

# Операционная система Android



Панчук Владимир, 19 мая 2011г.



**Android** — операционная система для мобильных телефонов, планшетных компьютеров, нетбуков и смартбуков, основанная на ядре Linux. Изначально разрабатывалась компанией Android Inc., которую затем купила Google. Впоследствии Google инициировала создание [Open Handset Alliance \(OHA\)](#), которая сейчас и занимается поддержкой и дальнейшим развитием платформы. Android позволяет создавать [Java](#)-приложения, управляющие устройством через разработанные Google библиотеки. Также есть возможность писать приложения на [Си](#) и других языках программирования с помощью Android Native Development Kit. Первым устройством, работающим под управлением Android, стал разработанный компанией [HTC](#) смартфон [T-Mobile G1](#), презентация которого состоялась 23 сентября 2008 года. Вскоре другие производители заявили о намерении выпустить устройства с

Android. Кроме того, энтузиасты портировали Android на ряд известных устройств, среди которых: [Nokia N810](#), смартфоны [HTC Touch](#), [HTC TyTN II](#). Также известны случаи успешного портирования на компьютеры архитектуры x86. Отдельного упоминания достойна инициатива компании [Coolc](#), которая занялась не только портированием Android на [Neo FreeRunner](#), но и строит свой бизнес на продаже этих [OpenSource](#)-смартфонов с предустановленной мобильной платформой Google.

*Альтернативные прошивки, разрабатываемые открытым сообществом*

Существует сообщество энтузиастов, разрабатывающее полностью open-source варианты Android-прошивок. CyanogenMod и VillainROM — известные примеры данных прошивок.

- удаления из Android-устройства сервисов Google (например, таких как синхронизация данных) — для обеспечения локализации данных пользователя только на Android-устройстве — исключения возможности передачи идентификационной информации (IMEI, номера телефона, GPS-координат и т. д.) на серверы Google;
- более оперативного и частого (по сравнению с производителями самих аппаратов) предоставления новых версий Android OS;
- дополнения Android-прошивки новыми настройками и функциями. Возможность для хранения загруженных приложений на MicroSD карте (для версий Android до 2.2) и т. д.

Некоторые производители Android-устройств аппаратно блокируют возможность перепрошивки (например, HTC), некоторые аппаратно не блокируют (но тем не менее сохраняют трудности по перепрошивке; например, LG (для версий Android до 2.2.1)). А некоторые производители (Sony Ericsson в некоторых моделях аппаратов) делают всё чтобы продвинутый пользователь мог не только установить другую прошивку, но и самостоятельно создать свою (предоставляется детальная инструкции по замене прошивок, программное обеспечение, документация по архитектуре программной платформы, код оригинальной прошивки и т.д.). Риск, связанный с возможной поломкой телефона в процессе разблокирования, перекладывается на пользователя, который в случае разблокирования загрузчика соглашается с условиями, указывающими на досрочную потерю гарантии на телефон (для восстановления гарантии необходимо снова установить официальным флэшером официальную прошивку).

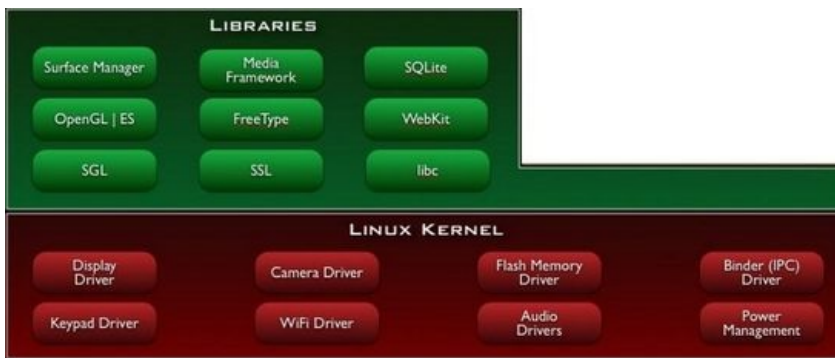
## Архитектура Android

### Уровень ядра Linux (Linux Kernel)



Как известно, Андроид основан на несколько урезанном ядре ОС Linux и поэтому на этом уровне мы можем видеть именно его (версии 2.6.x). Оно обеспечивает функционирование системы и отвечает за безопасность, управление памятью, энергосистемой и процессами, а также предоставляет сетевой стек и модель драйверов

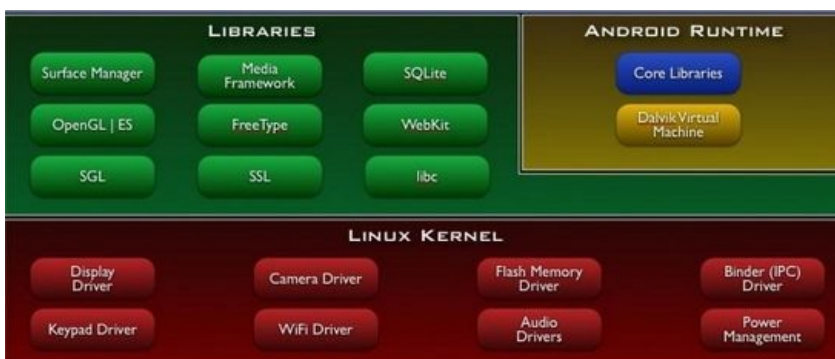
### Уровень библиотек (Libraries)



набор библиотек (Libraries), предназначенный для обеспечения важнейшего базового функционала для приложений. То есть именно этот уровень отвечает за предоставление реализованных алгоритмов для вышележащих уровней, поддержку файловых форматов, осуществление кодирования и декодирования информации (в пример можно привести мультимедийные кодеки), отрисовку графики и многое другое. Библиотеки реализованы на C/C++ и скомпилированы под конкретное аппаратное обеспечение устройства, вместе с которым они и поставляются производителем в предустановленном виде.

1. Surface Manager – в ОС Android используется композитный менеджер окон. Вместо того чтобы производить отрисовку графики напрямую в буфер дисплея, система посылает поступающие команды отрисовки в закадровый буфер, где они накапливаются вместе с другими, составляя некую композицию, а потом выводятся пользователю на экран. Это позволяет системе создавать интересные бесшовные эффекты, прозрачность окон и плавные переходы.
2. Media Framework – библиотеки. С их помощью система может осуществлять запись и воспроизведение аудио и видео контента, а также вывод статических изображений. Поддерживаются многие популярные форматы, включая MPEG4, H.264, MP3, AAC, AMR, JPG и PNG.
3. SQLite – легковесная и производительная [реляционная СУБД](#), используемая в Android в качестве основного движка для работы с базами данных, используемыми приложениями для хранения информации.
4. OpenGL ES ([OpenGL for Embedded Systems](#)) – подмножество графического программного интерфейса OpenGL, адаптированное для работы на встраиваемых системах.
5. FreeType – библиотека для работы с битовыми картами. Это высококачественный движок для шрифтов и отображения текста.
6. WebKit – библиотеки браузерного движка, используемого также в десктопных браузерах Google Chrome и Apple Safari.
7. SGL (Skia Graphics Engine) – открытый движок для работы с 2D-графикой. Графическая библиотека является продуктом Google и часто используется в других их программах.
8. SSL - библиотеки для поддержки одноименного криптографического протокола.
9. Libc – стандартная библиотека языка C, а именно её BSD реализация, настроенная для работы на устройствах на базе Linux. Носит название Bionic.

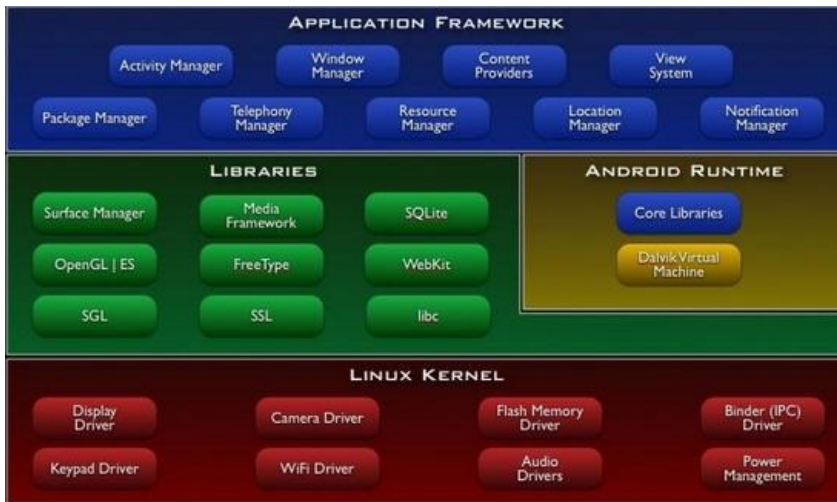
На этом же уровне располагается *Android Runtime* – среда выполнения.



Ключевыми её составляющими являются набор библиотек ядра и виртуальная машина [Dalvik](#). Библиотеки обеспечивают большую часть низкоуровневой функциональности, доступной библиотекам ядра языка Java.

Каждое приложение в ОС Android запускается в собственном экземпляре виртуальной машины Dalvik. Таким образом, все работающие процессы изолированы от операционной системы и друг от друга. И вообще, архитектура Android Runtime такова, что работа программ осуществляется строго в рамках окружения виртуальной машины. Благодаря этому осуществляется защита ядра операционной системы от возможного вреда со стороны других её составляющих. Поэтому код с ошибками или вредоносное ПО не смогут испортить Android и устройство на его базе, когда сработают. Такая защитная функция, наряду с выполнением программного кода, является одной из ключевых для надстройки Android Runtime.

## Уровень каркаса приложений (Application Framework)



Android позволяет использовать всю мощь **API**, используемого в приложениях ядра. Архитектура построена таким образом, что любое приложение может использовать уже реализованные возможности другого приложения при условии, что последнее откроет доступ на использование своей функциональности. Таким образом, архитектура реализует принцип многократного использования компонентов ОС и приложений.

Основой всех приложений является набор систем и служб:

1. Богатый и расширяемый набор представлений (Views), который может быть использован для создания визуальных компонентов приложений, например, списков, текстовых полей, таблиц, кнопок или даже встроенного web-браузера.
2. Контент-провайдеры (Content Providers), управляющие данными, которые одни приложения открывают для других, чтобы те могли их использовать для своей работы.
3. Менеджер ресурсов (Resource Manager), обеспечивающий доступ к ресурсам без функциональности (не несущими кода), например, к строковым данным, графике, файлам и другим.
4. Менеджер оповещений (Notification Manager), благодаря которому все приложения могут отображать собственные уведомления для пользователя в строке состояния.
5. Менеджер действий (Activity Manager), который управляет жизненными циклами приложений, сохраняет данные об истории работы с действиями, а также предоставляет систему навигации по ним.
6. Менеджер местоположения (Location Manager), позволяющие приложениям периодически получать обновленные данные о текущем географическом положении устройства.

## Уровень приложений (Applications)



В состав Android входит комплект базовых приложений: клиенты электронной почты и SMS, календарь, различные карты, браузер, программа для управления контактами и много другое. Все приложения, запускаемые на платформе Android написаны на языке Java, но нужно отметить, что существует возможность разрабатывать программы и на C/C++ , и на Basic и с использованием других языков.. Что интересно, Android не делает разницы между основными

приложениями телефона и сторонним программным обеспечением - таким образом, ключевые компоненты, такие как набор номера, рабочий стол или почтовый клиент GMail, можно заменить альтернативными аналогами.

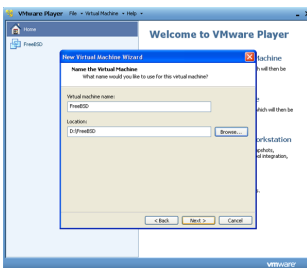
## Запуск Android OS на виртуальной машине VMware

Нам потребуется [VMware Player](#), и образ [Android-x86-1.6 rc2](#)

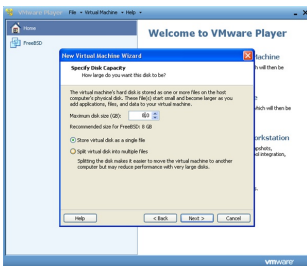
Запускаем VMware Player



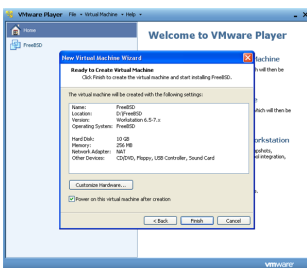
Выбираем из приведенных пунктов **New Virtual Machine**.



Прописываем имя виртуальной машины и место куда ее установим.

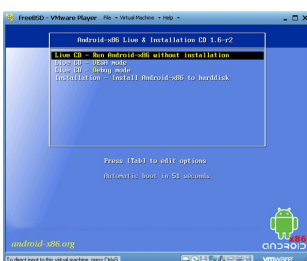


Определяем объем диска виртуальной машины рекомендованный размер 8GB



Завершающий этап виртуальная машины готова к созданию

Запуск образа [Android-x86-1.6 rc2](#)



После выбора Vesa mode происходит загрузка и через несколько секунд появляется рабочий стол Android OS:



Список использованной литературы:

1. [Википедия](#)
2. <http://3ppc.net/forum/showthread.php?t=9913>
3. <http://www.mobisoftinfotech.com/blog/tag/android/>



Статья опубликована на сайте Omoled.ru - Образовательные сообщества  
Ссылка на статью: <http://omoled.ru/publications/view/37>