Лекция 3. Разработка приложений на платформе Android

Процесс разработки приложений на платформе Android и под Linux. Среда разработки Eclipse, инструменты Android SDK, плагин ADT, элементы пользовательского интерфейса и layouts в Android API. Отладка в Eclipse и в консоли Comms Services. Процесс разработки приложений под Android и Linux на платформе Intel Atom. Интеловские инструменты для разработки программного обеспечения для мобильных устройств.

3.1. Введение

В последнее время наряду с ноутбуками и смартфонами набирает популярность еще одна разновидность мобильных компьютеров – планшетные. Наиболее известным примером такого компьютера является Apple iPad. От ноутбуков планшеты отличает отсутствие традиционной клавиатуры, от смартфонов – большие размеры. Они ориентированы в первую очередь на развлечение пользователя, пассивное восприятие информации. В линейке аппаратных платформ на Intel Atom планшеты занимают одно из главных мест.

В двух последующих главах мы рассмотрим аппаратные особенности планшетов с точки зрения прикладного программиста.

Во-первых, планшет обладает набором датчиков, например, датчиком ориентации, освещения, акселерометром и т.п. Датчики, в свою очередь, позволяют реализовать более удобный, более интуитивный интерфейс управления программой на планшете, который был бы невозможен на нетбуке.

Во-вторых, это сенсорный экран, который в планшете заменяет почти все традиционные устройства ввода, такие как «мышь» и клавиатура. В случае «мыши» эти замена почти эквивалента по природе самого сенсорного экрана – оба они являются pointing device, и мы сосредоточимся как раз на этом случае.

Кроме аппаратных особенностей, у планшетов есть особенности пользовательского интерфейса, для которого и исчезновение клавиатуры, и изменение размера экрана критичны. Эти особенности также являются очень существенными, но мы их касаться не будем.

В этой главе мы познакомимся с общими средствами разработки приложений под Android и под Linux.

3.2. Среда разработки Eclipse

Перед тем, как приступать к созданию приложений на Android, необходимо выбрать подходящий инструментарий разработки и установить соответствующий Android SDK.

В качестве инструментария мы будем использовать Eclipse IDE. Конечно, можно выбрать любую другую платформу разработки, например Intelij IDEA, но лучше всего подходит Eclipse, так как он превосходно «заточен» для Java разработчиков.

Убедитесь в том, что ваше ОС удовлетворяет системным требованиям Android SDK, описанным в http://developer.android.com/sdk/ requirements.html.

Установите JDK – Java Development Kit, который можно скачать по ссылке: http://www.oracle.com/technetwork/java/javase/downloads/ index.html, а также проверьте, совместима ли ваша версия Eclipse с Android SDK. Если нет, то лучше скачайте последнюю версию http://www.eclipse.org/downloads/.

3.3. Android SDK

Скачать последнюю версию Android SDK можно по адресу http://developer.android.com/sdk/index.html.

Android SDK представляет собой средство разработки приложений под ОС Android. Android SDK распространяется в виде ZIP или EXE файлов, который распаковываются в папку на жестком диске.

В Android SDK входят:

- android.jar – файл архива Java, содержащий все классы SDK Android, необходимые для создания приложений;

- documentation.html и каталог docs – документация SDK предоставляется локально и через Интернет. В основном она выполнена в формате Javadocs, что позволяет легко ориентироваться во множестве пакетов SDK;
- каталог инструментов – содержит все инструменты командной строки для создания Android-приложений;
- USB-driver – каталог, содержащий все необходимые драйвера для подключения к поддерживающим Android устройствам.

Начальный пакет Android представляет только основные инструменты. Остальные компоненты при необходимости можно скачать и установить (например, GoogleAPI).

3.4. Плагин ADT для Eclipse

Android предлагает плагин ADT для Eclipse IDE, который обеспечивает разработчиков эффективными инструментами создания Android-приложений. Он расширяет богатые возможности ОС Eclipse новыми полезными утилитами. К примеру, после установки ADT на панелях toolbar и menubar появятся «контролы», позволяющие быстро создавать проект.

Для установки ADT, необходимо открыть панель расширения среды (InstallNewSoftware), а затем в появившемся окне добавить новый путь https://dl-ssl.google.com/android/eclipse/.

Далее Eclipse выполнит поиск требуемых данных, и в выпадающем списке вы сможете выбрать ADT и установить его, нажав Install. Запустится процесс инсталляции плагина, если в ходе установки возникнут ошибки, то, возможно, что указанный ресурс более по каким-то причинам недоступен, либо вы используете старую версию Eclipse, либо не установили JDK.

Запустите AVD Manager прямо из Eclipse (Window->Android SDK and AVD Manager). На экране появится окошко, содержащее слева список из трех основных компонентов: виртуальные устройства, установленные и доступные пакеты. В установленных пакетах находятся все компоненты, установленные по умолчанию или необходимости. С помощью вкладки «доступные пакеты» можно автоматически скачивать и устанавливать необходимые решения. К примеру, мы можем установить Google API's любой

версии, с целью использования в разрабатываемом приложении сервисов Google (карты, поиск).

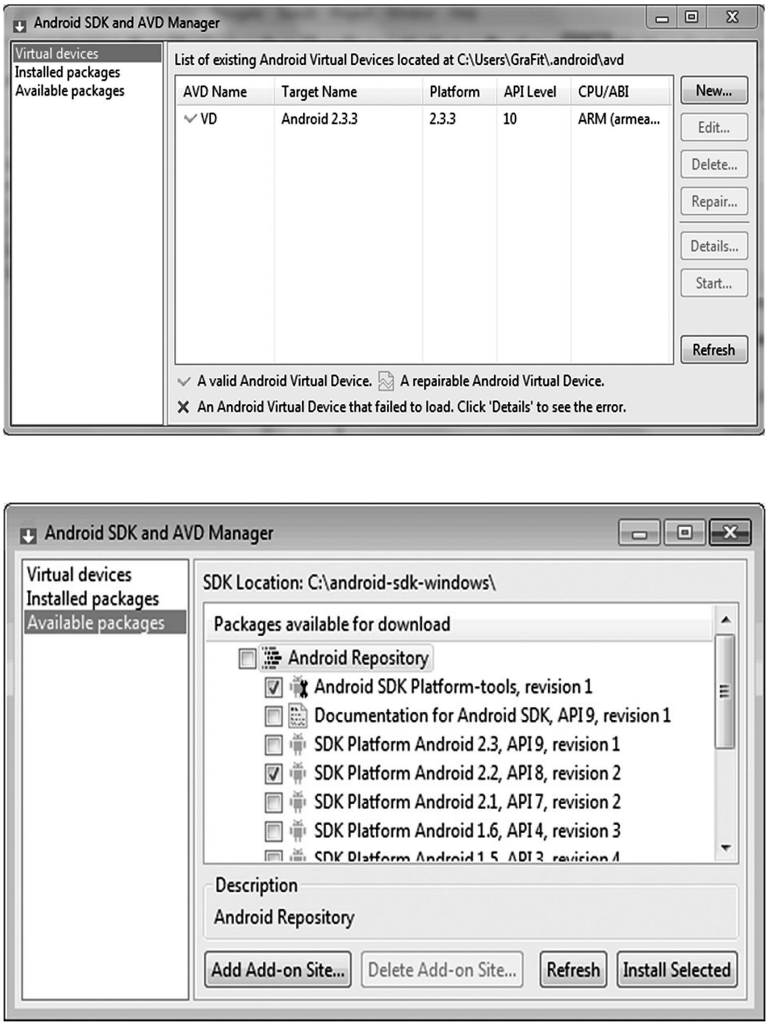

Рис. 3.4.1. Android API для элементов UI

Прежде чем писать приложение с использованием Android API's необходимо создать виртуальное устройство, которое позволит тестировать написанное. Чтобы создать виртуальное устройство в

главном окне Android SDK and AVD Manager, кликните на кнопку New. Появится диалоговое окно, которое позволяет настроить виртуальную машину. Заполняйте поля сверху вниз, всплывающие мини-окошки помогут вам при заполнении. На этом установка Андроида полностью завершена, теперь можно приступать к написанию приложений.

Кнопки являются неотъемлемым компонентом любого интерактивного приложения. ОС Android обладает превосходными средствами создания и управления кнопками. Кнопка описана классом android.widget.Button, который является суперклассом для всевозможных типов кнопок (флажковая кнопка, радиокнопка, переключатель).

У кнопки определено несколько конструкторов:

→ Button(Context context);

→ Button(Context context, AttributeSetattr);

→ Button(Context context, AttributeSetattr, intdefStyle);

Context – это класс содержащий информацию о среде приложения и предоставляющий доступ к различным системным ресурсам и классам.

Поля текстового ввода. Android.widget.EditText– является небольшой облицовкой вокруг TextView, позволяющей создавать редактируемый текст.

EditText имеет несколько конструкторов:

→ EditText(Contextcontext);

→ EditText(Contextcontext, AttributeSetattr);

→ EditText(Context context, AttributeSetattr, intdefStyle);

Класс EditText содержит несколько методов, рассмотрим некоторые из них:

→ public void selectAll(Spannable); метод фиксирует целиком в качестве текста, указанный ресурс. Spannable — это интерфейс для организации текста в виде блоков (кусков текста).

→ public Editable getText(); возвращает текст, отображаемый TextView. Editable интерфейс для описания динамического текста, содержимое и разметка которого могут быть изменены.

→ public void setText(CharSequence, TextView.BufferType); устанавливает текст, отображаемый в TextView. Первый параметр определяет текст, второй − способ его хранения: динамический текст, нормальный текст(default), блочный текст.

Списки. Android.widget.ListView – класс отображающий набор элементов в виде списка с возможностью прокрутки. Элементы ассоциированные с этим представлением описываются классом ListAdapter.

ListAdapter – наследует базовый класс Adapter и служит мостом между данными и ListView. Часто данные могут быть представлены курсорами, но необязательно. Удобство в том, что ListView может отображать любые данные, лишь бы они были завернуты в ListAdapter. ListAdapter имеет несколько подклассов (ArrayAdapter, BaseAdapter,..), которые предназначены для различных целей. Приведем пример использования ArrayAdapter.

ArrayAdapter – представляет данные в виде массива и добавляет удобный функционал (добавление, удаление, поиск) для работы с ними.

→ ArrayAdapter(Context, TextViewResourceId);

→ ArrayAdapter(Context, TextViewResourceId, T[] objects);

TextViewResourceId – ссылка на xml-файл представления, данных списка. Параметр T[] objects задает отображаемые данные.

LinearLayout. Линейный менеджер размещения используется для отображения элементов в виде линейных списков, горизонтальных или вертикальных. Причем можно использовать несколько Layouts одновременно, в этом случае они будут делить экран.

→ LinearLayout(Context context);

→ LinearLayout(Context context, AttributeSet);

→ LinearLayout(Context context, AttributeSet , intdefStyle);

Рассмотрим основные методы:

generateLayoutParams(AttributeSet attr) – задает параметры layout (ширина, длина, фон, притяжение);

getOrientation() – задает направление размещения.

RelativeLayout. Этот менеджер размещения используется в том случае, если необходимо размещать элементы относительно какого-

то виджета. Позиция размещения задается в виде: элемент X расположить ниже элемента Y.

RelativeLayout очень мощное средство расположения элементов, не зависящее от размеров окна приложения и не привязанное к какой-то конкретной позиции. Таким образом, с помощью RelativeLayout мы получаем «резиновый», легко программируемый интерфейс. Конструкторы те же самые, что и у LinearLayout.

TableLayout. Данный менеджер размещения, располагает элементы в табличном виде по строкам и колонкам. TableLayout состоит из элементов разметки TableRow, определяющих строку в которой будет содержаться виджет. TableLayout не отображает границы ячеек, из которых состоит. Каждая строка TableLayout имеет ноль или более ячеек, а каждая ячейка может хранить объект для отображения (ViewObject). Число столбцов в таблице определяется самой длинной по количеству ячеек строкой. Таблица может содержать пустые ячейки. Ячейки могут охватывать несколько столбцов, также как и в HTML. Ширина колонки есть максимум по длинам ячеек принадлежащих этому столбцу.

В конкретных случаях, можно настроить таблицу под собственные требования, в этом помогут стандартные методы, жестко задающие параметры таблицы.

TabLayout. Иногда нам приходиться описывать несколько различных отображений в одном окне, обеспечивая между ними быстрое и комфортное переключение. В таком случае необходимо использовать вкладочный менеджер размещения (tablayout). Чтобы создать tablayout, необходимо выполнить следующее: создать TabHost либо в виде xml описания, либо непосредственно кодом в программе. Этот элемент является контейнером для вкладок (табов) оконного вида.TabHost должен иметь два дочерних элемента TabWidget и FrameLayout.

TabWidget – элемент представляющий список вкладок FrameLayout, используемый для отображения содержимого каждой вкладки.

Существует два способа использования TabLayout:

1) Можно создать несколько представлений в одной Activity и переключаться между ними, кликая по табам.

2) Можно создать несколько Activity, описывающих различные представления, а в главном Activity обеспечить корректное переключение.

3.5. Отладочная сессия в Eclipse и консоли

Android SDK обеспечивает большой набор инструментов, которые необходимы для отладки приложений. У вас должен быть JDWP-совместимый отладчик (Java Debug Wire Protocol (JDWP) – это протокол, используемый для обмена данными между отладчиком и виртуальной машиной Java, которую он и отлаживает).

Основные компоненты среды отладки Android:

ADB (AndroidDebugBridge) – выступает в качестве посредника между устройством и программой.

Dalvik Debug Monitor Service (DDMS) – это элемент плагина Android, используемый для анализа работы ВМ. Последовательно отображает поток информации, которая описывает каждый шаг работы виртуальной машины.

Device or Android Virtual Device (AVD**).** Ваше приложение должно выполняться на устройстве или в AVD так чтобы оно могло быть отлажено. Демон adb устройства работает на устройстве или эмуляторе, и предоставляет средства для демона adb хоста для предоставления связи с устройством или эмулятором.

ADB – клиент-серверное приложение, состоящее из трех компонентов (рис. 3.5.1).

Клиент – запускается на машине разработчика. Клиент можно запустить из командной строки при помощи команд посылаемых adb. Другие инструменты Android вроде плагина ADT и DDMS тоже создают adb-клиенты.

Сервер – запускается на машине разработчика в виде фонового процесса. Сервер управляет соединениями между клиентами и adb-сервисом запущенным на эмуляторе или устройстве.

Сервис – фоновый процесс, который запускается на каждом эмуляторе или устройстве.

Debug	`logcat [<option>] [<filter-specs>]`	Prints log data to the screen.	
	`bugreport`	Prints `dumpsys`, `dumpstate`, and `logcat` data to the screen, for the purposes of bug reporting.	

	`jdwp`	Prints a list of available JDWP processes on a given device.	You can use the `forward jdwp:<pid>` port-forwarding specification to connect to a specific JDWP process. For example:
			`adb forward tcp:8000 jdwp:472`
			`jdb -attach localhost:8000`

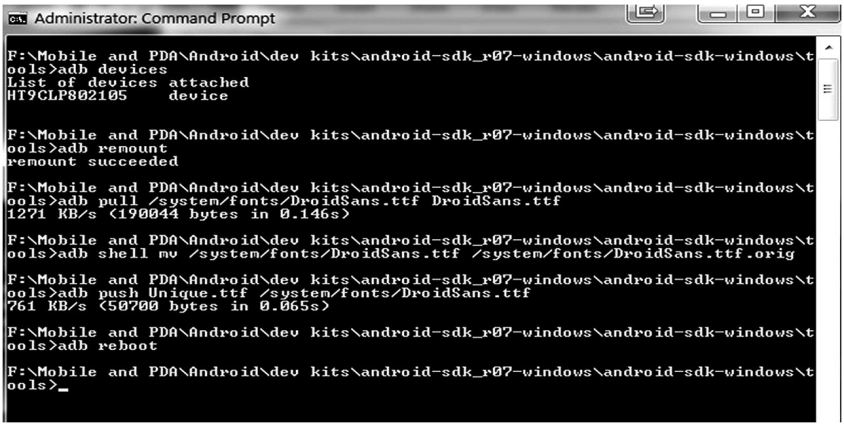

Рис. 3.5.1. Пример использования клиент-серверного приложения ADB

LogCat. Система ведения лога в Android обеспечивает механизм для сбора и просмотра системных отладочных сообщений. Логи из различных приложений и элементов системы Android собираются воедино и их затем можно просматривать и фильтровать посредством команды logcat (рис. 3.5.2).

Logcat можно использовать для просмотра и слежения за содержанием буферов системного лога. Формат использования:

```
[adb] logcat [<option>] ... [<filter-spec>] …
```

Фильтрация вывода лога: Каждое сообщение лога в Android имеет *Тэг* и *приоритет*.

Тэг – это строка указывающая компонент системы, от которого принято сообщение (например: View для системы view).

Приоритет – имеет одно из нижеследующих значений (в порядке от меньшего к большему):

V – Verbose (низший приоритет); D – Debug; I – Info; W – Warning; E – Error; F – Fatal; S – Silent (наивысший приоритет, при котором ничего не выводится).

Выражения фильтра позволяют указать системе нужные комбинации <тэг> и <приоритет>, остальные сообщения система не выводит. Выражения фильтра имеют следующий формат <тэг>:<приоритет>.

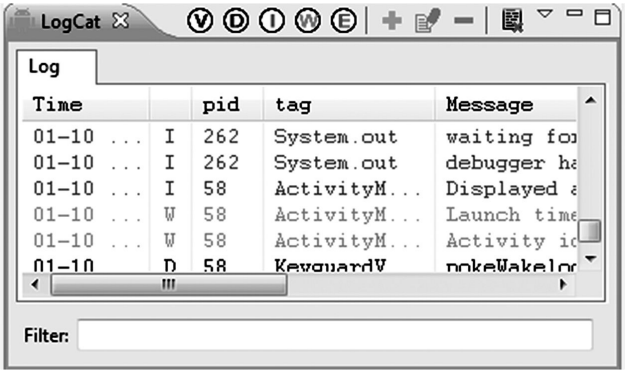

Рис. 3.5.2. Пример использования LogCat

Отлаживать приложения в Eclipse гораздо проще, чем в консоли, благодаря использованию standpoints. Но проблема отладки через Eclipse может возникнуть, если вы отлаживаете приложения, запущенные на эмуляторе, а не на устройстве. Эмулятор работает довольно медленно, а при отладке каждый переход работает более 3 секунд, это очень долго. Так, если вы отлаживаете большое приложение и при отладке наткнулись на цикл, содержащий 100 итераций, то чтобы его пройти вам потребуется более length(circle)*3*100 секунд. В таких случаях лучше использовать отладку с фильтрами через консоль и выводить отладочную информацию в файл с дальнейшим выявлением ошибок.

При отладке в Eclipse пользуйтесь утилитой

```
LogCat (Window->Show view->Android->LogCat)
```

которая, по сути, является оболочкой команд отладки через консоль.

3.6. Инструменты Intel для разработки под Linux

Корпорация Intel предлагает свой большой набор инструментов для разработки под Linux, наиболее важными среди которых являются:

- Intel C++ Compiler — оптимизирующий компилятор языков C\C++ для платформы x86. icc использует особые возможности и преимущества процессоров Intel и в большинстве случаев даёт значительно более производительный код в сравнении с gcc. Значительным преимуществом icc является то, что он использует синтаксис командной строки, похожий на синтаксис командной строки gcc. Это позволяет переводить достаточно большие проекты, использующие gcc, на компиляцию icc, без особых трудозатрат. В частности, компилятором icc было успешно скомпилировано ядро Linux.
- Intel® VTune™ Performance Analyzer — профилировщик. Позволяет изучить производительность отдельных участков кода и общую производительность, выявляя узкие места. В отличие от инструментов, симулирующих выполнение кода на виртуальном процессоре, VTune™ выполняет код на CPU от Intel, используя для измерений многочисленные отладочные регистры процессора. В этом VTune™ аналогичен OProfile, однако работа с последним значительно менее наглядна и не вполне раскрывает богатые отладочные функции процессоров Intel.

3.7. Свободные IDE для разработки программного обеспечения на C/C++ под Linux

В UNIX/Linux есть большое количество инструментов для разработки проектов на C/C++. Некоторые разработчики

предпочитают традиционные Vi/Emacs/утилиты командной строки, а другие — современные средства разработки.

При создании больших проектов, использование средств разработки особенно оправдывает себя. Как правило, они дают разработчику возможности автоматического дополнения кода, свертывания кода, подсветки синтаксиса, предоставляют шаблоны кода, встроенный компилятор и отладчик. В особенности он помогает людям справиться с несколькими файлами при использовании GUI, в отличие от утилит командной строки или традиционных редакторов без GUI.

Перечислим наиболее популярные IDE для разработки на C/C++ под Linux:

- `QtCreator` — кроссплатформенная IDE для работы с фреймворком *Qt*, разработанная *Qt* Software.IDE для работы с фреймворком *Qt*, разработанная *Qt* Software. Эта IDE была специально разработана для работы с *Qt*, имеет возможности расширения плагинами, встроенный `QtDesigner` и `QtAssistant` и графический фронтенд для `gdb`.
- `NetBeans` — свобаодная интегрированная среда разработки приложений на языках программирования Java, JavaFX, Ruby, Python, PHP, JavaScript, C++, Ada и другие. NetBeans поддерживает рефакторинг, профилирование, выделение синтаксических конструкций цветом, автодополнение набираемых конструкций на лету, множество переопределенных шаблонов кода, удаленную отладку и др. В NetBeans поддерживается UML, SOA, языки программирования Ruby, а также средства для создания приложений на J2ME для мобильных телефонов. NetBeans IDE поддерживает плагины, позволяя разработчикам расширять возможности среды.
- `Eclipse` — в первую очередь платформо-независимая Java IDE, нацеленная на групповую разработку: среда интегрирована с системами управления версиями — CVS в основной поставке, для других систем (например, Subversion, MS SourceSafe) существуют плагины. Второе назначение Eclipse — служить платформой для разработки новых расширений, чем он и завоевал популярность: любой разработчик может расширить Eclipse своими модулями. Уже существуют C/C++ Development Tools (CDT), разрабатываемые инженерами QNX совместно с

IBM, и средства для языков COBOL, FORTRAN, PHP и пр. от различных разработчиков. Множество расширений дополняет среду Eclipse менеджерами для работы с базами данных, серверами приложений и др.

- `Anjuta` — это гибкая интегрированная среда разработки (Integrated Development Environment — IDE) для языков C и C++ в GNU/Linux, которая была написана для GTK/GNOME и включает ряд мощных средств для программирования. Среди них — средства управления проектом, мастера приложений, встроенный интерактивный отладчик, мощный редактор исходного кода со средствами просмотра и подсветкой синтаксиса.
- `Kdevelop` — свободная среда разработки программного обеспечения для Unix-подобных систем. Она поддерживает подсветку исходного кода с учетом синтаксиса используемого языка программирования; менеджер проектов, для проектов разного типа, таких как Automake, qmake для проектов базирующихся на Qt и Ant для проектов, базирующихся на Java; навигатор классов (Class Browser); front-end для GNU Compiler Collection; front-end для GNU Debugger; wizards для генерации и обновления определения классов и framework; автоматическую систему завершения кода (Си/C++); встроенную поддержку Doxygen; систему контроля версий.

3.9. Инструменты профилировки и отладки

Для разработки эффективного программного обеспечения часто приходится выполнять профилирование кода, которое включает в себя сбор характеристик работы программы,таких как время выполнения отдельных фрагментов (обычно подпрограмм), число верно предсказанных условных переходов, число кэш промахов и многое другое.

Перечислим основные средства для профилирования программ при разработке под ОС Linux:

- `GNU profiler (gprof)` используется для того, чтобы определить, сколько времени уходит на выполнение той или иной части программы, как часто вызываются те или иные

процедуры; для использования gprof необходимо компилировать программу со специальными опциями для включения «профилирования».

- `Valgrind` – инструментальное программное обеспечение, предназначенное для отладки утечек памяти, профилировки, построения дерева вызовов.
- `KCacheGrind` – графический анализатор вывода `Valgrind`.
- `strace` – утилита, выполняющая трассировку системных вызовов
- `OProfile` – профилировщик, использующий счётчики производительности процессора.

3.8. Разработка мобильных приложений

Разработка мобильных приложений – частный случай так называемой кросс-разработки. В случае кросс-разработки платформа, для которой пишется приложение (целевая платформа, например, Linux ARM в виде смартфона) отличается от платформы, на которой работает программист и создается исполняемый код для целевой платформы (платформа разработчика, например, Linux x86 в виде обычного PC). В случае ОС MeeGo аппаратная часть платформы может быть одинакова – x86 – но способ разработки все равно останется кроссплатформенный. Мы рассмотрим четыре составляющих кросс-платформенной разработки – создание исполняемого кода, его запуск, отладка и интегрированные среды разработчика.

Наиболее важной частью кросс-платформенной разработки является набор инструментов для создания исполняемого кода. В него входит компилятор, компоновщик, другие утилиты, набор библиотек для целевой платформы – этого достаточно, чтобы создать исполняемый файл для целевой платформы.

Для управления процессом сборки, документирования и т.п. можно использовать те же средства, что и для обычной разработки под Linux.

Запуск объектного кода может может быть осуществлен разными способами. Самый естественный – запуск непосредственно на целевом устройстве. Для этого требуется само целевое устройство,

подключенное к компьютеру разработчика по USB или Ethernet. Это наиболее надежный способ, но не всегда самый удобный. Финальное тестирование необходимо производить именно на целевом устройстве.

Альтернатива – установить целевую операционную систему на универсальную виртуальную машину на платформе разработчика. В качестве такой машины для MeeGo рекомендуется Virtual Box, также можно использовать VMWare. При этом надо учитывать, что по некоторым параметрам виртуальная машина может отличаться от реального устройства – например, по скорости работы, особенностям периферии и т. п.

Промежуточным вариантом является использование предоставляемого производителем целевой платформы эмуляторы. Фактически это разновидность виртуальной машины, которая, с одной стороны, максимально адаптирована для эмуляции целевой платформы, а с другой стороны, интегрирована со средствами разработки, например, IDE, которые также предоставляются производителем.

В настоящее время для запуска обычно не ограничиваются переносом на целевую платформу одного исполняемого файла, а формируют целый пакет со всеми зависимостями и правилами установки (например, APK для Android), который затем устанавливается на целевой платформе. С помощью такого же пакета финальная версия приложения попадает на устройства конечных пользователей. Это, с одной стороны, увеличивает переносимость приложения и облегчает запуск сложных приложений, с другой – слегка усложняет и удлиняет процедуру запуска.

При кросс-платформенной разработке процедура отладки приложения изменяется не сильно.

Еще одним популярным способом отладки при кросс-платформенной разработке является полный отказ от `gdb` и интенсивное использование логов.

Современные IDE, такие как `QtCreator` или `Eclipse`, поддерживают кросс-платформенную разработку. Эта поддержка может быть встроенной или реализовываться в виде плагинов. В любом случае IDE берут на себя львиную долю организационной работы, освобождая программисту время для творчества. Внешне

процесс кросс-разработки с использованием IDE практически не отличается от обычного.

Список литературы

1. Инструкция для разработчика под Android. http://developer.android.com/guide/index..html
2. Android APIs. http://developer.android.com/reference/packages.html
3. Rehman R.U., Paul C. The Linux Development Platform.