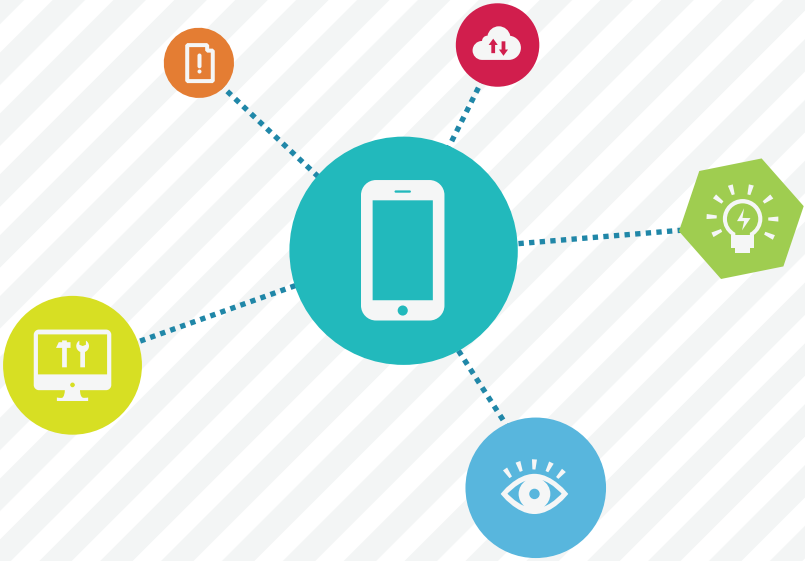
Сильвен Ретабоуил

Android NDK. Разработка приложений под Android на C/C++



ДМК ПРЕСС
издательство

УДК 004.451.9Android
ББК 32.973.26-018.2
Р31

Ретабоуил Сильвен

Р31 Android NDK. Разработка приложений под Android на C/C++: пер. с англ. Киселева А.Н. – М.: ДМК Пресс, 2012. – 496 с.: ил.

ISBN 978-5-94074-657-7

В книге показано, как создавать мобильные приложения для платформы Android на языке C/C++ с использованием пакета библиотек Android Native Development Kit (NDK) и объединять их с программным кодом на языке Java. Вы узнаете как создать первое низкоуровневое приложение для Android, как взаимодействовать с программным кодом на Java посредством механизма Java Native Interfaces, как соединить в своем приложении вывод графики и звука, обработку устройств ввода и датчиков, как отображать графику с помощью библиотеки OpenGL ES и др.

Издание предназначено для разработчиков мобильных приложений, как начинающих так и более опытных, уже знакомых с программированием под Android с использованием Android SDK.

УДК 004.451.9Android
ББК 32.973.26-018.2

ISBN 978-1-84969-152-9 (анг.)
ISBN 978-5-94074-657-7 (рус.)

Содержание

Глава 1
Подготовка окружения

Вы готовы заняться созданием программ для мобильных устройств? Ваш компьютер работает, мышь и клавиатура подключены, а монитор освещает рабочий стол? Тогда не будем ждать ни минуты!

В этой главе мы сделаем следующее:

- загрузим и установим инструменты, необходимые для разработки приложений на платформе Android;
- настроим среду разработки;
- подключим и подготовим для работы устройство на платформе Android.

Приступая к разработке программ для Android

Человек отличается от животных способностью использовать инструменты. Разработчики для Android, особый вид, к которому вы собираетесь примкнуть, ничем не отличаются от людей!

При разработке приложений для Android можно использовать следующие три *платформы*:

- Microsoft Windows PC;
- Apple Mac OS X;
- Linux PC.

Поддерживаются 32- и 64-битные версии Windows 7, Vista, Mac OS X и Linux, однако Windows XP – только в 32-битной версии. Из Mac OS X поддерживаются только версии от 10.5.8 и выше и только для архитектуры Intel (платформа на процессоре PowerPC не поддерживается). Операционная система Ubuntu поддерживается, лишь начиная с версии 8.04 (Hardy Heron).

Все это неплохо, но если только вы не способны читать и писать двоичный код, как текст на русском языке, наличия одной опера-

ционной системы будет недостаточно. Нам также потребуется *специальное программное обеспечение*, предназначенное для разработки для платформы Android:

- *инструменты разработки ПО на Java* (Java Development Kit – **JDK**);
- *инструменты разработки ПО для Android* (Software Development Kit – **SDK**);
- *инструменты разработки низкоуровневого ПО для Android* (Native Development Kit – **NDK**);
- *интегрированная среда разработки* (Integrated Development Environment – IDE): Eclipse.

Платформа Android, а точнее система компиляции в Android NDK, тесно связана с операционной системой Linux. Поэтому нам потребуется настроить некоторые утилиты и установить среду окружения, поддерживающую их: **Cygwin** (до версии NDK R7). Подробнее данная тема обсуждается ниже в этой же главе. Наконец, для использования всех этих утилит нам потребуется старая, добрая командная оболочка: мы будем использовать **Bash** (является командной оболочкой по умолчанию в Cygwin, Ubuntu и Mac OS X).

Теперь, когда известно, какие инструменты потребуются при разработке для Android, приступим к установке и настройке.

> **Примечание.** Следующий раздел описывает процесс установки и настройки в Windows. Если вы пользуетесь Mac или Linux, можете сразу перейти к разделу «Настройка в Mac OS X» или «Настройка в Linux».

Настройка в Windows

Прежде чем начинать установку инструментов, необходимых при разработке для Android, следует должным образом *подготовить Windows*.

Время действовать — подготовка Windows для разработки на платформе Android

Для работы с Android NDK необходимо настроить Cygwin-среду, подобную Linux для Windows:

> **Совет.** Начиная с версии NDK R7, устанавливать Cywin (шаги с 1 по 9) больше не требуется. Android NDK уже содержит необходимые утилиты для Windows (таке как `ndk-build.cmd`).

1. Откройте страницу http://cygwin.com/install.html.
2. Загрузите файл **setup.exe** и запустите его.
3. Выберите пункт **Install from Internet** (установите из Интернета).
4. Следуйте указаниям мастера установки.
5. Выберите сайт, откуда будут загружаться пакеты Cygwin, как показано на рис. 1.1. Возможно вы предпочтете выбрать сервер, находящийся в вашей стране.

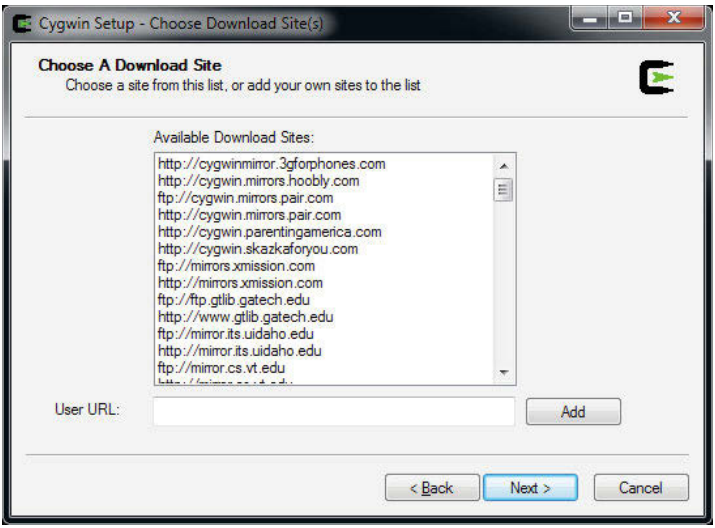


Рис. 1.1. Выбор сайта для загрузки

6. Когда будет предложено, выберите пакеты **Devel/make** и **Shells/bash**, как показано на рис. 1.2.
7. Следуйте инструкциям мастера установки до конца. Это может потребовать некоторого времени в зависимости от пропускной способности вашего подключения к Интернету.
8. После установки запустите Cygwin. При первом запуске будут созданы файлы параметров.
9. Выполните следующую команду, как показано на рис. 1.3, чтобы убедиться в работоспособности Cygwin:

```
$ make -version
```

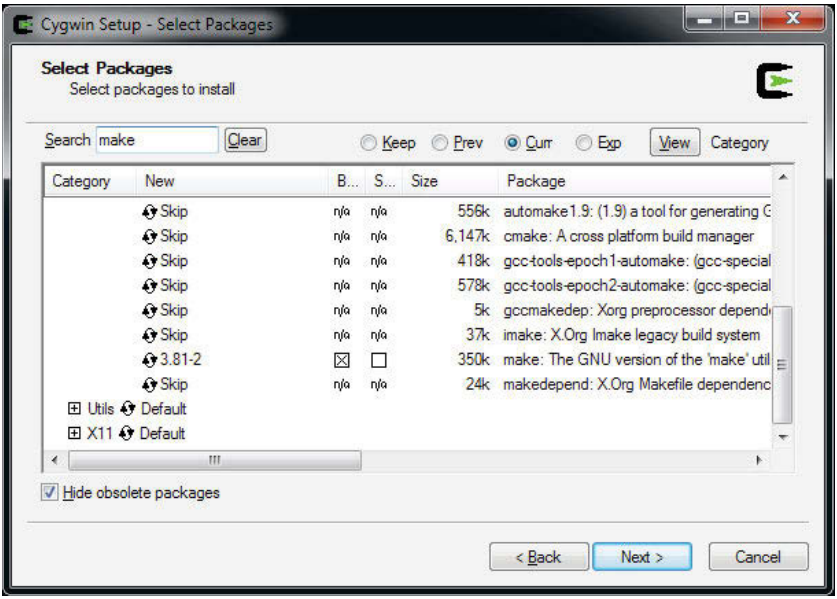


Рис. 1.2. Выбор пакетов для установки

Рис. 1.3. Результат выполнения команды make -version

Для работы Eclipse и компиляции программного кода на языке Java в байт-код необходимо установить Java Development Kit. Очевидным выбором в Windows является пакет Oracle Sun JDK:

1. Посетите веб-сайт компании **Oracle** и загрузите последнюю версию пакета Java Development Kit на странице http://www.oracle.com/technetwork/java/javase/downloads/index.html.
2. Запустите загруженную программу и следуйте инструкциям мастера установки. В конце установки автоматически откроется окно браузера и будет предложено зарегистрировать загруженную копию JDK. Этот шаг не является обязательным и его можно пропустить.

3. Чтобы гарантировать использование вновь установленного пакета JDK, следует определить его местоположение в переменных окружения. Откройте **Control panel** (Панель управления) и перейдите в панель **System** (Система) (или щелкните правой кнопкой мыши на пункте **Computer** (Компьютер) в меню **Start** (Пуск) и выберите пункт **Properties** (Свойства) контекстного меню). Затем перейдите в раздел **Advanced system settings** (Дополнительные параметры системы). Появится окно с заголовком **System Properties** (Свойства системы). Наконец, выберите вкладку **Advanced** (Дополнительно) и щелкните на кнопке **Environment Variables** (Переменные окружения).
4. В окне **Environment Variables** (Переменные окружения) добавьте в список **System variables** (Системные переменные) переменную `JAVA_HOME`, значением которой должен быть путь к каталогу установки JDK. Затем отредактируйте значение переменной `PATH` (или `Path`), добавив в самое начало каталог `%JAVA_HOME%\bin` и разделительную точку с запятой. Проверьте правильность введенной информации и закройте окно.
5. Откройте окно терминала и выполните команду `java -version`, чтобы проверить установленную версию. Вы должны получить результат, похожий на представленный на рис. 1.4. Убедитесь, что номер версии, выведенный в терминале, совпадает с номером версии только что установленного пакета JDK:

```
$ java -version
```

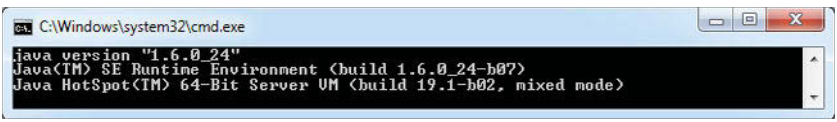


Рис. 1.4. Результат выполнения команды java -version

Для компиляции проектов из командной строк, пакет Android SDK поддерживает *Ant* – утилиту на языке Java, позволяющую автоматизировать процесс сборки. Установите ее:

1. Откройте страницу http://ant.apache.org/bindownload.cgi и загрузите выполняемые файлы Ant, упакованные в ZIP-архив.
2. Распакуйте архив Ant в любой каталог по своему выбору (например, `C:\Ant`).

3. Откройте снова окно **Environment Variables** (Переменные окружения), как описывается в п. 3 в списке выше, и создайте переменную `ANT_HOME`, значением которой должен быть путь к каталогу `Ant`. Добавьте путь `%ANT_HOME%\bin` в конец значения переменной `PATH`:

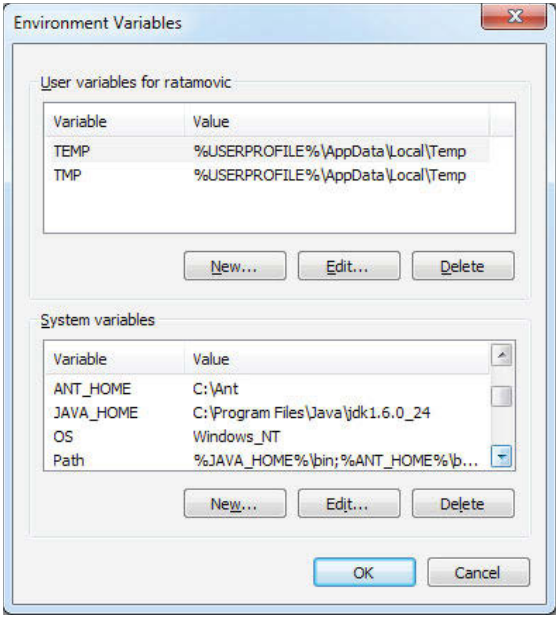


Рис. 1.5. Результат добавления переменной ANT_HOME

4. В окне терминала Windows проверьте версию Ant, чтобы убедиться, что она работает, как показано на рис. 1.6:

Рис. 1.6. Результат проверки версии Ant

Что получилось?

Мы подготовили Windows и все утилиты, необходимые для установки инструментов разработки ПО для платформы Android: **Cygwin** и **Java Development Kit**.

Cygwin – это пакет открытого программного обеспечения, позволяющего на платформе Windows эмулировать Unix-подобное окружение. Его целью является интеграция в Windows программного обеспечения, следующего стандарту POSIX (для таких ОС, как Unix, Linux и др.). Его можно рассматривать как промежуточный слой между приложениями для Unix/Linux (но скомпилированными в Windows) и самой ОС Windows.

Мы также развернули пакет Java Development Kit версии 1.6 и убедились в его работоспособности, выполнив команду в терминале. Поскольку в Android SDK используется механизм обобщенных типов (generic – генерики), при разработке приложений для Android минимально необходимой является версия JDK 1.5. Установка JDK в Windows выполняется очень просто, однако важно убедиться, что предыдущие версии, такие как JRE (Java Runtime Environment – окружение времени выполнения Java, предназначенное для выполнения программ, но не для их разработки) не будут мешать нам. Именно поэтому мы определили переменные окружения `JAVA_HOME` и `PATH` и тем самым гарантировали использование соответствующей версии JDK.

Наконец, мы установили утилиту Ant, которую будем использовать в следующей главе для сборки проектов вручную. Утилита Ant не является обязательной при разработке приложений для Android, но она обеспечивает отличную возможность объединения различных операций в последовательности.

Где находится домашний каталог Java? Определение переменной окружения `JAVA_HOME` не является обязательным условием. Однако `JAVA_HOME` является распространенным соглашением, которому следуют многие Java-приложения. Одним из таких приложений является утилита Ant. Она сначала пытается отыскать команду `java` в каталоге, описываемом переменной `JAVA_HOME` (если определена), а затем в списке путей `PATH`. Если позднее вы установите более новую версию JDK в другой каталог, не забудьте переопределить значение переменной `JAVA_HOME`.

Установка инструментов разработки для Android в Windows

После установки JDK можно приступать к установке *Android SDK и NDK*, необходимых для создания, компиляции и отладки программ для платформы Android.

Время действовать – установка Android SDK и NDK в Windows

1. Откройте веб-браузер и перейдите по адресу http://developer.android.com/sdk. На этой странице перечислены все доступные версии SDK, по одной для каждой платформы.
2. Загрузите пакет Android SDK для Windows, упакованный в выполняемый файл установки.
3. Перейдите по адресу http://developer.android.com/sdk/ndk и загрузите пакетAndroid NDK (не SDK!) для Windows, упакованный в ZIP-архив.
4. Запустите программу установки *Android SDK*. Выберите каталог для установки (например, `C:\Android\android-sdk`), учитывая, что пакеты Android SDK и NDK в сумме займут более 3 Гб дискового пространства (в настоящее время!) при установке всех официальных версий *прикладных программных интерфейсов* (Application Programming Interface – API). В качестве меры предосторожности не используйте пробелы в именах промежуточных и конечного каталогов, куда выполняется установка.
5. Следуйте инструкциям мастера установки до конца. В конце отметьте флажок **Start SDK Manager** (Запустить панель управления SDK):

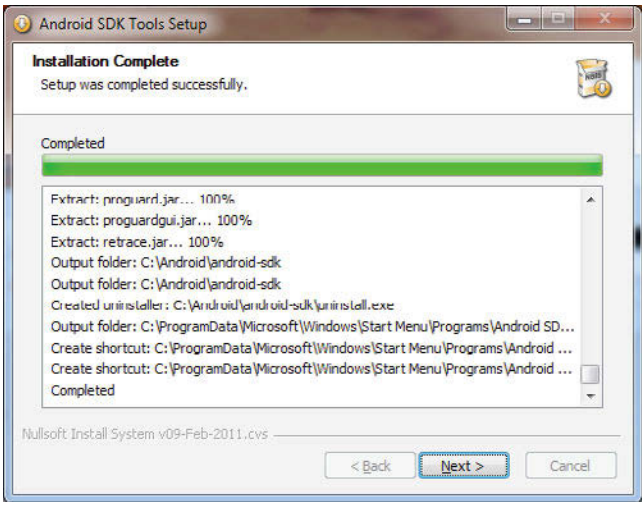


Рис. 1.7. Установка пакета Android SDK

6. По окончании установки запустится программа **Android SDK and AVD Manager** (Панель управления Android SDK и AVD) и автоматически откроется окно **Package installation** (Установка пакетов).
7. Отметьте флажок **Accept All** (Отметить все) и щелкните на кнопке **Install** (Установить), чтобы запустить установку компонентов платформы Android, как показано на рис. 1.8:

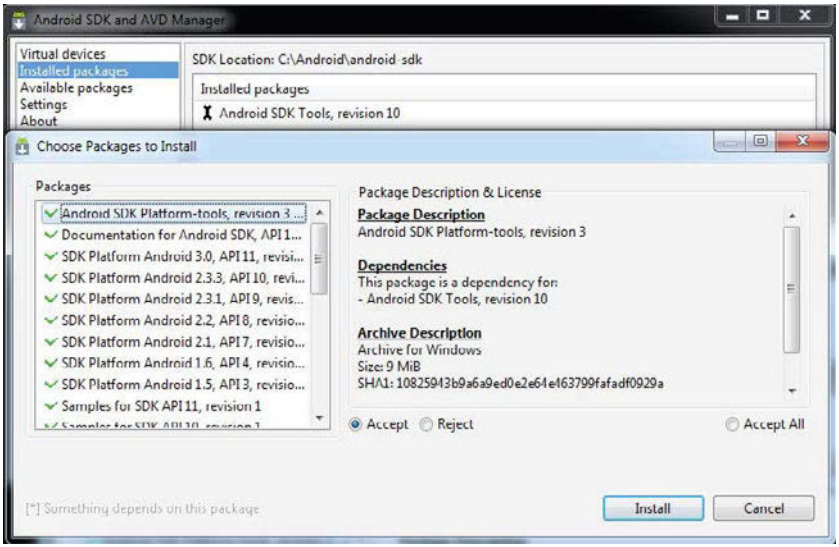


Рис. 1.8. Выбор и установка компонентов платформы Android

8. Спустя несколько минут, когда все компоненты будут загружены, появится сообщение, предлагающее перезапустить службу ADB (Android Debug Bridge – отладочный мост для Android). Ответьте щелчком на кнопке **Yes** (Да).
9. Закройте приложение.
10. Теперь распакуйте ZIP-архив с пакетом *Android NDK* в каталог установки (например, `C:\Android\android-ndk`). Опять же не используйте пробелы в именах промежуточных и конечного каталогов, куда выполняется установка (иначе могут возникать различные проблемы с утилитой **Make**).
    Чтобы упростить доступ к утилитам Android из командной строки, определите следующие переменные окружения:

11. Откройте окно **Environment Variables** (Переменные окружения), как это делалось выше. В список **System variables** (Системные переменные) добавьте *переменные окружения* `ANDROID_SDK` и `ANDROID_NDK`, значениями которых являются пути к соответствующим каталогам.
12. Добавьте `%ANDROID_SDK%\tools`, `%ANDROID_SDK%\platform-tools` и `%ANDROID_NDK%` через точку с запятой в конец переменной окружения `PATH`.
13. Все переменные окружения Windows должны автоматически импортироваться утилитой Cygwin при запуске. Убедитесь в этом, открыв терминал Cygwin и проверив доступность NDK, как показано на рис. 1.9:

```
$ ndk-build --version
```

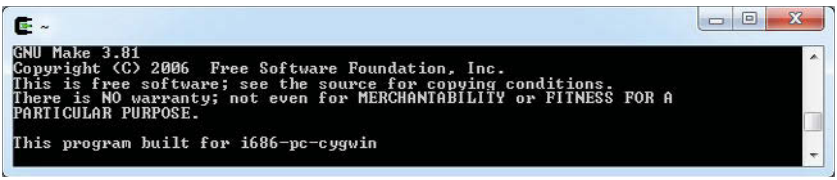


Рис. 1.9. Результат выполнения команды ndk-build --version

14. Теперь проверьте версию Ant, чтобы убедиться в его работоспособности под управлением Cygwin, как показано на рис. 1.10:

```
$ ant -version
```

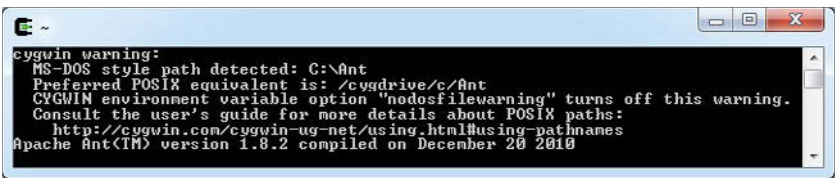


Рис. 1.10. Результат выполнения команды ant -version

При первом запуске утилита Cygwin должна вывести неожиданное предупреждение, сообщающее о том, что путь определен в стиле MS-DOS, а не POSIX. В действительности Cygwin эмулирует пути и использует формат `/cygdrive/<Буква диска>/<Путь к каталогу с использованием прямых слешей>`.

Например, если предположить, что утилита Ant установлена в каталог `c:\ant`, путь к ней будет выглядеть так: `/cygdrive/c/ant`.

15. Исправим эту проблему. Перейдите в свой каталог установки Cygwin. Там вы должны найти каталог с именем `home/<ваше имя пользователя>`, содержащий файл `.bash_profile`. Откройте его в текстовом редакторе.
16. В конец этого сценария добавьте преобразование значений переменных окружения Windows в переменные Cygwin с помощью утилиты **cygpath**. Переменную `PATH` не требуется преобразовывать, так как она преобразуется утилитой Cygwin автоматически. Не забудьте добавить символы (`) (чтобы выполнить одну команду внутри другой), которые в командной оболочке Bash имеют иное назначение, отличающееся от апострофов (') (используемых для определения значений переменных). Например, файл `.bash_profile` для этой книги содержит следующие строки:

```
export ANT_HOME=`cygpath -u "$ANT_HOME"`
export JAVA_HOME=`cygpath -u "$JAVA_HOME"`
export ANDROID_SDK=`cygpath -u "$ANDROID_SDK"`
export ANDROID_NDK=`cygpath -u "$ANDROID_NDK"`
```

17. Повторно откройте окно **Cygwin** и снова проверьте версию Ant. На этот раз не должно появиться никаких предупреждений, как показано на рис. 1.11:

```
$ ant -version
```

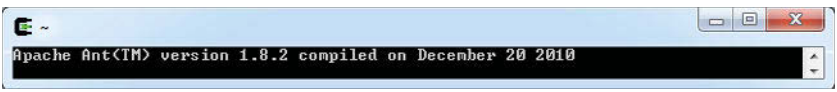


Рис. 1.11. Предупреждение больше не выводится

Что получилось?

Мы загрузили и установили пакеты *Android SDK и NDK* и с помощью переменных окружения обеспечили доступ к ним из командной строки.

Мы также запустили панель управления **Android SDK and AVD Manager**, предназначенную для управления установкой и обновлением компонентов SDK и средств эмуляции. С ее помощью можно

обновлять версии SDK API и добавлять в окружение разработки сторонние компоненты (такие как эмулятор Samsung Galaxet и др.) без переустановки Android SDK.

При подключении к Интернету через прокси-сервер во время выполнения пункта 7 можно столкнуться с проблемами. На этот случай в панели управления **Android SDK and AVD Manager** имеется раздел **Settings** (Настройки), где можно определить параметры подключения к прокси-серверу.

В пункте 16 описан порядок преобразования путей из формата, используемого в Windows, в формат, используемый в Cygwin. Этот формат, который первое время выглядит несколько необычно, используется утилитой Cygwin для представления путей в ОС Windows в формате путей в ОС Unix. Каталог `cygdrive` своим назначением напоминает каталог `mount` или `media` в Unix и содержит подкаталоги с именами, соответствующими дискам, доступным в Windows, со смонтированными к ним файловыми системами.

> **Пути в Cygwin.** При использовании путей в формате Cygwin следует помнить, что они должны содержать только символы прямого слеша, а буква, определяющая диск, должна замещаться строкой `/cygdrive/[буква диска]`. Но будьте внимательны – имена файлов в Windows и Cygwin не чувствительны к регистру символов, в противоположность настоящим системам Unix.

Подобно любой ОС Unix, в *Cygwin* имеется корневой каталог с именем /. Но так как в Windows нет настоящего корневого каталога, Cygwin имитирует его, отображая в собственный каталог установки. Чтобы увидеть содержимое этого каталога, введите в командной строке Cygwin следующую команду:

```
$ ls /
```

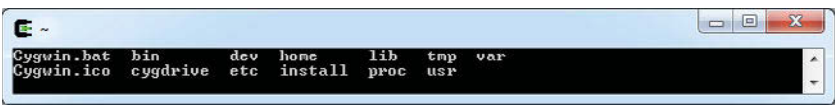


Рис. 1.12. Содержимое корневого каталога в Cygwin

Эти файлы находятся в каталоге Cygwin (кроме каталога `/proc`, который существует только в оперативной памяти). Это объясняет, почему мы редактировали файл `.bash_profile` непосредственно

в домашнем каталоге, располагающемся внутри каталога установки Cygwin.

Утилиты, входящие в состав пакета Cygwin, обычно работают с путями в формате Cygwin, но почти все они способны воспринимать пути в формате Windows. Благодаря этому мы могли бы не включать преобразование в файл `.bash_profile` (ценой вывода предупреждающего сообщения), но более естественным способом работы с Cygwin, позволяющим избежать возможных ошибок в будущем, является использование путей в формате Cygwin. Однако утилиты Windows (например, `java.exe`) не поддерживают пути в формате Cygwin, поэтому при их использовании требуется выполнять обратное преобразование. Для этих целей утилита `cygpath` позволяет использовать следующие ключи:

- -u: для преобразования путей из формата Windows в формат Unix;
- -w: для преобразования путей из формата Unix в формат Windows;
- -p: для преобразования списков путей из формата (в которых элементы отделяются друг от друга точкой с запятой в Windows и двоеточием в Unix).

При выполнннии пункта 16, на этапе редактирования файла `.bash_profile`, могут возникать различные сложности: в окне редактора могут отображаться странные символы в квадратиках, и все содержимое файла может располагаться в одной длинной строке! Это обусловлено использованием кодировки символов, принятой в Unix. Поэтому для редактирования файлов в Cygwin желательно использовать редакторы, совместимые с Unix (такие как Eclipse, PSPad или Notepad++). Если вы уже столкнулись с проблемами, попробуйте изменить в редакторе настройку преобразования символов конца строки (такая настройка имеется в Notepad++ и PSPad) или примените утилиту командной строки **dos2unix** (входит в состав пакета Cygwin) к файлу, вызвавшему проблемы.

Символ возврата каретки в Cygwin. Для обозначения концов строк в текстовых файлах в ОС Unix используется одиночный символ перевода строки (более известный как `\n`), тогда как в Windows используется последовательность из символа возврата каретки (CR или `\r`) и символа перевода строки. С другой стороны, в MacOS используется единственный символ возврата каретки. Способ обозначения конца строки, принятый в Windows, может вызвать немало проблем, если использовать его в сценариях командной оболочки Cygwin, поэтому их следует сохранять в формате Unix.

Примечание. На этом заканчивается раздел, описывающий настройку окружения в Windows. Если вы не пользуетесь Mac OS или Linux, то можете сразу перейти к разделу «Настройка среды разработки Eclipse».

Настройка в Mac OS X

Компьютеры компании Apple и Mac OS X славятся простотой и удобством использования. И если честно, это на самом деле так, когда речь заходит о разработке программ для платформы Android. В действительности Mac OS X основана на операционной системе Unix, прекрасно приспособленной для использования инструментов из NDK, и по умолчанию уже содержит последнюю версию JDK. Mac OS X содержит в себе практически все, что необходимо, кроме инструментов для разработки, которые требуется устанавливать отдельно. В состав этих инструментов разработки входят среда разработки XCode IDE, множество утилит для разработки в Mac, а также некоторые утилиты Unix, такие как Make и Ant.

Время действовать — подготовка Mac OS X для разработки на платформе Android

Все инструменты разработки входят в состав пакета XCode (последней на момент написания этих строк была версия 4). Получить этот пакет можно четырьмя способами, как описывается ниже:

- если у вас есть установочный диск Mac OS X, откройте его и найдите пакет XCode;
- установочный пакет XCode можно также бесплатно получить в репозитории AppStore (но такая возможность появилась совсем недавно и может исчезнуть в будущем);
- установочный пакет XCode можно загрузить с веб-сайта Apple, при наличии платной подписки на получение программного обеспечения, по адресу http://developer.apple.com/xcode/;
- старую версию 3, совместимую с инструментами разработки для платформы Android можно бесплатно получить в виде образа диска на той же самой странице, зарегистрировав бесплатную учетную запись разработчика Apple.

Используйте способ, наиболее подходящий для вашего случая. А теперь установите XCode:

1. Отыщите установочный пакет XCode и запустите его. Установите флажок **UNIX Development** (Разработка для UNIX) в диалоге настройки. Завершите установку. Вот и все!

2. При разработке с использованием Android NDK нам потребуется инструмент сборки Make. Откройте окно терминала и убедитесь в работоспособности этой утилиты, как показано на рис. 1.13:

```
$ make --version
```

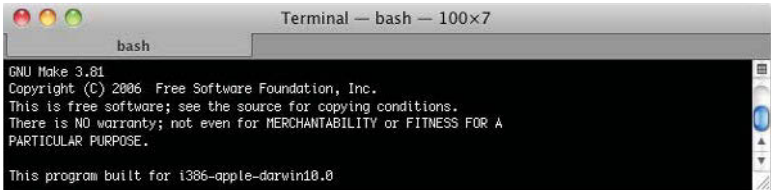


Рис. 1.13. Проверка работоспособности утилиты Make

3. Для работы Eclipse и компиляции программного кода на языке Java в байт-код необходим пакет Java Development Kit. Убедитесь в работоспособности пакета JDK, установленного в Mac OS X по умолчанию, как показано на рис. 1.14:

```
$ java -version
```

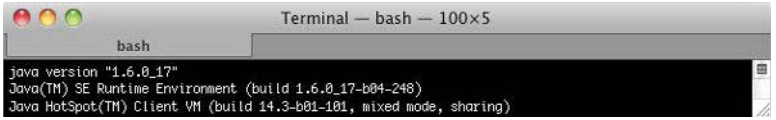


Рис. 1.14. Проверка работоспособности пакета JDK

4. Для компиляции проектов из командной строки пакет Android SDK поддерживает Ant – утилиту на языке Java, позволяющую автоматизировать процесс сборки. Также в окне терминала убедитесь, что утилита Ant установлена в системе, как показано на рис. 1.15:

```
$ ant -version
```

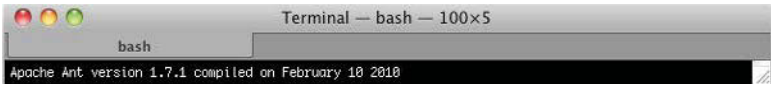


Рис. 1.15. Проверка работоспособности утилиты Ant