

**Секция «Современные технологии
проектирования и разработки
программного обеспечения»**

Обзор и перспективы развития архитектуры операционной системы Android

Android – операционная система, разработанная компанией Google и ее партнерами для использования на смартфонах, планшетных компьютерах. Поддержку и совершенствование системы в данный момент осуществляет альянс ИТ-компаний Open Handset Alliance. Платформа позиционируется создателями как открытая и предоставляет возможность независимым разработчикам распространять свое программное обеспечение через сервис Google Play (ранее именовался Android Market) [1-3].

Перед началом разработки любой программы необходимо разобраться в особенностях работы целевой платформы. Структуру ОС Android можно представить в виде иерархии, состоящей из четырех уровней (рис.1).

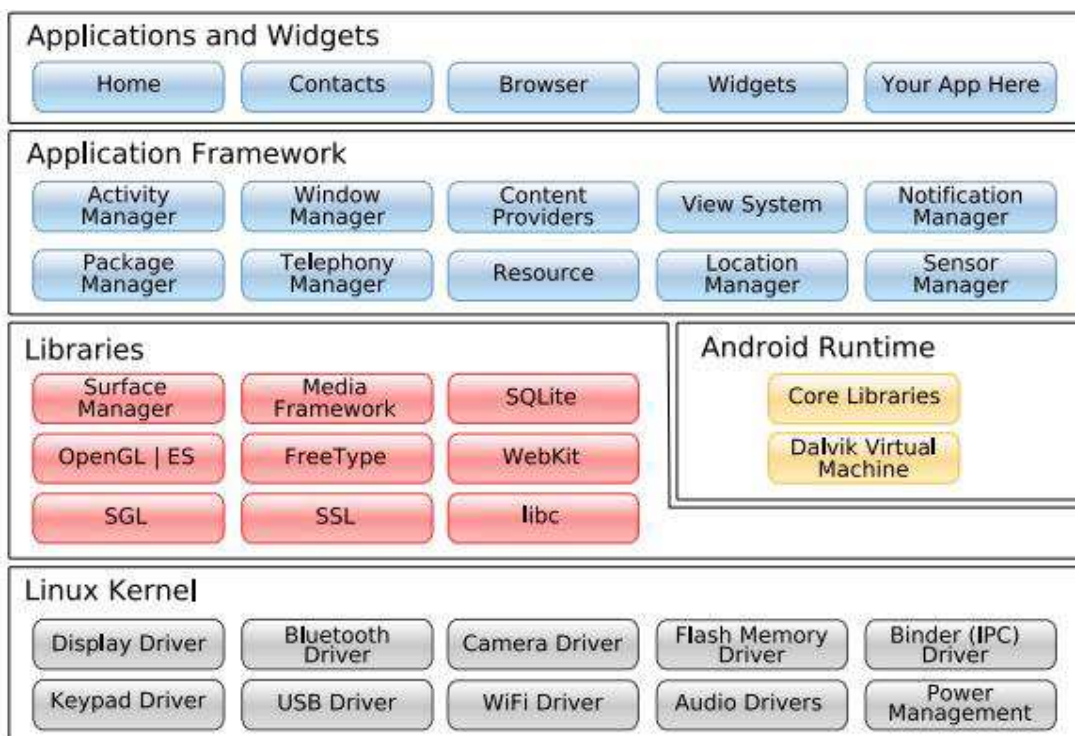


Рис. 1. Архитектура операционной системы Android

Первый уровень – ядро операционной системы. В данном случае – это несколько урезанная версия ядра Linux 2.6. Ядро взаимодействует с аппаратным обеспечением устройства, содержит драйверы для работы с экраном, камерой, Wi-Fi, Bluetooth и т.д.

Второй уровень – уровень библиотек. Содержит библиотеки, написанные и скомпилированные под конкретное программное обеспечение. Функционал этих библиотек используется различными частями операционной системы.

Третий уровень – уровень каркаса приложений. Включает набор систем и служб, необходимых для нормального функционирования любого приложения на платформе.

Четвертый уровень – уровень приложений. Включает базовое ПО платформы: клиенты электронной почты, календари, программы для спутниковой навигации. Платформа позволяет пользователю заменять стандартные приложения приложениями от сторонних разработчиков.

ОС Android продолжает развиваться разработчиками. За последние два года была введена единая оболочка для смартфонов, планшетов, улучшена многозадачность. Была проделана большая работа по улучшению производительности операционной системы, что позволило в бюджетном сегменте выпускать устройства, не уступающие по функциональности флагманом.

Литература

1. Голошапов А. Google Android: программирование для мобильных устройств. - СПб.: БХВ-Петербург, 2010. 448 с.
2. Роджерс Р., Ломбардо Д. Android. Разработка приложений. - М.: ЭКОМ Паблишерз, 2010. 400с.
3. Коматинэни С., Маклин Д., Хэшими С. Google Android: программирование для мобильных устройств Pro Android 2. - 1-е изд. - СПб.: Питер, 2011. 736 с.

А.А. Артемов
Научный руководитель: старший преподаватель Я.Ю.Кульков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: classicxe@ya.ru

Сравнительный анализ методов поисковой оптимизации

Поисковая оптимизация (англ. search engine optimization, SEO) — это комплекс мер направленных, на продвижение Интернет-ресурсов в “глазах” поисковых машин по определенным запросам. Чем выше сайт в поисковой выдаче тем, больше пользователей на него заходят.

Поисковая оптимизация проводится для привлечения посетителей на раскручиваемый Интернет-ресурс или для увеличения продаж, если это интернет-магазин.

При анализе эффективности работы Интернет-ресурса, стоит подумать о том, как попадают на него потенциальные пользователи (клиенты). Здесь есть несколько вариантов: кликнул на баннер (рекламу), найдут в каталоге или расскажет друг. Но по данным статистики более 80% посетителей попадают на Интернет-ресурсы с помощью поисковых машин. При всем этом 8 из 10 это самые что ни на есть целевые посетители данного Интернет-ресурса, которых интересует товары, представленные на нем или какой-либо другой материал. В интернете, часто публиковались отчеты, что пользователь не желает смотреть дальше второй страницы в поисковой выдаче, а это значит, если сайт не оптимизирован, то он к нам не придет.

Существует три метода поисковой оптимизации.

Белая оптимизация – поисковая оптимизация, с использованием только разрешенных и добросовестных методов. Этот комплекс мер по повышению посещаемости Интернет-ресурса основан, на анализе поведения целевых посетителей.

Серая оптимизация – поисковая оптимизация, делающая Интернет-ресурс неудобным для пользователя. В основном в статьях на ресурсе присутствует слишком много ключевых слов.

Черная оптимизация – поисковая оптимизация, с использованием недобросовестных и запрещенных поисковыми машинами способов.

Для поисковой оптимизации сайта используется различная техника написания текстов, для того что бы он был легок для восприятия пользователя, а также “нравился” поисковым машинам, то есть что бы содержал ключевые слова в нужном месте и определенном количестве. Так же SEO подразумевает внедрение ключевых слов в различные элементы страницы. Но и это далеко ещё не конец. Важно правильно выбрать ключевые слова, под ресурс.

Ключевые слова делятся на НЧ (низкочастотные) и ВЧ (высокочастотные). Начинать нужно с НЧ слов, так как, конкуренции здесь меньше, ведь только сделанному сайту конкурировать с гигантами сложно. После того, как мы определились с выбором ключевых слов, стоит проанализировать конкурентов по нашим поисковым запросам. Здесь все будет зависеть от ТИц (тематический индекс цитирования) и PR (page rank), а так же от возраста доменного имени конкурента, чем показатели выше, тем сложнее будет конкурировать.

Для успешной оптимизация сайта необходимо, учитывать различные условия. Текстовое оформление веб-страниц. Объем текста на странице – поисковые машины ценят, сайты богатые информацией. Оптимальным содержанием текста на странице принято считать от 2-20 тысяч символов. Число ключевых слов на странице – идеально от 2-4(5) ключевых слов или фраз на одной странице. Плотность ключевых слов – оптимальная плотность не более 7%. Расположение ключевых слов на странице, существует короткое правил, чем ближе слова находятся к вершине страницы, тем больший вес они получают. Стилистическое оформление текста – нужно пользоваться тегами <H> размер заголовка и выделение жирным шрифтом. Необходимо правильно заполнять тег TITLE, можно сказать это визитка Интернет-ресурса. Если на Интернет-ресурсе находятся фотографии или картинки, то нужно обязательно

заполнять тег <ALT> нужно это для того, если картинка не загрузилась, пользователю будет показан альтернативный текст, а поисковые машины этот текст запоминают. Ссылочный текст – текст, стоящий между тегами <A>. Релевантность ссылающихся страниц – это информация на странице ссылающаяся на Интернет-ресурс.

Так же в поисковой оптимизации имеет место быть правильная структура Интернет-ресурса. Ресурс нужно постоянно обновлять, размещать материалы по тематике сайта, это дает не только новый приток посетителей, но и общий вес ресурса в глазах поисковых машин, будет расти. В меню Интернет-ресурса желательно использовать ключевые слова. Каждую страницу нужно оптимизировать, под одну определенную фразу.

В итоге мы имеем, что оптимизировать ресурс нужно и только белыми методами. Это постоянный приток посетителей, постоянный рост ресурса. Но если человек хочет быстрого продвижения и недолгого, то это однозначно чёрный метод.

В.Б. Баканов
Научный руководитель: канд. техн. наук М.В. Усачев
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: bakanvladimir@yandex.ru

Обзор и перспективы мобильных решений Tegra

В современном мире существует большое количество электронных устройств, которые человек использует для облегчения своего труда, развлечений, получения новой информации и общения. Данные устройства постоянно совершенствуются. Вычислительные мощности современных смартфонов можно сравнить с мощностью компьютеров пяти – семи – летней давности. Они могут частично заменить настольные компьютеры, а некоторые из них, например планшетные компьютеры, иногда и превосходят их.

Одним из перспективных решений в области мобильных технологий является архитектура NVIDIA Tegra. Она представляет собой систему-на-кристалле (System-on-Chip), разработанную как платформа для производства мобильных интернет-устройств, таких как смартфоны, смартбуки, коммуникаторы, КПК и др. [1]

Основными отличиями Tegra от конкурентов являются: высокая производительность центрального и графического процессоров, низкое энергопотребление и совместимость с современными операционными системами.

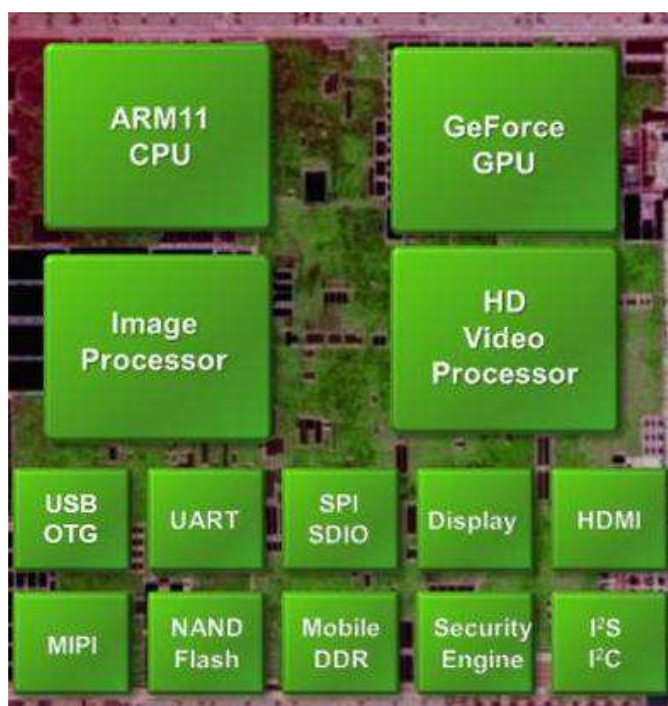


Рис. 1. Архитектура Tegra

Самым актуальным вопросом для переносных устройств является энергосбережение. В архитектуре Tegra (рисунок 1) применяются пять ядер в процессоре с архитектурой ARM11 (четыре основных с частотой до 1,5 ГГц и одно вспомогательное с частотой около 500 МГц). Маломощное ядро используется для управления электропитанием и может использоваться для решения простых задач (чтение электронных книг или прослушивание музыки), а остальные четыре ядра в это время отключаются. Благодаря этому устройства на Tegra работают без подзарядки в несколько раз дольше [2,3]. В архитектуре выделены три блока, отвечающие за обработку видеoinформации: графический процессор GPU, аппаратный ускоритель обработки изображений и видео-процессор для воспроизведения видеоряда в формате высокой четкости (HD).

Необходимо отметить, что сочетание высокой производительности при низком энергопотреблении является ключевым преимуществом при выборе разработчиками базовой архитектуры для построения современных мобильных решений. Дальнейшее развитие платформы Tegra будет способствовать увеличению мощности и функциональности мобильных устройств.

Литература

1. URL: <http://www.mobile-review.com/print.php?filename=/articles/2010/nvidia-tegra.shtml> [Электронный ресурс] Дата обращения 10.03.2013
2. URL: <http://www.nvidia.ru/object/tegra-ru.html> [Электронный ресурс] Дата обращения 11.03.2013
3. URL: <http://www.ferra.ru/ru/system/100662/> [Электронный ресурс] Дата обращения 10.03.2013

М. П. Борунова, М.С. Волков
Научный руководитель: старший преподаватель Д.В. Бейлекчи
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: kaf-eivt@yandex.ru

Исследование средств разработки программного обеспечения для ОС Android

Android – операционная система для смартфонов, планшетов и нетбуков. Компания Google приобрела разработчика программного обеспечения Androidinc. в 2005 году. Операционная система Android основана на модифицированном ядре Linux.

Android приложения включают в себя java-приложения и библиотеки, которые запускаются виртуальной машиной Dalvik с JIT компилятором. Библиотеки включают в себя систему управления, графику Open GL ES 2.0, библиотеку Web Kit, графический подсистему SGL, SSL и библиотеки Bionic. OS Android состоит из 12 миллионов строк кода, в том числе 3-х миллионов строк XML, 2.8 миллионов строк на C, 2.1 миллиона строк на Java и 1.75 миллиона строк на C++. [1]

В ходе выполнения работы проанализированы два основных средства разработки программного обеспечения для ОС Android: SDK на основе языка программирования Java и NDK на основе языка C/C++.

Разработку приложений для Android можно вести на языке Java (не ниже Java 1.5). Существует плагин для Eclipse – Android Development Tools (ADT), предназначенный для Eclipse версий 3.3-3.7. Также существует плагин для IntelliJ IDEA, облегчающий разработку Android-приложений, и для среды разработки Net Beans IDE, который, начиная с версии NetBeans 7.0, перестал быть экспериментальным, хоть пока и не является официальным. Кроме того, существует Motodev Studio for Android – комплексная среда разработки на базе Eclipse, позволяющая работать непосредственно с Google SDK [2].

Java – это объектно-ориентированный язык программирования, программы на котором транслируются в байт-код, выполняемый виртуальной машиной Java. Виртуальная машина обрабатывает байтовый код и передает инструкции оборудованию как интерпретатор. Достоинство подобного способа выполнения программ заключается в полной независимости байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы вызывают немедленное прерывание. Большинство архитектурных решений, принятых при создании Java, было направлено на сходство по синтаксису с языком C/C++. Программисты имеющие опыт программирования на C/C++, смогут быстро освоиться и начать писать Java-приложения.

Android Software Development Kit (SDK) содержит множество инструментов и утилит для создания и тестирования приложений. Например, с помощью SDK Manager можно установить Android API любой версии, а также проверить репозиторий на наличие доступных, но еще не установленных пакетов и архивов.

В 2009 году в дополнение к ADT был опубликован Android Native Development Kit (NDK) – пакет инструментариев и библиотек, позволяющий вести разработку приложений на языке C/C++. NDK рекомендуется использовать для разработки участков кода, критичных к скорости.

NDK обеспечивает:

- набор инструментов и создание файлов, используемые для создания собственных библиотек написаны на C и C++;
- способность включать соответствующие библиотеки в файлы пакетов прикладных программ (.arks), которые могут быть развернуты на устройствах Android.

Иерархия уровней приложений на ОС Android представлена на рисунке 1.

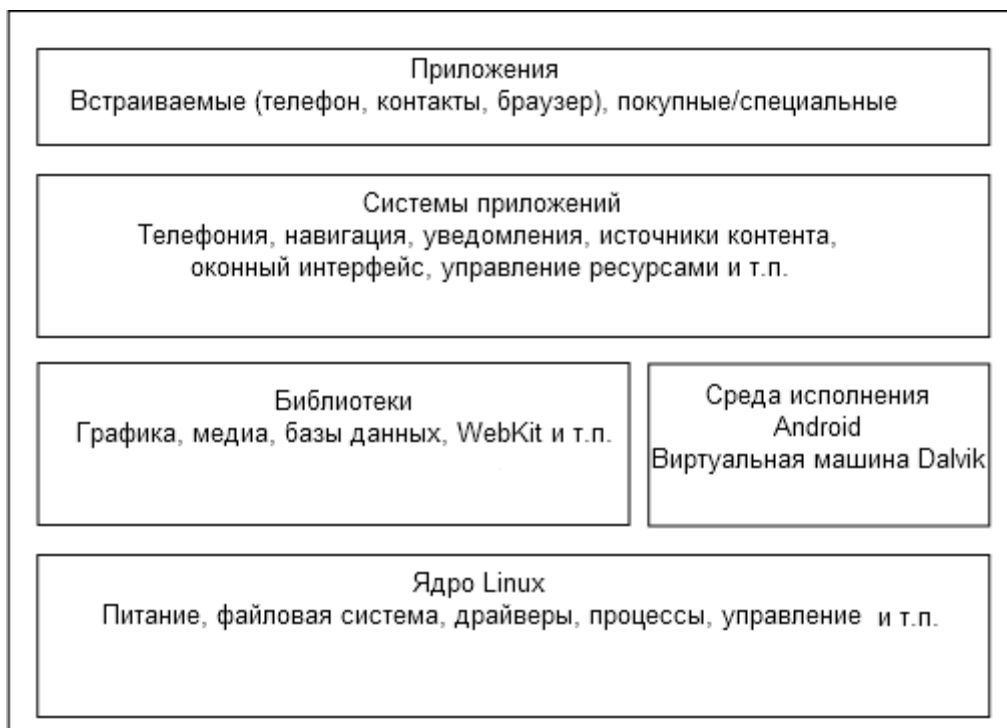


Рис. 1. Структура программного обеспечения ОС Android

Программное обеспечение разделяется на пять уровней [3]:

- уровень приложений (Applications) содержит набор базовых приложений: клиенты электронной почты и SMS, календарь, различные карты, браузер, программу для управления контактами и другие прикладные программы. Все данное программное обеспечение написано на языке Java;

- уровень каркаса приложений (ApplicationFramework) реализует возможность многократного использования компонентов ОС и приложений;

- уровень библиотек (Libraries) включает набор C/C++ библиотек, используемых различными компонентами ОС. Для программистов доступ к функциям этих библиотек реализован через использование ApplicationFramework;

- уровень среды исполнения (Android Runtime) входит набор библиотек ядра, которые реализуют большую часть функциональности библиотек ядра языка Java;

- уровень ядра Linux (LinuxKernel) основан на ОС Linux версии 2.6, при этом программам доступны системные службы ядра, такие как управление памятью и процессами, обеспечение безопасности, работа с сетью и драйверами, то есть ядро служит слоем абстракции между аппаратным и программным обеспечением.

Таким образом, в результате данной работы, были проанализированы средства разработки программного обеспечения операционной системы Android. Проведено сравнение между такими средствами разработки как Android SDK и Android NDK. Выявлены их положительные и отрицательные стороны.

Литература

1. Голощанов, А. Google Android. Программирование для мобильных устройств. – СПб.: БХВ-Петербург, 2010. – 448 с.
2. Роджерс, Р. Android. Разработка приложений / Р. Роджерс, Д. Ломбардо. – М.: ЭКОМ Паблшерз, 2010. – 400 с.
3. Сайт разработчиков Android: [Электронный ресурс]. Режим доступа: <http://developer.android.com/about/index.html>

К.Н. Бурик, С.О. Потопнин
Научный руководитель: старший преподаватель Д.В. Бейлекчи
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: kaf-eivt@yandex.ru

Исследование средств разработки мультимедийных приложений с использованием XNA Framework

Для разработки мультимедийных приложений, требуется написание большого объема кода, связанного с программированием графики и воспроизведением аудио- или видеоданных. Поэтому существуют готовые библиотеки, реализующие основные функции мультимедийных приложений. Но к этим библиотекам нельзя получить доступ с помощью стандартных средств при создании мультимедийных программ для платформы Microsoft .NET.

В 2005-м году Microsoft выпустила игровую консоль – XBOX 360. Архитектура игровой консоли сильно отличалась от архитектуры персональных компьютеров, работающих под операционной системой Windows. В результате у Microsoft появились две несовместимых платформы – Xbox 360 и Windows. Это осложнило жизнь разработчикам, так как создание приложения для обеих платформ сводится к написанию двух отдельных приложений для каждой из них, что могут себе позволить только достаточно крупные компании. Такое разделение платформ не устраивало Microsoft и это привело к необходимости создания инструментария позволяющего одиночным разработчикам и малым компаниям создавать кроссплатформенные приложения, работающие на обеих платформах. За основу была взята платформа .NET, для которой не имеет значения, какой процессор установлен в системе.

Но при использовании .NET Framework возникают проблемы:

1. .NET Framework содержит мощную библиотеку классов, функциональность, которой является избыточной для игровой консоли. Также, ряд классов .NET Framework привязаны к операционной системе Windows.

2. Managed DirectX, является настройкой над DirectX. Так как DirectX – это один из компонентов платформы Windows, Managed DirectX автоматически оказывается привязанным к платформе Windows.

Для решения первой проблемы на XBOX были реализованы подмножества классов .NET Framework. Вторая проблема была решена разработкой XNA Framework. Так же вторая задача XNA Framework – снизить объем работы для портирования приложений на другие платформы Microsoft. В настоящее время XNA Framework также поддерживает разработку приложений для мобильных устройств с операционной системой Windows Phone. [1]

Таким образом, XNA Framework – это набор библиотек .NET Framework, позволяющий разработчикам создавать мультимедийные приложения на любых поддерживаемых платформах с использованием языка C#.

Структура XNA Framework приведена на рисунке 1

Все компоненты XNA Framework подразделяются на 4 уровня [2]:

1. Platform (Платформа) – наиболее низкий уровень, содержащий платформо-зависимые API. В основном приложения не знают о существовании этого уровня, используя компоненты более высоких уровней. Также обращение к этому уровню снижает диапазон поддерживаемых платформ.

2. Core Framework - нижний платформо-независимый уровень XNA, обеспечивающий базовую функциональность. Содержит компоненты для работы с графикой, для проведения математических расчетов, для работы устройств ввода – вывода, для работы со звуком и видео и для работы с файловой системой.

3. Extended Framework - набор высокоуровневых классов, решающих основные задачи, решаемые разработчиками мультимедийных приложений: инициализация графического устройства, организация цикла обработки сообщений, экспорт моделей и текстур из графических редакторов.

4. Код мультимедийного приложения, который реализует прикладную функциональность программного обеспечения.

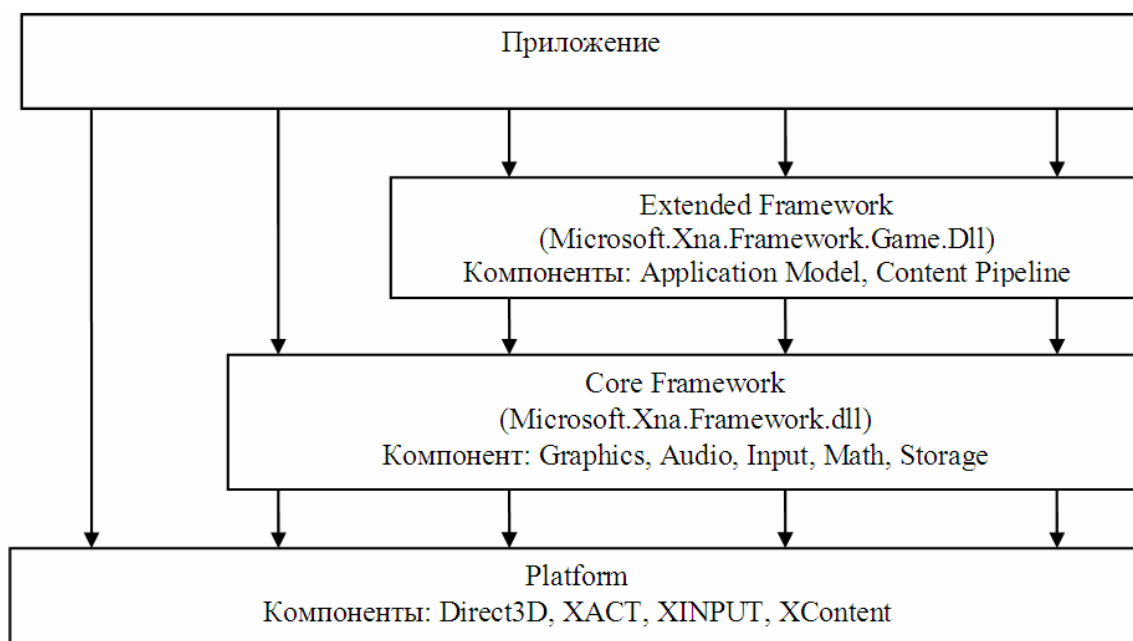


Рис.1. Структурная схема программного обеспечения

Для разработки приложений с использованием XNA Framework можно использовать среду программирования Microsoft Visual Studio (начиная с версии 2005) или её специальную модификацию XNA Game Studio, которая содержит дополнительные инструменты для разработки мультимедиа и игровых приложений, такие как редактор архивов аудио файлов, редактор текстур.

Таким образом, в результате данной работы была проанализирована средство программирования XNA Framework и исследовано его использование для создания мультимедийных приложений.

Литература

1. XNA Developer Center: [Электронный ресурс]. Режим доступа: <http://msdn.microsoft.com/en-us/aa937791.aspx>.
2. Часто задаваемые вопросы по XNA Framework и XNA Game Studio : [Электронный ресурс]. Режим доступа: <http://www.xnadev.ru/faq.php>.

А.А. Габдрахманова
Научный руководитель: канд. техн. наук, доцент А.А. Быков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: sapres@mivlgu.ru

Исследование современных программных средств проектирования баз данных

Цель данной работы - оценить сегодняшние проблемы и тенденции развития технологий проектирования БД.

Многие современные СУБД – средства, позволяющие создавать новую схему СУБД или просмотреть уже готовую. Но в последнее время получили развитие CASE-средства (Computer-Aided System Engineering). Существует большое количество различных CASE-средств, но для создания баз данных чаще всего предпочитают те программные продукты, что имеют средство создания структурных диаграмм. Современные средства проектирования данных поддерживают несколько типов СУБД (например, ERwin фирмы Computer Associates поддерживает более 20 различных СУБД) [1]. Уровень поддержки той или иной платформы в разных средствах проектирования данных может быть разным. Поэтому, выбирая средство проектирования следует учитывать возможности данной программы с точки зрения поддержки особенностей той или иной платформы. Также стоит учитывать факт о том, что средство поддерживающие больше возможностей и платформ, стоит дороже.

В настоящее время существует множество CASE-средств, наиболее популярными из которых являются [2]:

1. ERwin (Computer Associates)

ERwin разработан компанией Logic Works, которая в 1988 году была приобретена фирмой Platinum Technologies, а ее, в свою очередь, приобрела компания Computer Associates. Данная программа в течение последних десяти лет занимает лидирующие позиции на рынке, а так же среди многих других программ [3, 5]. ERwin представляет собой специализированное средство проектирования данных. Предполагается, что моделирование бизнес-процессов и потоков данных производится с помощью других продуктов, с которыми можно осуществлять обмен сведениями о моделях. ERwin не ориентирован на какую-то конкретную БД и поддерживает более 20 типов СУБД, за исключением новых типов СУБД т.к. последняя версия ERwin была выпущена в 2009 году как минимум более года для данной программ не было выпущено. Поэтому при использовании данного продукта с последними версиями СУБД могут возникнуть проблемы. Но это не помеха программы ERwin остаться одной из самых популярных в мире программных продуктов данного класса благодаря поддержке огромного количества платформ, удобного интерфейса и главное поддержке специфических особенностей организации физической памяти наиболее популярных серверных СУБД.

2. PowerDesigner (Sybase)

PowerDesigner (бывший S-Designor, принадлежавший компании PowerSoft) представляет собой программу, в составе которой входят средства создания логических моделей, и такие средства как создания физических моделей и объектно-ориентированного моделирования, используемое при генерации клиентских приложений [3, 5]. Данная программа делится на отдельные продукты, которые в свою очередь отвечают за определенное средство. Такой продукт как PowerDesigner PhysicalArchitect отвечает за создания физических моделей. А в состав продукта PowerDesigner DataArchitect входит средства создания концептуальных и физических моделей, в состав PowerDesigner Developer - средства объектно-ориентированного моделирования и создания физических моделей. Но на мой взгляд самый оптимальный продукт это PowerDesigner ObjectArchitect где в состав входят сразу все три средства. Физические и концептуальные модели в PowerDesigner DataArchitect хранятся в разных файлах, однако возможна генерация как физической модели на основе модели концептуальной, так и наоборот. Помимо серверных СУБД производства Sybase (Adaptive Server Enterprise 12.0, Sybase SQL Anywhere) PowerDesigner DataArchitect способен работать с любыми ODBC-источниками.. Поддерживается также генерация кода Web-приложений и объектов для Sybase Enterprise

Application Server где в основе находится физическая модель. PowerDesigner DataArchitect может импортировать как логические так и физические модели ERwin. PowerDesigner DataArchitect может хранить свои модели данных в коллективно разделяемом репозитории, управляемом с помощью средства PowerDesigner MetaWorks и доступном как дополнительный модуль в составе любого из перечисленных выше продуктов.

3. ER/Studio (Embarcadero Technologies)

ER/Studio менее известен в нашей стране, чем ERwin и PowerDesigner DataArchitect.

По своему назначению этот продукт сходен с выше описанным продуктом ERwin. ER/Studio представляет собой специализированное средство проектирования данных но не содержащий инструментов для объектно-ориентированного моделирования или моделирования бизнес-процессов. Список поддерживаемых БД у данного продукта достаточно широк и включающие наиболее популярные серверные и настольные БД [3, 5]. Но в отличие от продукта ERwin последняя версия этого продукта корректно поддерживает новые типы данных .ER/Studio поддерживает написание макросов на SAX Basic. Но как и в любой программе в ER/Studio тоже есть недочеты , данный продукт не позволяет добавить к каждой таблице свои шаблоны триггеров или просмотреть код конкретного триггера в процессе разработки модели - чтобы получить код одного триггера, нужно сгенерировать скрипт для всей модели. ER/Studio поддерживает сохранение в двух форматах DDL-скрипта, XML. Также создать репозиторий для хранения в любой серверной СУБД. ER/Studio может импортировать модели ERwin, но при импорте теряются связи шаблонов серверного кода с конкретными таблицами, и не все макросы ERwin корректно преобразуются в макросы SAX Basic. [4, 5]

Мною были рассмотрены самые популярные программы проектирования СУБД. Можно утверждать, что на данный период времени существует множество программ поддерживающих многие функции по построению СУБД. В условиях современного развития средств проектирования актуальной становится деятельность, использующая многие методы классического проектирования, но требующая иной организации и многих дополнительных методов, а также новых, которые заменили бы некоторые из тех, что были разработаны 10 и более лет назад. Таким образом, даже при создании сравнительно небольших баз данных необходимо использовать современные технологии проектирования. При этом становится актуальной задача создания свободно распространяемых CASE-средств.

Литература

1. Проектирование СУБД [Электронный ресурс] Материал из Википедии — свободной энциклопедии. Режим доступа: [http://ru.wikipedia.org/wiki/Система_автоматизированного_проектирования; свободный.]
2. Латышев П.Н. Каталог САПР. Программы и производители: Каталогное издание. — М.: ИД СОЛОН-ПРЕСС, 2006, 2008, 2011. — 608, 702, 736 с.;
- 3.Пакет ERwin[Электронный ресурс] Режим доступа: [<http://www.hardline.ru/2/22/676/1047-1.html>.]
4. [Электронный ресурс] режим доступа[[http://www.compress.ru/article.aspx?id=9431 &iid=402](http://www.compress.ru/article.aspx?id=9431&iid=402)]
- 5.[Электронный ресурс] режим доступа [[http://www.studarhiv.ru/dir/cat32/subj45 /file1435/view1435/page2.html](http://www.studarhiv.ru/dir/cat32/subj45/file1435/view1435/page2.html)]

А.Н. Журавский
Научный руководитель: старший преподаватель Д.Г. Привезенцев
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: sapres@mivlgu.ru

Принципы проектирования электронных словарей

В настоящее время в век стремительного развития компьютерных технологий в лексикографии появилась новая отрасль — электронная лексикография, занимающаяся созданием электронных словарей. Электронные словари сочетают большой объем с удобством пользования, поэтому пользуются спросом. Существующая система гиперссылок обеспечивает возможность быстрого доступа к нужной качественной информации. В этой связи для создания терминологических электронных словарей важна консолидация сил и умений не только программистов и специалистов по языку — лингвистов, но и специалистов в определенной области знаний, т. е. тех, кто будет пользоваться им в своей профессиональной деятельности.

Электронный словарь — словарь в компьютере или другом электронном устройстве. Позволяет быстро найти нужное слово, часто с учётом морфологии и возможностью поиска словосочетаний (примеров употребления), а также с возможностью изменения направления перевода (например, англо-русский или русско-английский). Внутренне устроен как база данных со словарными статьями. Электронные словари не стоит путать с компьютерными словарями, предназначенными не для пользователей, а для компьютерных программ, работающих с текстами на естественных языках. Учебный электронный словарь состоит из следующих 8 взаимосвязанных блоков

1) Справочный материал (включая общую информацию о словаре, правила пользования, условные обозначения и т.п.).

- 2) Корпус словаря.
- 3) Грамматическое приложение.
- 4) Страноведческое приложение.
- 5) Мультимедиа-приложение.
- 6) Разговорник.
- 7) Тематические иллюстрированные страницы.
- 8) Интерактивные упражнения.

При создании любого электронного словаря необходимо учитывать данные пользователя, его портрет:

1. Если это переводчик, то надо учитывать уровень знания языка пользователем
2. Если это толковый словарь, то надо учитывать его конкретное направление:
 - а) Экономический толковый словарь
 - б) Толковый словарь Русского языка и т.д.

Создание словаря можно разбить на три этапа:

Первый этап

- 1) Критический анализ современного состояния лексикографии:
 - а. анализ существующих словарей;
 - б. исследование потребностей адресата – конечного пользователя словаря.
- 2) Определение требований к словарям на основе полученных данных.
- 3) Разработка макроструктуры (и медиа структуры) словаря.
- 4) Разработка микроструктуры словаря.

На этом этапе мы конкретизируем вид и состав словаря.

Второй этап

- 1) Составление словника.
- 2) Распределение слов по семантическим полям (для словарей с элементами тезауруса).
- 3) Подбор переводных эквивалентов для слов словника.

4) Исследование словарных единиц (переводных эквивалентов) с точки зрения их денотативно-сигнификативных, экспрессивно-прагматических и синтагматических характеристик и занесение результатов исследования в лексикографическую базу данных.

5) Оформление словарных статей и словаря в целом в соответствии с разработанной ранее структурой.

На этом этапе мы создаем саму базу необходимой информации(слов)

Третий этап

1) Экспериментальная проверка соответствия полученного продукта поставленным целям.

В разработанной нами модели. На этом этапе мы проверяем работу словаря и исправляем недочеты.

Таким образом, нами разрабатывается модель электронного словаря. Предлагаемая концепция основана на теоретических достижениях современной лексикографии и смежных с ней лингвистических дисциплин и соответствует поставленным требованиям.

А. М. Замотаева
Научный руководитель: канд. техн. наук, доцент К.В. Макаров
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: nastenka.a.a.a@yandex.ru

Автоматизация управления учебным процессом на базе web-технологий

Автоматизация любого реального процесса – сложная задача. Учебный процесс – не исключение. Большое число участников, разнообразие взаимосвязей между ними, многообразие документов, сопровождающих осуществление учебного процесса затрудняет задачу автоматизации.

Полноценная автоматизация возможна лишь с привлечением целого комплекса программных и аппаратных средств. Частично снизить затраты на реализацию проекта автоматизации позволяет сделать рациональный выбор архитектуры автоматизированного комплекса.

Одним из подходов, который может снизить затраты является использование web-технологий для формирования программной составляющей автоматизированного комплекса. Применение в качестве основы информационного комплекса web-технологий может быть обосновано простотой разворачивания, отсутствием необходимости в конфигурации отдельных рабочих мест и специального программного обеспечения. На пользовательской стороне потребуется наличие лишь интернет-браузера и офисного пакета.

Все вычислительные работы будут выполняться сервером. Пользователю будут передаваться лишь результаты в виде динамически сформированных HTML-страниц и файлов с готовыми документами в формате MS Excel.

На начальном этапе использования комплекса возможно привлечение существующей сетевой инфраструктуры вуза без ее модификации.

К составу программного обеспечения сервера предъявляются следующие требования: наличие СУБД MySQL, PHP и расширения PHPExcel.

Обеспечение единого хранилища ресурсов, обеспечивающих организацию и осуществление учебного процесса вуза с возможностью доступа с удаленных рабочих мест.

Структура хранилища данных в рамках проектируемого комплекса представлена на рисунке 1.

Для обеспечения упорядоченного и санкционированного доступа к информационным ресурсам комплекса будет предусмотрено разграничение прав доступа для различных категорий пользователей. Будут выделены группы пользователей, которые в соответствии со своими должностями будут иметь доступ к соответствующим их должностным обязанностям возможностям комплекса.

Работа с комплексом будет производиться по следующему алгоритму: заведующие кафедрами будут отвечать за наполнение хранилища рабочими учебными планами. На основе рабочих учебных планов заместители деканов по учебной работе будут формировать комплекс календарных планов [1], которые в свою очередь будут использованы учебным отделом для составления расписания и планирования нагрузки преподавателей. Преподаватели на основе данных о нагрузке и рабочих учебных планов будут формировать рабочие программы дисциплин и индивидуальные планы преподавателей.

Сотрудники учебного отдела могут осуществлять контроль за правильность формирования всех документов, которые будут собираться в централизованном хранилище. Наличие централизованного хранилища позволит исключить наличие дубликатов документов и обеспечит доступ к актуальным версиям различных документов.

Генерация документов будет проводиться в автоматизированном режиме. При таком подходе от пользователей комплекса будет требоваться лишь задание необходимых параметров для получения документа. Операции по сохранению и извлечению документов, а также вспомогательных данных из хранилища будет осуществляться автоматически средствами серверной части программного комплекса.



Рис. 1. Структура хранилища и ключевые пользователи комплекса

Главное преимущество комплекса автоматизации учебного процесса, построенного на базе web-технологии, – это наличие единого хранилища актуализированной информации, рабочих учебных планов и прочих документов; средств генерации календарных планов, индивидуальных планов преподавателя, рабочих программ дисциплин.

Литература

1. А.М. Замотаева Автоматизация учебного процесса на основе электронного учебного плана / Научный потенциал молодежи – будущее России. IV Всероссийские научные Зворыкинские чтения: сб. тез. докл. IV Всероссийской межвузовской научной конференции. Муром, 20 апреля 2012 г.– [Электронный ресурс].– Муром: Изд.-полиграфический центр МИ ВлГУ, 2012.– 1102 с.: ил.– 1 электрон. опт. диск (CD-ROM)

О. С. Конихина, С. С. Костин
Научный руководитель: канд. техн. наук, доцент А.А. Быков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: konikhina_olesya@mail.ru

Анализ методик получения неправомерного доступа к программному обеспечению

В настоящее время проблема безопасности компьютерных сетей является актуальной. В странах, где высок уровень компьютеризации, проблема борьбы с компьютерной преступностью уже довольно давно стала одной из первостепенных. Основным методом защиты против злонамеренных атак в компьютерной сети является система парольной защиты, которая имеется во всех современных программных продуктах. Можно отметить, что компьютерная информация является предметом или средством совершения преступления. Подавляющее большинство компьютерных преступлений совершается с использованием возможностей, которые предоставляет своим пользователям сеть Интернет.

В защитной броне любой системы существует немало прорех. Системные администраторы могут бесконечно усиливать защиту, но способ обойти ее всегда найдется.

Рассмотрим основные методы взлома компьютерных сетей; программное обеспечение любой универсальной компьютерной системы состоит из трех основных компонентов: операционной системы, сетевого программного обеспечения и системы управления базами данных

Попытки взлома защиты компьютерных систем можно разделить на три группы [1]:

1. атаки на уровне операционной системы;
2. атаки на уровне сетевого программного обеспечения (СПО);
3. атаки на уровне систем управления базами данных (СУБД).

Наиболее встречаемые способы взлома программного обеспечения:

1. ввод серийного номера — данный метод заключается во взломе программы посредством введения правильного регистрационного ключа (или фразы), полученного нелегальным способом

2. использование загрузчика — это способ обходить некоторые виды защиты программного обеспечения, заключающиеся в использовании внешних (навесных) систем защиты.

3. применение (бинарного) патча — способ, при котором изменение производится динамически в файлах программы. Считается, что это один из самых наиболее простых и быстрых способов взлома программного обеспечения.

4. использование взломанной версии файла(ов) — способ заключается в подмене оригинальных файлов программы файлами, которые уже взломаны или содержат вирусную программу.

5. использование эмулятора ключа — предоставленный способ используется для обмана защит, построенных на использовании в качестве защиты электронного ключа (обычно подключаемого к USB порту компьютера). [2]

Атаки на уровне систем управления базами данных

Введем понятие систем управления базами данных – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных. Защита СУБД является одной из самых легких задач. Это связано с тем, что СУБД имеют строго определенную внутреннюю структуру, и операции над элементами СУБД заданы довольно отчетливо. Есть четыре основных действия — поиск, вставка, удаление и замена элемента. Другие операции являются служебными и применяются достаточно редко или не используются вовсе.

Атаки на уровне операционной системы

Защищать операционную систему гораздо сложнее. Внутренняя структура современных операционных систем крайне сложна, и поэтому соблюдение адекватной политики безопасности является значительно более нелегкой задачей. Элементарные методы взлома

оказываются ничуть не хуже самых изощренных, так как при простом алгоритме атаки, вероятность ее завершения без ошибок и сбоев больше, особенно если возможности предварительного тестирования этого алгоритма в условиях, приближенных к "боевым", весьма ограничены.

Атаки на уровне сетевого программного обеспечения

Сетевое программное обеспечение предназначено для организации совместной работы группы пользователей на разных компьютерах. Оно позволяет организовать общую файловую структуру, общие базы данных, доступные каждому члену группы, поэтому СПО является наиболее уязвимым. Канал связи, по которому передаются сообщения, чаще всего не защищен, и всякий, кто может обладать доступом к данному каналу может перехватывать сообщения и отправлять свои собственные. [1]

Поэтому на уровне СПО возможны следующие хакерские атаки: прослушивание сегмента локальной сети; перехват сообщений на маршрутизаторе; создание ложного маршрутизатора и т.д.

Способами (методами) защиты от атак на уровне СПО являются: максимальное ограничение размеров компьютерной сети; изоляция сети от внешнего мира; шифрование сетевых сообщений; электронная цифровая подпись сетевых сообщений; использование брандмауэров.

Стоит отметить, что существуют атаки, которым может быть подвергнута почти любая операционная система, такие как: кража пароля, наблюдение за пользователем при вводе пароля, перебор всех возможных вариантов пароля; восстановление ранее удаленных объектов, превышение полномочий, запуск программы от имени пользователя, имеющего необходимые полномочия, или в качестве системной программы.

Безопасность программного обеспечения (ПО) в широком смысле является свойством данного ПО функционировать без проявления различных негативных последствий для конкретной компьютерной системы. В настоящее время одним из наиболее опасных средств информационного воздействия на компьютерные системы являются программы - вирусы или компьютерные вирусы. [2, 3]

Таким образом, для повышения надежности эксплуатации программного обеспечения необходимо соблюдать следующие требования:

1. использование лицензионных антивирусов;
2. применение сложных паролей;
3. своевременное обновление ПО;
4. исключение возможности передачи информации о своих паролях посторонним.

Литература

1. Акулов О.А., Медведев Н.В. Информатика базовый курс. - М.: Омега-Л, 2005.
2. Wikipedia.ru. [Электронный ресурс]: Свободная энциклопедия. – Электро. дан. – М., Режим доступа: <http://ru.wikipedia.org>
3. Безопасность программного обеспечения компьютерных систем. Казарин О.В. www.cryptography.ru Издано: Москва, МГУЛ, 2003, 212 с.

С.С. Костин
Научный руководитель: канд. техн. наук, доцент А.А. Быков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: sapres@mivlgu.ru

Исследование архитектуры компьютерных вирусов на примере Trojan Winlock

Винлок (с англ. Winlock) — троян-вымогатель, начал своё развитие в начале 2007 года. Самостоятельно очень сложно выводится — большинство Winlock'ов имеют достаточно хорошую защиту от удаления.

Внешне выглядит как окно, в котором вас просят отправить СМС на определённый номер в течение дня, иначе ваш компьютер будет заблокирован. После того, как вы отправите смс, должен прийти код для разблокировки. Код, конечно, не приходит, и деньги не возвращаются.

После скачивания Winlock запускается и прописывает себя в автозагрузку Windows. Соответственно, после перезагрузки (а в последних версиях — после скачивания), пользователь увидит перед собой окошко, с просьбой отправить СМС. Выключить компьютер почти невозможно — он блокирует Ctrl+alt+Del, Alt+F4, Winkey+r, а также завершает explorer.exe путём подмены пути к нему в реестре на файл вируса. Последнее поколение Winlock 'ов появляются даже при запуске безопасного режима, поэтому удаление их становится ещё сложнее. Также вирус может представлять собой DLL-библиотеку, генерирующую случайный exe-файл с этим Winlock'ом.

Некоторые модификации Winlock'a прописывают значения, используя символы (буквы) из разных алфавитов. Пример: C:\WINDOWS\system32\userinit.exe, значение \userinit.exe написано с использованием кириллической Е. Самое оптимальное решение — перезаписать самим значения важных параметров. Однако новая версия антивируса Данилова (возможно, и другие антивирусы) полностью блокирует данные реестра от редактирования (как пользователем, так и вирусами). После перезагрузки и входа в систему трояна можно удалить вручную по уже известному пути. Этот способ хоть и эффективен, но подходит только для опытных пользователей. [1]

Последние модификации Winlock'ов действуют иначе. Можно скачать вирус, не меняющий пути в реестре, а меняющий непосредственно файлы explorer.exe и/или userinit.exe. Последние версии (MBR-Locker'ы) устанавливаются как отдельный загрузчик и запускаются ещё до загрузки windows. Очень редко встречались вирусы, которые переписывали BIOS и стартовали сразу, вместе с ним, но работало это только на некоторых моделях материнских плат [2].

Существует множество способов борьбы с данным вирусом. Рассмотрим некоторые из них [3]:

1. Самый простой способ — переустановить windows и форматировать жесткие диски. Не обращайте внимания на угрозы, BIOS они не удалят и файлы тоже.

2. Попробуем запустить компьютер в безопасном режиме (если не сработало, то переходим к другому способу). Находим в реестре (run → regedit.exe) папки автозапуска и удаляем вирус оттуда.

3. Если вирус блокирует обычный безопасный режим, то при нажатии клавиши F8 необходимо выбрать безопасный режим с поддержкой командной строки. На данный момент времени большинство вирусов не способны его заблокировать. После загрузки надо запустить редактор реестра при помощи команды regedit и искать там подозрительные записи. В первую очередь необходимо проверить ветку HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon, в частности в параметре Shell должно быть написано explorer.exe, а в параметре Userinit — C:\WINDOWS\system32\userinit.exe, (обязательно с запятой). Если там всё в порядке, необходимо проверить этот же путь, но в ветке HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run. В случае обнаружения подозрительных записей, их

необходимо заменить на стандартные значения (в случае с автозагрузкой удалить). Также необходимо очистить папки C:\Windows\Temp, C:\Documents and Settings\%Username%\Local Settings\Temporary Internet Files, cash вашего браузера. [1]

4. На официальном сайте Касперского есть возможность скачать бесплатное ПО, которое требуется записать, например, на пустой диск. При помощи этого ПО, проверяем компьютер и удаляем все вирусы, которые будут найдены [4]

5. Попробуем нажать Ctrl+Alt+Del. Часть вирусов не блокирует это сочетание клавиш. Если удалось, пробуем сменить пользователя — при перезапуске компьютера вирус удаляется. Другой вариант: в диспетчере задач завершить процесс Explorer.exe. Не обращая внимания на предупреждение системы, о том, что при завершении процесса могут возникнуть неполадки, отвечаем положительно, после чего, вручную запускаем этот же процесс. Баннер пропал. Затем необходимо выполнить откат к более раннему состоянию системы, а также проверить компьютер на наличие вируса.

Таким образом, видно, что вирус получает доступ к системным файлам при его активации, в отличие от вирусов-червей. То есть, прежде всего, необходимо следить за запускаемыми программами, чтобы предотвратить заражение компьютера. Так же необходимо иметь заранее заготовленный набор программных средств для лечения компьютера от вируса. В случае крайней необходимости может потребоваться восстановление файлов пользователя из заранее подготовленной резервной копии.

Литература

1. Винлок – Lurkmore [Электронный ресурс] / Режим доступа: <http://lurkmore.to/%C2%E8%ED%EB%EE%EA> свободный. Заглавие с экрана.
2. Sucurelist – всё об интернет-безопасности [Электронный ресурс] / Режим доступа: <http://www.viruslist.com> свободный. Заглавие с экрана.
3. Электронная энциклопедия [Электронный ресурс] / Режим доступа: <http://ru.wikipedia.org/wiki/Trojan.Winlock> свободный. Заглавие с экрана.
4. Лаборатория Касперского [Электронный ресурс] / Режим доступа: <http://www.kaspersky.ru> свободный. Заглавие с экрана.

А.А. Миналиев
Научный руководитель: старший преподаватель Я.Ю. Кульков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: corvojensen@gmail.com

Сравнительный анализ «облачных» сервисов хранения данных

Облачное хранилище данных — модель онлайн-хранилища, в котором данные хранятся на многочисленных, распределённых в сети серверах, предоставляемых в пользование клиентам, в основном третьей стороной. В противовес модели хранения данных на собственных, выделенных серверах, приобретаемых или арендуемых специально для подобных целей, количество или какая-либо внутренняя структура серверов клиенту, в общем случае, не видна. Данные хранятся, а равно и обрабатываются, в так называемом *облаке*, которое представляет собой, с точки зрения клиента, один большой, виртуальный сервер.

С наступлением эпохи информационного общества. У человека возникло огромное количество данных которые требуется обрабатывать, хранить, передавать между другими людьми. Большинство пользователей хранят информацию на жестком диске, а если нужно взять ее с собой, то копируют на съёмные носители. Между тем появление облачных хранилищ дало человеку возможность гораздо проще и выгоднее пользоваться своими данными. Теперь в любой момент к ним есть доступ при наличии Интернета. С помощью облака удобно обмениваться информацией с другими людьми. Достаточно отправить ссылку адресату, и тот загрузит файлы в любой подходящий момент. Так же при помощи облака можно спасти важную информацию (например, при выходе из строя жесткого диска), достаточно скопировать ее в облако и периодически обновлять.

Идея принадлежит J.C.R. Licklider, в 1970 году. Он работал ARPANET (Advanced Research Projects Agency Network). Идея заключалась в том, что каждый человек будет подключен к сети, из которой он будет получать не только данные, но и программы. John McCarthy высказал идею о том, что вычислительные мощности будут предоставляться пользователям как услуга (сервис). На этом развитие облачных технологий было приостановлено до 90-х годов, после чего ее развитию поспособствовал ряд факторов.

1. Расширение пропускной способности Интернета дал толчок скорейшему развитию облачных вычислений.
2. Salesforce.com в 1999 году данная компания стала первой компанией предоставившей доступ к своему приложению через сайт.
3. Разработка облачного веб-сервиса компанией Amazon в 2002 году. Данный сервис позволял хранить, информацию и производить вычисления.
4. В 2006, Amazon запустила сервис Elastic Compute cloud (EC2), который позволял его пользователям запускать свои собственные приложения. Amazon EC2 и Amazon S3 стали первыми доступными сервисами облачных вычислений.
5. Создание компанией Google, платформы Google Apps для веб-приложений в бизнес секторе.
6. Развитие аппаратного обеспечения способствовало быстрому росту облачных технологий, доступности данной технологии для малого бизнеса и индивидуальных лиц. Значительную роль сыграло создание многоядерных процессоров и увеличение емкости накопителей информации.

Много компаний занятых в сфере информационных услуг. Представили свои сервисы. К примеру: 4shared, ADrive, Asus WebStorage, Amazon S3, Bitcasa, Box, Comodo Cloud, CrashPlan, Cubby, DiskaNet, Dropbox, HDD4ever, iWork, Jottacloud MediaFire, Mega, SkyDrive, SpiderOak, SugarSync, symform, Ubuntu One, Wuala, ZumoDrive, Диск Google и Яндекс Диск. Несмотря на новизну этих технологий уже сейчас можно выделить ряд лидеров. В интернете много обзоров на облачные хранилища данных, но среди них нет актуального на сто процентов. Это из-за того что данная технология как и любая другая информационная технология совершенствуется. Рассмотрим сервисы наиболее крупных компаний таких как:

Диск Google 24 апреля 2012 года сервис был представлен пользователям. В Диске Google предоставляется 5 гигабайт для бесплатного хранения данных. Если выделенного объёма недостаточно, можно приобрести дополнительно от 25 ГБ до 16 ТБ. На момент запуска сервиса клиентское программное обеспечение было доступно для следующих платформ: персональные компьютеры под управлением Windows, Mac OS, смартфоны и планшеты с ОС Android и iPhone и iPad.

Microsoft SkyDrive Первый выпуск в августе 2007 года. Сервис SkyDrive позволяет хранить до 7 ГБ информации (или 25 ГБ для пользователей, имеющих право на бесплатное обновление) в упорядоченном с помощью стандартных папок виде. Для изображений предусмотрен предпросмотр в виде эскизов, а также возможность их просмотра в виде слайдов. Выпущены приложения для Android, iOS, Windows Phone, Windows, Mac OS.

Dropbox Даёт 2ГБ. Бесплатно. Работает на Windows, Mac OS, Android, iOS, BlackBerry OS. 16 декабря 2010 года вышла стабильная версия сервиса. Изначально идея родилась у основателя компании Дрю Хьюстона во время автобусной поездки из Бостона в Нью-Йорк. Чтобы не скучать в пути, Хьюстон (в то время — студент MIT) взял с собой ноутбук, но забыл захватить флэшку. В итоге ему нечем было заняться с пустым ноутбуком, раздосадованный, он начал прямо в автобусе писать приложение для синхронизации доступа к файлам через интернет.

SugarSync Поддерживает: Windows, Mac OS. А так же мобильные платформы: Android, iOS, Symbian OS, BlackBerry OS и Windows Mobile. Сервис был впервые представлен в 2008 году, когда компания Sharpcast анонсировала программу, развивающую функционал их старой разработки Sharpcast Photos, которая предоставляла возможность синхронизировать фотографии пользователя.

Яндекс. Диск В мае 2012 года регистрация новых пользователей была доступна только по приглашениям. В настоящее время регистрация пользователей доступна всем. Основные платформы

- Поддерживает Windows, MacOS.
- Мобильная версия для iOS и Android. Symbian

Сервис полностью бесплатен. По умолчанию предоставляются 3 ГБ пространства, которые можно расширить до 10 ГБ.

Рассмотреть всех и выделить лидирующий сервис сложно. Из-за того что технически они примерно на одном уровне. Остаётся рассматривать по удобству для пользователя. Одним из минусов у большинства «облачных» хранилищ, на мой взгляд, является отсутствие русского языка, что не даёт многим пользователям и России полноценно пользоваться сервисом. Даже зная язык приятнее работать с интерфейсом на родном языке. По этому параметру можно выделить: Диск Google, SkyDrive, Яндекс Диск. Очень приятно и удобно будет работать с SkyDrive пользователям Windows 8 благодаря его сильной интеграции в ОС можно работать в проводнике Windows. Имеется поиск и настройки публичности. Пользователям смартфонов Android достался урезанный клиент SkyDrive. У многих возникают вопросы по поводу безопасности данных в «облаке» Dropbox и жалобы по поводу низкой скорости загрузки. На Google позволяют хранить любые файлы размером не более 1 Гб. Есть возможность редактировать документы совместно с другими пользователями и делиться ими с теми кто не имеет учётной записи Google. Плюсом можно назвать удобство сервиса на мобильных устройствах. Произведение музыки напрямую из папки Drive без загрузки на ПК.

Неудобна у Drive синхронизация. Синхронизируется одна папка, и если из нее удалить файл, то он удалится в корзину и на Drive. Яндекс диск имеет возможность подключаться как сетевой диск.

Если говорить в общем об «облаках» стоит отметить что сохранность данных всё же остаётся под вопросом. Так как нет стопроцентной защиты от хакерских атак, кражи данных. Поэтому важную информацию лучше оставлять на жёстких дисках компьютера. А выбор осуществлять по принципу где зарегистрирован там и использую. Потому как я уже говорил среди всего множества нет идеала, у каждого свои преимущества, у кого интеграция в ОС, кто предоставляет огромное количество места, кто синхронизирует с другими своими сервисами.

Моряков А.А.
Научный руководитель канд. техн. наук, доцент Н.П.Мольков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: sapres@mivlgu.ru

Система клиент-сервер для получения информации о движении городских автобусов

В настоящее время люди очень часто пользуются общественным транспортом, который приходится подолгу ждать на остановках, ожидание может продлиться до часу и даже больше. Столь продолжительное ожидание может служить массой проблем, начиная от неэкономичного использования времени, заканчивая безопасностью и здоровьем человека. Решить эту проблему можно взглянув на расписание общественного транспорта, выбрав несколько вариантов маршрута и рассчитав время прибытия на нужную Вам остановку. Но это заставляет тратить множество усилий и времени.

Решение данной проблемы может служить автоматизированный сервис определения времени прибытия общественного транспортного средства на нужную остановку, созданный на основе современных цифровых технологий. Сейчас практически у каждого человека есть мобильный телефон либо другое мобильное устройство, которое имеет доступ в интернет и поддержку установки приложений. Следуя из этого возможно создать клиент-сервер систему программных средств, с помощью которой любой человек сможет узнать нужную ему информацию. Решение нужно реализовать таким образом, чтобы оно было доступно практически всем на всех устройствах, это можно сделать, используя протоколы: OSCAR (служба ICQ), XMPP или аналогичные как канал передачи данных.

Для реализации выбран протокол XMPP. XMPP выбран потому, что у него отсутствуют: ограничение в частоте и количестве переданных сообщений, ограничение переподключений, отсутствие шифрование информации, отсутствие конфиденциальности. Данный проект состоит из сервера и клиента. Сервер – это специально написанная программная архитектура, основанная на открытом протоколе для передачи XML-данных через Интернет XMPP. С целью обеспечения клиентов нужной им информацией программа будет принимать от них сообщения определенного, заранее оговоренного формата. После того как запрос от клиента принят сервер обработает его и рассчитает время прибытия на запрошенный маршрут общественного транспортного средства. Клиент – любой пользователь подключенный к серверу используя клиент XMPP на любом устройстве поддерживающим данный протокол. Чтобы получить нужную информацию клиенту нужно отправить информацию о маршруте, который он хочет преодолеть и сервер тут же подберет и пришлет о времени прохождении всех ближайших общественных транспортных средств по данному маршруту.

Исследование методик повышения личной эффективности в задачах создания программных продуктов

В настоящее время у человека каждый день появляется множество задач, некоторые можно выполнить за пару часов, а некоторые нуждаются в месяцах работы для их выполнения. Задачи, которые требуют длительный срок на своё выполнение, несомненно, можно разделить на подзадачи, откуда появляется еще больше объектов которые нужно реализовывать. Накапливаясь задачи не только сложно удержать в голове, но еще более сложно определить наиболее приоритетные, что ведет к реорганизации трудового процесса. Планирование своей деятельности весьма важно для человека, ведь в появившееся свободное время не придётся осмысливать своё следующее действие. Отсутствие плана на длительный срок ведет к непрогрессивному и некачественному выполнению поставленных задач.

Решение данной проблемы может стать составление списков своих дел, что освободит от запоминания текущих задач, и позволит сконцентрироваться конкретно на их выполнении. Для организации рабочего процесса можно применить существующие системы, такие как GTD и Agile Results Getting. GTD- Getting Things Done (рус. «доведение дел до завершения») описана Дэвидом Алленом в книге «методика повышения личной эффективности» позволяет более эффективно распланировать дела.[1] Agile Results - система личной эффективности, которая нацелена на достижение значимых результатов, методологию разработал и описал в книге “Getting Results the Agile Way” J.D. Meier, топ-менеджер в команде разработчиков Microsoft Enterprise Strategy.[2] Составить список дел можно на бумаге или используя специализированные программы, как с применением существующих систем повышения личной эффективности, так и без них. Использование программ это более удобный вариант, так как принцип организации уже продуман разработчиком и пользователю не придется изучать дополнительную литературу для эффективного составления плана.

В данной работе для реализации решения, предложена структура программной системы для планирования, контроля задач и проектов. Решение основано на модели планировании дел разработанной при совместном использовании систем повышения личной эффективности GTD и Agile Results, так как обе системы имеют некоторые ограничения в достижении эффективности при раздельном использовании. Данная модель облегчает процесс организации рабочего времени, основываясь на распределении приоритетов, хранении, отслеживании и использовании подробной информации, связанной с выполняемыми задачами.

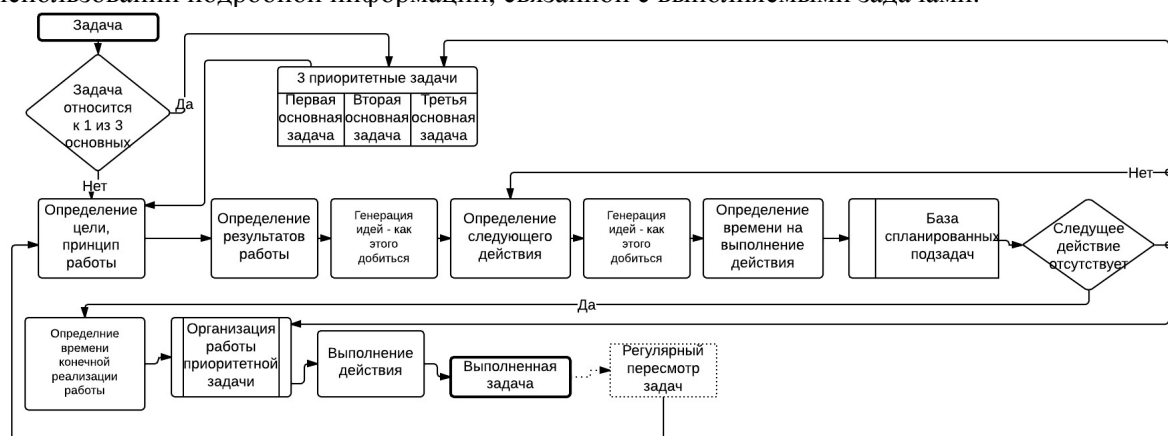


Рис. 1. Модель планирования созданная на основе методик GTD и Agile Results

Основная цель программной системы – создание программных решений, основанных на предложенной модели планирования с применением микроэргономики к интерфейсу, помогающие любому человеку использовать весь функционал программ.[3]

Для обеспечения успешной организации времени и задач, реализована программа, которая позволяет создавать список действий, выполнение которых приводит к организации дня и освобождению от некоторых ненужных действий при работе с ПК, что влечет к экономии времени.

Возможности программы это выполнение таких задач как: отключение, перезагрузка, сон, оповещение, проигрывание списка аудиокomпозиций. Задачи выполняются в зависимости от некоторых заданных пользователем условий: срабатывание таймера, будильника; определенное состояния заряда аккумулятора и загрузки процессора.

Также присутствует механизм «конструктор задач» основанный на разработанной модели, который помогает организовать задачу, учитывая множество подзадач необходимых для реализации главного проекта.

В течении дальнейшего развития проекта планируется создание программы-органайзера, которая позволит создавать списки задач, для избавления от необходимости тратить усилия на ручную организацию и запоминание дел которые нужно совершить.

Главные возможности программы: создание списка задач, которые могут быть физически выполнены пользователем; формирование проектов из отдельных подзадач, группировка задач по контексту, использование совокупности систем GTD и Agile Results.

Для сравнения были рассмотрены другие подобные решения: PowerOff, LeaderTask. PowerOff - органайзер для создания таймера и выполнения различных действий, будильника и списка дел. Отсутствует возможность проигрывать список аудиокomпозиций, очень перегруженный интерфейс, сложность в настройке, ограниченное число созданных задач.

LeaderTask - это большой проект созданный компанией "Органайзер ЛидерТаск". Данный продукт разработан следуя методикам GDT. Программа создана исключительно для создания проектов, задач, дел. Отсутствует функция автоматизации своих действий при работе с ПК. Чтобы начать пользоваться программой следует пройти обучение, программа платная, как основа при планировании задач взята только методика GTD.[4]

Предложенная в данной работе программная система основана на предложенной модели планирования, что позволит создать уникальный продукт.

При дальнейшем развитии данного проекта будет добавлено несколько полезных «механизмов»:

- Синхронизация всех созданных и сохраненных данных с сервером, чтобы в дальнейшем можно было легко восстановить все свои параметры в любой момент на любом компьютере.
- Возможность публикации в общественный доступ своих созданных задач, чтобы каждый пользователь мог увидеть данные списки и на их основе легче распланировать свой день. Данный механизм будет служить «опорой» для тех, кому трудно составить распорядок дня.
- Механизм сбора данных о поведении пользователя за компьютером, такие как: наиболее часто прослушанные аудиокomпозиции, наиболее часто используемые задачи, и т.д. на основе этих данных будут предлагаться готовые варианты заданий и автоматически настраиваться некоторые процессы работы ПК.

Литература

1. Getting Things Done — Википедия. [Электронный ресурс]. Режим доступа: http://ru.wikipedia.org/wiki/Getting_Things_Done, свободный.- Загл. с экрана.
2. Agile Results - новый подход к личной эффективности. Описание основных приемов и принципов. [Электронный ресурс]. Режим доступа: <http://betteri.ru/post/agile-results---novyy-podhod-k-lichnoy-effektivnosti-opisanie-osnovnyh-priemov-i-principov.html>, свободный.- Загл. с экрана.
3. Эргономическое обеспечение Web-дизайна - № 05, 2002 | Мир ПК | Издательство «Открытые системы». [Электронный ресурс]. Режим доступа: <http://www.osp.ru/pcworld/2002/05/163334>, свободный.- Загл. с экрана.
4. LeaderTask Справка - Getting Thing Done (GTD) Методология. [Электронный ресурс]. Режим доступа: <http://www.help.leadertask.ru/raznoe/50-gtd>, свободный.- Загл. с экрана.

И.С. Нагичев, Д.С. Дубинкин
Научный руководитель: канд. техн. наук, доцент А.А. Быков
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: 21strike12@mail.ru, denis6543@yandex.ru

Анализ методов тестирования программного обеспечения

Тестирование программного обеспечения - проверка соответствия между реальным и ожидаемым поведением программы. Тестирование проходит на конечном наборе тестов, выбранном определенным образом.

На сегодняшний день методы тестирования ПО не идеальны. Они не позволяют полностью выявить все ошибки анализируемой программы.

Существуют много методов тестирования ПО. Данный процесс является творческим и зависит от людей которые тестируют программу. Каждый процесс проверки может выявить ошибки программы только с точки зрения используемого метода.

Существует международный стандарт правил тестирования ПО ISO 9126. Этот стандарт правил содержит следующие атрибуты и критерии:

- Надёжность
- Сопровождаемость
- Практичность
- Эффективность
- Мобильность
- Функциональность

Благодаря этим критериям можно выявить максимальное количество ошибок алгоритма ПО.

Одними из наиболее популярных видов тестирования являются такие виды как:

По знанию системы (обозначает имеет ли разработчик тестов доступ к исходному коду тестируемого ПО):

1. Тестирование чёрного ящика (black box) - тестировщик имеет доступ к ПО только через те же интерфейсы, что и заказчик или пользователь, либо через внешние интерфейсы, позволяющие другому компьютеру либо другому процессу подключиться к системе для тестирования. [2]

2. Тестирование белого ящика (white box) - разработчик теста имеет доступ к исходному коду программ. Это типично для юнит-тестирования, при котором тестируются только отдельные части системы. [2]

3. Тестирование серого ящика (grey box) - разработчик теста имеет доступ к исходному коду, но при непосредственном выполнении тестов доступ к коду, как правило, не требуется. [2]

По времени проведения тестирования:

- Альфа-тестирование (alpha testing) — имитация реальной работы с системой штатными разработчиками, либо реальная работа с системой потенциальными пользователями/заказчиком. Чаще всего альфа-тестирование проводится на ранней стадии разработки продукта, но в некоторых случаях может применяться для законченного продукта в качестве внутреннего приёмочного тестирования. Иногда альфа-тестирование выполняется под отладчиком или с использованием окружения, которое помогает быстро выявлять найденные ошибки. Обнаруженные ошибки могут быть переданы тестировщикам для дополнительного исследования в окружении, подобном тому, в котором будет использоваться ПО. [1]

- Бета-тестирование (beta testing) — в некоторых случаях выполняется распространение предварительной версии (в случае проприетарного ПО иногда с ограничениями по функциональности или времени работы) для некоторой большей группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок. Иногда бета-тестирование выполняется для того, чтобы получить обратную связь о продукте от его будущих пользователей. [1]

Благодаря этим методам можно найти наибольшее количество ошибок в ПО. Но при этом эти методы не могут выявить все виды ошибок. Это связано с тем что в тестирование задействовано сравнительно не большое количество участников. Так же сильное влияние оказывает человеческий фактор. Все недочеты могут быть выявлены и исправлены только в процессе эксплуатации ПО.

Литература

1. Синицын С. В., Налютин Н. Ю. Верификация программного обеспечения. - М.: БИНОМ, 2008. - 368 с. - ISBN 978-5-94774-825-3
2. Бейзер Б. Тестирование чёрного ящика. Технологии функционального тестирования программного обеспечения и систем. - СПб.: Питер, 2004. - 320 с. - ISBN 5-94723-698-Режим доступа: <http://ru.wikipedia.org>

П. С. Павельев
Научный руководитель: канд. техн. наук, доцент Е. Е. Канунова
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: fevan@mail.ru

Разработка кросс - платформенных приложений на C++ с использованием Qt Creator

В рамках научной работы студента Муромского института на конференции будет рассмотрена реализация кросс - платформенных приложений на языке программирования C++ с использованием Qt Creator.

В настоящее время разработка приложений на одну платформу, например Windows, не является актуальной. Среди программистов и обычных пользователей всё большую популярность набирают такие ОС, как Linux, Mac OS и прочие.

На сегодняшний момент в Муромском институте студентов учат программировать для ОС семейства Windows на MS Visual Studio C++. Код, в результате, не является легко переносимым под другие операционные системы.

Прежде всего, хотелось бы обратить внимание разработчиков на другие ОС (не только семейства Windows). На конференции будет рассмотрен один из вариантов кросс - платформенного программирования. Преимущества очевидны: привлечение пользователей других ОС, это бесплатно и не требует много времени на изменения кода программы. Практически всё, что можно написать на Qt, можно скомпилировать на ОС Windows, Linux и др.

Также, хочется отметить, что большая доля серверов находятся под управлением операционной системы Linux. Программисты, которые раньше программировали только под Windows, сталкиваются с некоторыми трудностями при разработке под другую операционную систему.

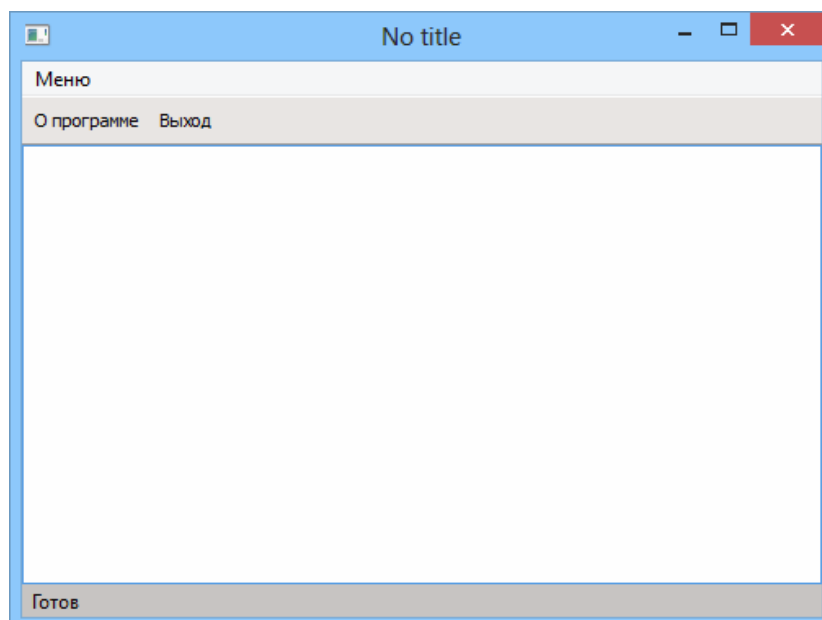


Рис. 1. Пример программы на Windows

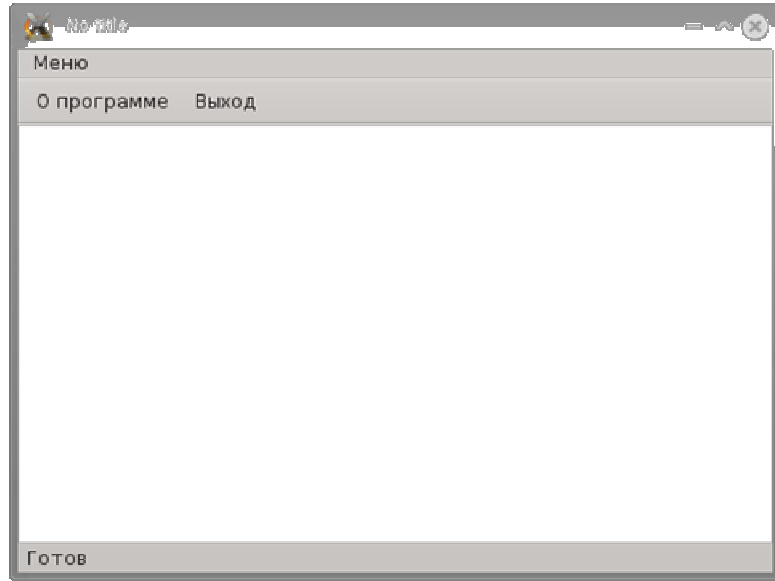


Рис. 2. Пример программы на Linux

С.В. Савинов
Научный руководитель: старший преподаватель Д.В. Бейлекчи
Муромский институт (филиал) Владимирского государственного университета
Владимирская обл., г. Муром, ул. Орловская, д.23
E-mail: kaf-eivt@yandex.ru

Исследование и разработка программного обеспечения под ОС Android для работы с метрическими системами мер

В настоящее время идёт активное развитие рынка персональных мобильных коммуникационных устройств, таких как смартфоны, коммуникаторы, персональные планшеты. Данные устройства характеризуются наличием полноценной и многофункциональной операционной системы, которая является открытой для разработки приложений сторонними разработчиками. Возможность установить специализированные приложения значительно увеличивает диапазон выполняемых функций мобильных устройств. Одной из самых распространённых операционных систем для портативных устройств в настоящее время является ОС Android.

Android - одна из самых новых разработок на рынке операционных систем. Эта ОС предназначена для широкого спектра мобильных устройств. Данная операционная система основана на базе ОС Linux. Разработчиком Android OS является компания Google. Сейчас разработками и развитием всех систем на базе Android занимается компания Open Handset Alliance, которая включает в себя множество крупнейших компаний в области информационных и телекоммуникационных технологий. [1]

Современный темп жизни человека предполагает взаимодействие с различными типами метрических систем, вследствие чего усложняется процесс обработки данных из одной системы в другую. Современные электронные средства решают эту проблему, но не всегда оказываются под рукой, а вот смартфон почти всегда находится поблизости. Всё что необходимо это произвести установку специализированного приложения, и все расчёты будут выполнены.

Таким образом, существует необходимость в разработке алгоритмов и программного обеспечения для мобильных устройств, позволяющих работать с метрическими системами мер, с возможностью обновления и модификации типов величин и алгоритмов преобразования.

Вследствие широкого распространения ОС Android было решено разработать программное обеспечение, именно для данного вида операционных систем.

Для разработки программного обеспечения для операционной системы Android активно используется Eclipse с дополнительно установленным пакетом дополнительных модулей и библиотек Android Developer Tools [2], что позволяет быстро разработать высокопроизводительное приложение. Для программирования используется объектно-ориентированный язык высокого уровня Java. В процессе разработки программы была использована именно эта среда разработки.

Разработанные алгоритмы позволяют эффективно работать с различными метрическими системами мер данных, что значительно повышает область применения. Для увеличения области использования пользователь приложения имеет возможность создать свои правила для перевода любых величин, воспользовавшись встроенным в приложение мастером добавления правил. Алгоритмы работы обеспечивают проведение финансовых расчётов, при этом данные о соответствующих валютных курсах могут браться из интернета, что обеспечивает эффективное и своевременное обновление правил перевода. При необходимости имеется возможность создания отчёта используемых пользователем данных. Также с помощью глобальной сети можно подключаться к сервисам, предоставляющим необходимую информацию для перевода, или сформировать собственную базу данных для последующего использования.

Структурная схема программного обеспечения, реализующего описанные алгоритмы, приведена на рис.1.

Программное обеспечение позволяет произвести быстрый перевод величин, с использованием входного и выходного параметра преобразования, или вывести все возможные варианты перевода, в зависимости от установленных пользователем параметров обработки данных, для наглядности представления данных.

Конвертер величин обладает удобным графическим интерфейсом, что позволяет пользователям любого уровня профессионализма оперативно выполнять операции.

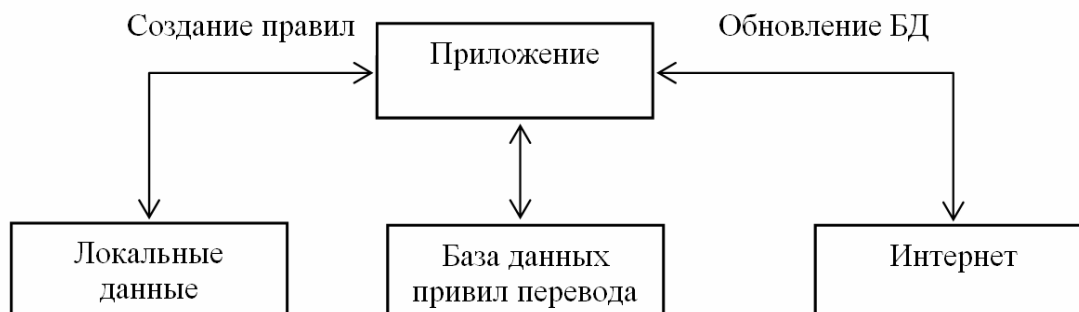


Рис. 1. Структурная схема программного обеспечения

Распространение программного продукта будет осуществляться через систему Google Play, и с помощью установочного файла формата *.apk для установки с любого носителя информации; установка не потребует от пользователя каких-либо сложных настроек. Все, что нужно – это запустить установочный файл и, если необходимо, ввести адреса Интернет онлайн-сервисов.

Таким образом, в результате данной работы были разработаны и реализованы алгоритмы обработки данных, позволяющие оперативно производить операции перевода данных из одной метрической системы мер в другую. Также в приложении была реализована работа, как с локальной базой данных правил, так и с дополнительной базой перевода, сформированной пользователем или загруженной (обновленной) через Интернет-сервисы.

Литература

1. Хашими, С. Разработка приложений для Android / С. Хашими, С. Коматинени, Д. Маклин. – СПб.: Питер, 2011. – 736 с.
2. Голощапов, А. Google Android. Программирование для мобильных устройств. – СПб.: БХВ-Петербург, 2010. – 448 с.

Н.В. Шарков, Д.Ю. Руди,
Научный руководитель: канд. техн. наук, доцент А.В. Тюкин
*Омский институт водного транспорта (филиал) ФБОУ ВПО «Новосибирская
государственная академия водного транспорта»*
Омск, ул. Ивана Алексеева, 4
E-mail: tav71.LIMA@yandex.ru

Использование Java 2 Micro Edition технологий в образовательном процессе

Необходимость широкого внедрения персональных компьютеров во все виды учебного процесса в настоящее время не вызывает сомнения, а применение ПК не просто улучшает преподавание, но и в ряде случаев качественно изменяет его. Использование компьютерного тестового контроля знаний и умений студентов, отличается объективностью, экономит время преподавателя, в значительной мере освобождает его от рутинной работы и позволяет в большей степени сосредоточиться на творческой части обучения. Тестовый контроль также дает возможность в значительной мере индивидуализировать процесс обучения путем подбора индивидуальных заданий для самостоятельной работы студентов, позволяет прогнозировать темпы и результативность их обучения [1].

Если получение информации для подготовки к лекционным занятиям не вызывает трудностей у студента, так как в этом случае ему на помощь могут прийти как печатные так и электронные книги, а также интернет, то выполнение тестовых заданий по дисциплине требует аудиторных занятий и присутствия, как преподавателя, так и учебно-вспомогательного персонала обслуживающего компьютерный класс. Одним из способов решения этой проблемы является использование Java технологий используемых в сотовых телефонах.

Для разработки программ тестового контроля по дисциплине «Физика» авторами была использована платформа J2ME компании Sun Microsystems, которая на сегодняшний день является универсальным инструментом программирования для мобильных телефонов. Поскольку Java 2 Micro Edition поддерживает большое количество разных устройств, то существуют различные конфигурации этой платформы. Для создания программы для тестирования использовалась конфигурация CDLC (Connected Limited Device Configuration), расширенная профайлом MIDP (Mobile Information Device Profile), который обеспечивает необходимую функциональность для программирования под мобильные телефоны, пользовательский интерфейс, постоянное хранение данных, таймеры, средства сетевой связи [2].

Разработанные программы для тестового контроля находятся на основном сервере ВУЗа. Эти программы можно скачать на ПК и перенести на сотовый телефон. Ознакомиться с тестовыми заданиями и проверить свои знания можно в любое время, так как сотовый телефон является постоянным спутником современного человека. К основным достоинствам таких программ для сотовых телефонов можно также отнести частичное снятие проблемы материальной базы учебного заведения, повышение активности студента и индивидуальность обучения. Все вышеперечисленные достоинства позволяют сделать вывод о необходимости более широкого применения таких методов в образовании.

Литература

1. Фаустова Э.Н. Студент нового времени: социокультурный профиль. – М., 2004. – 72 с. – (Система воспитания в высшей школе: Аналитические обзоры по основным направлениям развития высшего образования / НИИВО; Вып. 4).
2. Буткевич Е. Л. Пишем программы и игры для сотовых телефонов.– СПб.: Питер, 2006. – 204с.