

КРОССПЛАТФОРМЕННОЕ ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ QT

Бабкин А.А., Теплинский С.В.

Донецкий национальный технический университет

Недавно я столкнулся с ОС Linux, и все бы ничего, но порой не хватает того ПО, к которому привык в «винде». К примеру, под «линух» есть тьма архиваторов и сколько из них может корректно открыть запороленный rar-архив? Лично я, попробовав ark и file-roller, в конце концов пришел к тому самому WinRar'у под Wine. К тому же мне нравится Download Master, которого, кстати, часто выбрасывает под линуксом. А еще UltraISO, да тот же QIP – короче, этот список можно продолжать долго и напрашивается вывод, что половина ПО будет работать под «вайном»? С одной стороны, это вопрос предпочтений, ведь линукс предоставляет свои аналоги Kopete или LICQ вместо «квипа», KISO вместо UltraISO ит.д. Но есть ПО, без которого просто не обойтись, к примеру, FineReader (в линуксе нет нормальной программы для распознавания текста на языке, основанном не на латинице), Pragma (не нашел я и нормальной системы перевода под линукс) и далеко не все будет работать под «вайном». Не переписывать же ПО под линукс? А если вспомнить про существование MAC OS, то становится совсем грустно, вот тут на помощь приходит Qt, библиотека для разработки платформеннонезависимых приложений.

Так что же такое Qt? Qt представляет собой инструментарий разработки ПО, предназначенный для разработки на C++ кроссплатформенных приложений с графическим пользовательским интерфейсом. Есть также «привязки» ко многим другим языкам программирования: Python — PyQt, PySide; Ruby — QtRuby; Java — Qt Jambi; PHP — PHP-Qt и другие. Остановимся на C++, так как именно для C++ эта библиотека получила наибольшее распространение.

Разработка библиотеки началось в Норвегии. Норвежская компания Trolltech AG (изначально называвшаяся Quasar Technologies, откуда, собственно, и пошла аббревиатура Qt) была создана двумя норвежцами, Эйриком Енгом (Eirik Eng) и Хавардом Нортгом (Haavard Nort) с целью подзаработать и создать пару универсальных библиотек.

Сегодня практически невозможно представить себе приложение, не обладающее интерфейсом пользователя. Понятия Software и GUI (Graphical User Interface) неразрывно связаны друг с другом. Библиотеки для создания пользовательского интерфейса применяются в большом количестве операционных систем, поэтому целью были поставлены две вещи. Первое: создать библиотеки для языка C++. Второе: основой всему был именно интерфейс пользователя, поскольку это самый несовместимый момент во всем софтвере — полностью разные парадигмы графического пользовательского интерфейса.

Как известно, MS Windows использует "локальный" GUI API, который, однако, недавно был транспонирован на сетевые подключения в XP Remoting. Хотя Windows API (Application Programming Interface) — интерфейс программирования приложений, обладает всем необходимым для создания графического интерфейса пользователя. Использование этих доступных "инструментов" требует больших затрат времени и практического опыта. Даже библиотека MFC, призванная облегчить процесс написания программ для ОС Windows, не дает той простоты и легкости в процессе создания программ, как хотелось бы. Поэтому разработчики, по-прежнему, тратят массу времени

на реализацию интерфейса пользователя. Но самый большой недостаток, связанный с применением этих библиотек, — это платформозависимость.

Все UNIX завязаны на X Windows. Наконец, графика Mac OS, зашита полуаппаратно в фирменных BIOS-"тулбоксах", всегда показывала скорость и красоту. Но это было: а) проприетарно, б) ни с чем не совместимо — даже приблизительно.

Конечно, в качестве переносимого графического интерфейса можно использовать Java, да хоть и Macromedia Flash.

Macromedia Flash для крупного проекта - это не серьезно, а для JAVA пока нет аппаратной поддержки в ПК, и, как следствие большое использование ресурсов, так как JAVA размещает виртуальную машину в памяти. Qt, в свою очередь, использует низкоуровневое API, поэтому работает почти так же, как и native application.

Графический интерфейс в Qt более всего похож на Java Swing — здесь тоже существуют схемы стилей, например Windows, CDE, Motif, копирующие известные оболочки. Присутствуют также layout'ы, автоматически размещающие элементы управления; кроме того, есть "спейсеры", которые расталкивают сопредельные компоненты.

Есть также немало инструментов, напоминающих Delphi: хинты размеров, масштабирование компонент и прочее.

Как и в Swing, все элементы перерисованы от руки, то есть стандартные механизмы рисования элементов управления не применяются — вместо этого используется, например, GDI WinAPI. Автоматически определяется версия ОС и, соответственно, реализуются или игнорируются те или иные свойства, вроде прозрачности. Однако, Qt использует и некоторые стандартные диалоговые окна Windows, в частности диалоги открытия файла и настройки печати [4].

Под X11 Qt не использует тулкиты вроде Motif или Xt, так как Qt и сам такой же тулкит. Вместо этого напрямую используется Xlib — с расширениями наподобие RENDER, если они доступны.

Для ускорения и упрощения создания пользовательских интерфейсов Qt предоставляет программу Qt Designer, позволяющую делать это в интерактивном режиме.

Многие привыкли считать, что Qt - средство для создания только интерфейса пользователя. Это неверно, ведь Qt — полный инструментарий для программирования, который состоит из отдельных модулей и предоставляет:

- поддержку двух- и трехмерной графики (фактически, являясь стандартом для платформонезависимого программирования на OpenGL);
- возможность интернационализации, которая позволяет значительно расширить рынок сбыта ваших программ;
- использование формата XML (extensible Markup Language);
- STL-совместимую библиотеку контейнеров;
- поддержку стандартных протоколов ввода/вывода;
- классы для работы с сетью;
- поддержку программирования баз данных, включая поддержку Oracle, Microsoft SQL Server, IBM DB2, MySQL, SQLite, Sybase;
- и многое другое.

Кроссплатформенная реализация приложений — будущее программной индустрии, с каждым днем приобретающая все более возрастающее значение. В самом деле, если задуматься — зачем оставлять без внимания пользователей ОС UNIX, только лишь потому, что вы являетесь программистом для ОС Windows. Приняв решение в пользу платформонезависимых приложений, возможно заметное увеличение количества пользователей (клиентов). Выигрыш от реализации платформонезависимых приложений налицо: значительно сокращается время разработки, так как нет

необходимости писать код дважды, и, что не менее важно, отпадает необходимость знать специфику каждой из платформ, для которой пишется программа. Также нет надобности во время разработки продукта формировать специальные подкоманды разработчиков для каждой платформы реализации, что может значительно сократить не только время разработки, но и себестоимость продукта. И, вместе с тем, заметно улучшится и качество приложений, так как оно будет тестироваться на нескольких платформах, а ошибки будут исправляться централизованным путем в одном и том же исходном коде программы.

Несмотря на то, что предоставляемая платформонезависимость является одной из самых заманчивых возможностей библиотеки, многие разработчики используют ее также для создания приложений, работающих только на одной платформе. Делается это из тех соображений, что им нравится инструментарий и идейный подход самой библиотеки. Подобный подход предоставляет им дополнительную гибкость. Ввиду того, что требования к программному продукту с течением времени постоянно подвергаются изменениям, то, при появлении необходимости предоставить продукт для какой-либо еще платформы, не составит большой сложности.

На сегодняшний день Qt — продукт, широко используемый разработчиками всего мира. Компаний, использующих эту библиотеку, более 4000. Из числа некоторых активных пользователей Qt можно назвать такие известные компании, как: Adobe, AT&T, Cannon, HP, Bosch, Boeing, IBM, Motorola, NASA, NEC, Pioneer, Sharp, Siemens, Sony и Xerox и др.

Используя сегодня ту или иную программу, вы, возможно, и не догадываетесь, что при ее написании использовалась библиотека Qt.

Самые яркие примеры [2].

- Рабочий стол KDE (K Desktop Environment) www.kde.org, используемый в Linux и FreeBSD.
- Веб-браузер Opera www.opera.com.
- Программа Skype www.skype.com, предназначенная для голосовой связи VoIP (Voice Over IP), звонков на обычные телефоны и проведения видеоконференций через Интернет.
- Программа для обработки растровых изображений Adobe Photoshop Album www.adobe.com.
- Программа Google Earth earth.google.com, сетевая карта мира.

Qt создала себе репутацию средства разработки межплатформенных приложений, но благодаря своему интуитивному и мощному программному интерфейсу, во многих организациях Qt используется для одноплатформенных разработок. Пакет программ «Adobe Photoshop Album» - один из примеров продукта на массовом рынке Windows, написанного средствами Qt. Многие сложные системы программного обеспечения на таких вертикальных рынках, как средства анимации 3D, цифровая обработка фильмов, автоматизация проектирования электронных схем (для проектирования чипов), разведка нефтяных и газовых месторождений, финансовые услуги и формирование изображений в медицине, строятся при помощи Qt. [1]

Библиотека Qt не является монолитной библиотекой. Она разбита на отдельные модули. Каждый модуль имеет свое назначение. Например, программирование интерфейса пользователя, графики, баз данных и др. Классы модулей предоставляют разработчику механизмы, расширяющие и, вместе с тем, упрощающие создание приложений. Вершиной модульной иерархии является модуль QtCore.

Итак, библиотека разделена на несколько модулей, для четвертой версии библиотеки это [3]:

QtCore — классы ядра библиотеки, используемые другими модулями;

QtGui — компоненты графического интерфейса;
QtNetwork — набор классов для сетевого программирования.
QtOpenGL — набор классов для работы с OpenGL;
QtSql — набор классов для работы с базами данных используя язык структурированных запросов SQL. Однако, для компиляции может потребоваться SDK базы данных.
QtScript — классы для работы с Qt Scripts;
QtSvg — классы для отображения и работы с данными Scalable Vector Graphics(SVG);
QtXml — модуль для работы с XML.
QtDesigner — классы создания расширений QtDesigner'a для своих собственных виджетов;
QtUiTools — классы для обработки в приложении форм Qt Designer;
QtAssistant — справочная система;
Qt3Support — модуль с классами, необходимыми для совместимости с библиотекой Qt версии 3.x.x;
QtTest — модуль для работы с UNIT тестами;
QtWebKit — модуль WebKit, интегрированный в Qt и доступный через её классы;
QtXmlPatterns — модуль для поддержки XQuery 1.0 и XPath 2.0;
Phonon — модуль для поддержки воспроизведения и записи видео и аудио, как локально, так и с устройств и по сети;
QtCLucene — модуль для поддержки полнотекстового поиска, применяющийся в новой версии Assistant в Qt 4.4;
ActiveQt — модуль для работы с ActiveX и COM технологиями для Qt-разработчиков под Windows.

Также реализована технология WoC — widgets on canvas, с помощью которой реализована Plasma в KDE 4.1.

Библиотека обладает рядом интересных свойств и особенностей. Qt полностью объектно-ориентированная библиотека. Новая концепция ведения межобъектных коммуникаций, именуемая "сигналы и слоты", полностью заменяет былую и ненадежную модель обратных вызовов. Также имеется возможность обработки событий. Например, нажатия клавиш клавиатуры, перемещения мыши и т. д.

Предоставляемая система расширений (plug-ins) позволяет создавать модули, расширяющие функциональные возможности приложений. Эти расширения пользователи программы могут получать не только от разработчиков основного ПО, но и от других разработчиков.

Программы, реализованные с помощью Qt, могут использовать язык сценариев Qt Script. Эта технология позволяет пользователям приложения расширить возможности без изменения исходного кода и без перекомпоновки самого приложения, изменить "поведение" приложения.

Qt прекрасно документирована, благодаря чему всегда можно почерпнуть любую интересующую информацию об этой библиотеке, используя программу Qt Assistant.

Библиотека использует собственный формат проекта, именуемый *.pro файлом, в котором собрана информация о том, какие файлы будут скомпилированы, по каким путям искать заголовочные файлы и много другой информации. Впоследствии, при помощи утилиты qmake из них получаются makefile для make-утилиты компилятора. Также имеется возможность работы при помощи интеграторов с Microsoft Visual Studio 2003/2005/2008.

Qt может применяться с различными лицензиями. Для коммерческих приложений следует приобрести лицензию Qt; для opensource ПО, можно использовать версию с открытым исходным кодом (с лицензией GPL). [1]

В заключении скажу, что в результате знакомства с Qt, если я что-то пишу на C++, то обязательно использую Qt даже не столько для кроссплатформенности, сколько из-за удобства и интуитивно понятной реализации библиотеки. К примеру, я ничего не знаю о программировании GUI для линукс или о сетевом программировании вообще под любую платформу, однако это мне не мешает делать сетевые и графические приложения под большинство используемых сегодня платформ сразу!

Литература

- [1] Ж. Бланшет, М. Саммерфилд Qt 4: Программирование GUI на C++. 2-е дополненное издание. — М.: «КУДИЦ-ПРЕСС», 2008
- [2] Макс. Шлее Qt 4: Профессиональное программирование на C++. — СПб.: «БХВ-Петербург», 2007
- [3] <http://ru.wikipedia.org/wiki/Qt>
- [4] <http://www.citforum.ru/programming/qt/>