного в мобільних пристроях і, найчастіше, недоступного на комп’ютерах.

Розробник зазвичай взаємодіє з двома верхніми рівнями архітектури Android для створення нових додатків. Бібліотеки, система виконання і ядро Linux приховані за каркасом додатків та абстракцій.

Для того, щоб встановити програму на пристроях з ОС Android створюється файл з розширенням \*.apk (Android package), який містить виконувані файли, а також допоміжні компоненти, наприклад, файли з даними і файли ресурсів. Після установки на пристрій кожен додаток «живе» в своєму власному ізольованому екземплярі віртуальної машини Dalvik. У версіях, починаючи з Android 4.4 Kitkat, є можливість переключитися з Dalvik на більш швидкий ART (Android Runtime, середовище виконання, розроблене Google). В Android 5.0 Dalvik був повністю замінений на ART.

**2.3 Інструменти розробника**

Як правило, розробка Android-додатків здійснюється на мові Java. Тому, в першу чергу, необхідно встановити Java Development Kit (JDK).

**Java** – це об’єктно-орієнтована мова програмування. Програми на Java транслюються в байт-код, що виконується віртуальною машиною Java, яка обробляє байт-код і передає інструкції обладнанню як інтерпретатор. Перевага подібного способу виконання програм полягає в повній незалежності байт-коду від операційної

системи і устаткування, що дозволяє виконувати Java-додатки на будь-якому пристрої, для якого існує відповідна віртуальна машина. Іншою важливою особливістю Java є гнучка система безпеки завдяки тому, що виконання програми повністю контролюється віртуальною машиною. Будь-які операції, які перевищують встановлені повноваження програми (наприклад, спроба несанкціонованого доступу до даних або з’єднання з іншим комп’ютером) викликають негайне їх переривання. Слід зауважити, що фактично, більшість архітектурних рішень, прийнятих при створенні Java, було продиктовано бажанням надати синтаксис, схожий з С/C++. В Java використовується практично ідентичний синтаксис для оголошення змінних, передачі параметрів і операторів. Тому ті, хто вже має досвід програмування на C/C++, зможуть швидко освоїтися і почати писати Java-додатки.

**Android Software Development Kit (SDK)** містить багато інструментів і утиліт для створення і тестування додатків. Наприклад, за допомогою SDK Manager можна встановити Android API будь-якої версії (рис. 2.2), а також перевірити репозиторій на наявність доступних, але ще не встановлених пакетів і архівів.



Рисунок 2.2 Вікно Android SDK

**JDK (Java Development Kit –** це безкоштовно розповсюджуваний комплект розробки додатків на мові Java, що включає в себе компілятор Java, стандартні бібліотеки класів Java, зразки коду, документацію, різні утиліти і систему Java Runtime Environment (JRE). До складу JDK не входить інтегроване середовище розробки IDE (Integrated Development Environment). Тому після того, як буде встановлено JDK, слід встановити IDE.

**Найпопулярніші середовища розробки під Android**

**Eclipse** – вільне модульне інтегроване середовище розробки програмного забезпечення. Розробляється і підтримується Eclipse Foundation.

**IntelliJ IDEA** – комерційне інтегроване середовище розробки для різних мов програмування (Java, Python, Scala, PHP та ін.) від компанії JetBrains.

**Android Studio** – середовище розробки під Android, яке базується на IntelliJ IDEA. Надає інтегровані інструменти для розробки і налагодження. Додатково до всіх можливостей, очікуваним від IntelliJ, в Android Studio реалізовані:

 підтримка процесу збирання додатку на основі Gradle;

 специфічний для Android рефакторинг і швидке виправлення дефектів;

 інструменти для пошуку проблем з продуктивністю, з юзабіліті, з сумісністю версій та інших;

 можливості ProGuard (утиліти для скорочення, оптимізації і обфускаціі коду) і підписування додатків;

 майстри для створення загальних Android конструкцій і компонентів, які працюють на основі шаблонів;

 WYSIWYG-редактор, який працює на екранах багатьох розмірів та розширень, вікно попереднього перегляду, що показує запущений додаток одразу на декількох пристроях і в реальному часі;

 вбудована підтримка хмарної платформи Google.

**Intel XDK** – середовище для розробки кросплатформних мобільних додатків; включає в себе інструменти для створення, налагодження та збірки ПЗ, а також емулятор пристроїв; підтримує розробку для Android, Apple iOS, Microsoft Windows 8, Tizen; підтримує мови розробки: HTML5 і JavaScript.

**Intel Beacon Mountain** – ще одне середовище розробки від компанії Intel. Надає всі інструменти, необхідні для проектування, розробки, налагодження та оптимізації додатків під Android. Підтримує розробку для цільових платформ на основі процесорів

Intel Atom і ARM. Beacon Mountain побудована на основі Android IDE (Eclipse, Android

ADT, Android SDK).

**Marmalade SDK** – кросплатформне SDK від Ideaworks3D Limited. Являє собою набір бібліотек, зразків коду, інструментів і документації, необхідних для розробки, тестування і розгортання додатків для мобільних пристроїв. Використовується, в основному, для розробки ігор.

**2.4. Емулятори**

Емуляція (англ. Emulation) в обчислювальній техніці – комплекс програмних, апаратних засобів або їх поєднання, призначений для копіювання (або емуляції) функцій однієї обчислювальної системи (гостя) на інший, відмінній від першої, обчислювальної системі (хост) таким чином, щоб поведінка, яка емулюється, як можна ближче відповідала поведінці оригінальної системи (гостя). Метою є максимально точне відтворення поведінки на відміну від різних форм комп’ютерного моделювання, в яких імітується поведінка деякої абстрактної моделі.

Емулятор – це віртуальний мобільний пристрій, який запускається на комп'ютері. За допомогою емулятора можна розробляти і тестувати програми без використання реальних пристроїв. На рис. 2.3 наведено приклад запущеного стандартного емулятора.



Рисунок 2.3 Стандартний емулятор Android

До переваг використання емуляторів можна віднести простоту їх використання і низьку вартість. Розробнику не потрібно купувати величезну кількість пристроїв з різними характеристиками, щоб перевірити працездатність програми на різних смартфонах. Досить створити кілька емуляторів з необхідними характеристиками і запустити на них додаток. На жаль, емулятори мають і ряд недоліків:

1) вимагають багато системних ресурсів;

2) через відмінності в архітектурі процесорів комп’ютера і смартфона повільно запускаються. Сучасні персональні комп’ютери побудовані на архітектурі x86 і x64, а більшість процесорів смартфонів на Android – ARM. Процес емуляції однієї архітектури на інший надзвичайно складний і відбувається досить повільно;

3) у деяких випадках стандартного емулятора недостатньо. Йдеться про можливості смартфонів, якими звичайні комп’ютери не володіють (наприклад, наявність датчика GPS або акселерометра). У таких випадках повноцінне відлагодження можна провести тільки з використанням реального пристрою.

**Альтернативні емулятори**

Стандартний емулятор, що поставляється разом з Android SDK, не влаштовує багатьох. Існують проекти, що підтримують розробку та розвиток альтернативних емуляторів. Як приклад можна привести Genymotion (https[://www.genymotion.com/](http://www.genymotion.com/)) – швидкий емулятор Android (на думку його розробників). Він містить попередньо налаштовані образи Android (x86 з апаратним прискоренням OpenGL). Genymotion доступний для Linux, Windows і MacOS X і вимагає для своєї роботи VirtualBox. Іншими словами, Genymotion є віртуальною машину з встановленою ОС Android, яку користувач запускає так само, як і інші віртуальні машини. Проблема високого споживання системних ресурсів, звичайно ж не зникає, проте швидкість запуску істотно збільшується. В даний час активно розвивається.

**Можливість відлагодження на реальних пристроях**

Розроблений додаток можна запустити на реальному пристрої, наприклад, на смартфоні. Для цього необхідно виконати попередню роботу. Для відлагодження додатків на реальних пристроях, як правило, необхідно:

1. налаштувати пристрій (включити режим налагодження по USB);

2. налаштувати комп’ютер (для Windows необхідно встановити потрібний драйвер вручну, потрібні права адміністратора);

3. налаштувати середовище розробки і запустити додаток на пристрої.