

Быстрая разработка приложений в Linux

Предисловие

В течение нескольких прошедших лет мною были написаны несколько статей, в которых мне хотелось бы рассказать о том, как в обычных домашних условиях любой пользователь ПК может сам довольно быстро научиться создавать компьютерные программы. Главным помощником в этом случае может стать любая из описанных ниже систем быстрого программирования.

Статьи предназначены для тех пользователей ПК, которых можно отнести к категории «начинающих любителей программирования». Описанные в этих статьях методы создания программ ни в коем случае нельзя считать образцами программистского совершенства или учебным материалом по программированию. В этих статьях автор просто показывает вам те методы и способы, которыми он решает те или иные задачи. Кроме того следует помнить, что все приведенные в этой серии статей программы могут свободно распространяться в соответствии с известным принципом «как есть», т.е. без гарантий автора и без претензий со стороны пользователя.

При подготовке указанных выше статей к публикации, я пришел к выводу, что две основные статьи стоит опубликовать в виде небольших книжек. Так появились книжки «Программирование в Borland Kylix» и «Программирование с помощью QT», тексты которых располагаются в Интернете на моем сайте, расположенном по адресу <http://r3xb-tga.narod.ru/>.

Коротко о RAD

Создание систем быстрой разработки компьютерных программ (по-английски **RAD** – Rapid Application Development) вызвано требованием нашего времени.

Начало было положено фирмой **Microsoft**, создавшей среду для возможностей визуального программирования **Visual Basic**. Эта среда программирования сформировала новый стиль взаимодействия разработчика программы с компьютером, позволяя наглядно конструировать пользовательский интерфейс с помощью мыши, а не обычным для прежних времен путем написания кодов, их последующей неоднократной трансляцией и выполнением программы, после чего только и можно было посмотреть, как же это выглядит на экране.

Visual Basic нашел широкий спрос при создании компьютерных программ самого разного назначения, помог открыть мир программирования для людей, не слишком искушенных в тонкостях программирования на низком уровне, но он не свободен от многих проблем и не всегда пригоден для создания приложений, которые должны работать с высокой скоростью. Главные из недостатков – низкая производительность разрабатываемых приложений при их выполнении, недостаточная строгость и объектная ориентированность языка.

Созданные фирмой **Borland** системы **Delphi** и **C++ Builder** – это следующий шаг в развитии среды быстрой разработки приложений. Они исправляют многие дефекты, обнаруженные в **Visual Basic**. Разработчики этих систем создали инструменты, которые на первый взгляд выглядят похожими на среду **Visual Basic**, хотя в действительности они заметно лучше.

Интегрированная среда разработки в **Delphi** и **C++ Builder** выглядит одинаково. Весь пользовательский интерфейс, все библиотеки, все приемы работы с этими системами практически одинаковы. Если быть более точным, то они различаются только в силу разного времени выпуска соответствующих версий. Версии **C++ Builder** выпускаются на полгода позже версий **Delphi** с аналогичными номерами. Поэтому каждая версия **C++ Builder** совершеннее аналогичной версии **Delphi**, но слабее последующей версии **Delphi**.

Основное различие **Delphi** и **C++ Builder** заключается в языках программирования, которые лежат в их основе. **Delphi** базируется на языке **Object Pascal**, а **C++ Builder** базируется на языке **C++**. Эти языки на сегодняшний день являются самыми мощными, при этом многие авторы отдают первенство языку программирования **C++**. Исторически сложилось так, что эти два языка программирования сначала существенно различались по своим возможностям, но со временем все более сближаются. Сейчас оба они представляют собой прекрасные инструменты объектно-ориентированного программирования, различающиеся, в основном, синтаксисом. Но **C++** все-таки богаче и опережает аналогичные версии **Object Pascal**. С этой точки зрения он предпочтительнее. Правда, эти преимущества **C++** перед **Object Pascal** проявляются только в достаточно сложных приложениях.

Фирма **Borland** позаботилась о том, чтобы приложения, разработанные на **C++ Builder** и на **Delphi**, можно было достаточно просто конвертировать друг в друга. Таким образом, в одной из этих систем вы можете использовать свои наработки, сделанные в другой системе.

Для всех желающих поближе познакомиться с быстрым созданием компьютерных Windows-приложений советую в Интернете на моем сайте, расположенном по адресу <http://r3xb-tga.narod.ru/>, скачать текс книги «Начальный курс быстрого программирования». Книга предназначена для начинающих интересоваться процессом создания компьютерных программ и

даже для тех, кто не знаком пока ни с одним из языков программирования. На мой взгляд, книга может быть полезной и для фанатов Linux'а.

Перечисленные выше системы для быстрой разработки компьютерных программ служат только для операционной системы **Windows**. Существуют аналогичные системы и для прогрессирующей операционной системы **Linux**.

Фирмой **Borland** для **Linux** создана среда программирования **Kylix**, которая по способам разработки программ полностью соответствует **C++ Builder**. В Linux используется также среда программирования для **KDE**, использующая **QT** – компоненты. Кроме того, быстрое создание компьютерных программ вполне возможно также с использованием таких программных продуктов, как **QT-designer** и **Glade2**.

Далее смотрите материал по быстрому программированию в книжках «Программирование в Borland Kylix» и «Программирование с помощью QT», где вы найдете много интересного о том, как можно быстро создать компьютерную программу в среде программирования **Kylix** и с помощью компонентов **QT**.