



Школа LXF

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

Программирование в Linux – так ли это сложно?

Программирование в Linux должно быть интуитивно понятно. Если для вас это не так, **Александр Казанцев** расскажет, как разрешить данную проблему.



Наш эксперт
Александр Казанцев
Старший научный сотрудник ЗАО «Тырнет» и автор образовательного проекта EduMandriva в настоящий момент посвящает все свое время скреплению железа и софта под свободными лицензиями.

» Программа на Scratch интуитивно понятна. Даже такая сложная.

Оглянувшись назад на наши публикации в рубрике Школа LXF, вы заметите одну особенность: вопросам программирования не уделялось много внимания. Да, среды программирования упоминались в статьях про исполнители, web-сервисы, учебники по информатике. Но, как показывают поступающие к нам вопросы от учителей, тема до сих пор, спустя несколько лет внедрения СПО в школах, как никогда актуальна. В рамках *LinuxFormat* опубликовано немало уроков по написанию программных продуктов на разных языках и в разных средах. Но единого обзора, увы, не было. И это надо срочно исправить.

Поэтому я решил снова пройтись по азам инструментария для изучения алгоритмических основ и программирования в рамках школьного курса информатики. Возможно, некоторые вещи для вас будут уже прописными истинами – тогда эта статья не для вас. Но многие найдут в ней нужную и полезную информацию, особенно те, кто пропустил предыдущие этапы перехода на Linux.

Первым делом давайте взглянем в школьную программу. Какие языки программирования и среды нужны и уже описаны в учебниках и методике, а также в заданиях ЕГЭ (что может быть самым важным выбором)? Ничего необычного – Паскаль, Бейсик, Алгоритмический язык и Си. Плюс упоминаются *VisualBasic*,

«Тема программирования в школах как никогда актуальна.»

Мы об этом писали

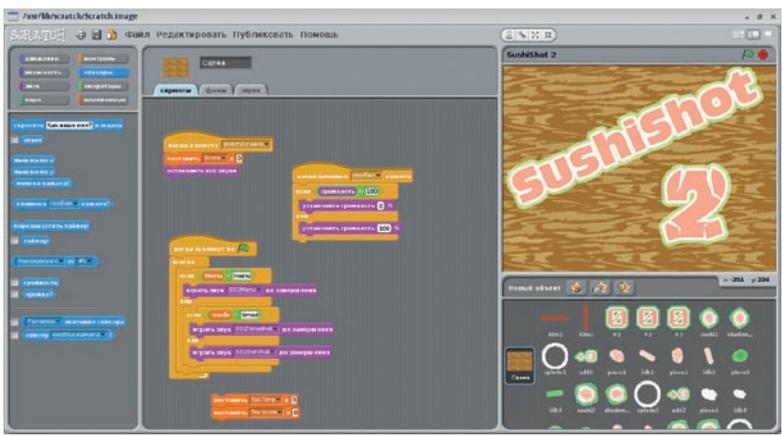
- Мы уже касались вопросов данной темы ранее. Чтобы полнее понять, о чем идет речь, советуем прочитать следующие статьи:
- » LXF126-127 Школа LXF: Исполнители и псевдокод
- » LXF135 Школа LXF: СПО и новый стандарт для начальных классов
- » LXF143 Школа LXF: Олимпийский ЕГЭ
- » LXF148 Школа LXF: Школа в «облаках»

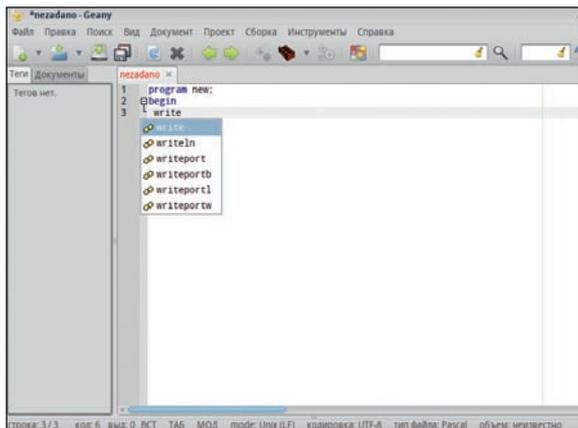
Gambas и *Delphi* в качестве сред для изучения ООП. Для младших классов предлагаются свои решения – обычно это аналоги NetLogo (Черепашка, Кенгуренок, ЛогоМиры и т.п.). Также встречаются исполнители, представляющие собой смесь Черепашки и Паскаля (Паркетчик в учебниках А. Г. Гейна). Мы не будем касаться исполнителей (о них у нас была отдельная статья – см. врезку

вверху), а поговорим о языках программирования.

Scratch. Начало

Начнем мы со *Scratch* (<http://scratch.mit.edu>) – языка и среды написания программ в виде «кирпичиков», выросшего из NetLogo и исповедующего его принципы. Для начальной школы и уроков технологии в 5-6-м классе лучшего способа вливания в траекторию алгоритмизации не придумаешь. В процессе создания с помощью *Scratch* интерактивных историй, игр и анимаций ученик может получить базовые навыки программирования приложений. Также в процессе работы в данной среде развиваются логическое и алгоритмическое мышление; системный подход к решению проблем; развитие идей, начиная от исходной концепции и до конечного результата создания проекта; навыки и опыт создания интерфейса пользователя; умение отладки и тестирования результата; развитие настойчивости и умения концентрироваться. Упор делается именно на алгоритмическую составляющую: программа однозначно задает компьютеру его действия шаг за шагом, а само составление программ требует не специальных знаний, а тщательной и ясной продуманности действий.





» Попробовав возможности автодополнения Geany, школьники больше не хотят возвращаться к скучным синим окошкам Паскаля.

Работая со *Scratch*, ученики изучают следующие основные понятия и концепции:

- » Интерфейс пользователя
- » Команды, блоки, процедуры и скрипты
- » Работу с объектами, методами и событиями
- » Различные виды и типы данных, константы, переменные, списки
- » Работу с переменными и присваиванием
- » Выражения и операции
- » Основы рисования и черчения
- » Ввод и вывод символьных данных
- » Управление процессами – последовательные и параллельные процессы, циклы и ветвления.
- » Событийный аппарат и взаимодействие объектов
- » Алгоритмизацию и моделирование.

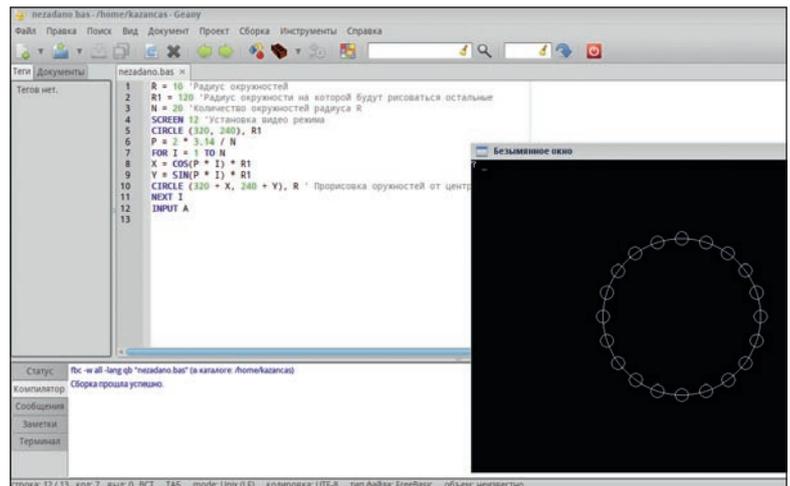
Но *Scratch* не всемогущ. В нем вы не сможете

- » задавать функции;
- » использовать параметры и аргументы;
- » использовать рекурсию;
- » определять свои классы объектов;
- » обрабатывать запросы и ошибки;
- » работать с вводом/выводом файлов.

В следующих статьях мы постараемся уделить побольше внимания данному отличному продукту, и ваши письма на info@linuxformat.ru подскажут нам, интересно ли вам данное направление.

Geany — Великий и Ужасный

Pascal и BASIC – два языка, с которых начинается погружение в увлекательный мир программирования для многих школьников. И начинался для нас – тех, кто дает знания в текущий момент. *TurboPascal* от Borland и *QBasic* от Microsoft стали де-фак-



» Графическая программа на *QBasic*. И работает в Linux!

то стандартом для многих школ в части процедурных языков. Синенькие оболочки эпохи DOS, увы, до сих пор властвуют в школах России, и тут есть одна причина, кроме банальной устаревшей техники: начинать сразу с объектно-ориентированного «накидывания элементов на форму» не является лучшим способом погружения в увлекательный мир программирования. Абстрагирование от внешнего вида и сосредоточение на структуре кода – вот главное, за что ценили данные среды и языки. Доводы про обучение с самого начала на Си или других языках, типа Python, возможно, правильны. Но если глянуть на ЕГЭ, то сразу станет понятно, почему процедурные языки еще в ходу.

В Linux мы имеем две отличных замены проприетарных компиляторов и интерпретаторов: *FreePascal* и *FreeBasic*, которые (для *FreeBasic* это достигается с помощью специального ключа) совместимы с *TurboPascal* и *QBasic*, но по сути предлагают гораздо больше возможностей.

Но писать программу в текстовом редакторе и запускать ее компиляцию с помощью командной строки современным детям очень и очень трудно. И тут на помощь нам приходит Geany.

Geany – это свободная среда разработки программного обеспечения, написанная с использованием библиотеки *GTK2*. Geany не включает в свой состав компилятор. Вместо этого используется GNU Compiler Collection (или любой другой компилятор) для создания исполняемого кода. Особенности Geany являются:

- » Подсветка исходного кода с учетом синтаксиса используемого языка программирования (язык определяется автоматически)
- » Автозавершение кода
- » Автоматическая подстановка закрывающих тэгов HTML / XML
- » Простой менеджер проектов
- » Поддержка модулей расширения
- » Встроенный эмулятор терминала
- » Поддержка большого количества кодировок
- » Гибкий интерфейс

Вы можете использовать Geany не только для написания программ на Паскале или Бейсике, но также как простой HTML-редактор, для написания программ на C, Haskell, Python и множестве других языков.

Еще раз хотим напомнить, что вам нужно установить компилятор для данного языка, чтобы получить возможность работы в Geany. К примеру для дистрибутива Edumandriva нужно установить пакеты **fpс** и **FreeBASIC**.

Еще Бейсики

BASIC (или Бейсик) до сих пор остается самым простым языком программирования, и *FreeBASIC* не является единственным выбором в Linux. Существуют и другие среды. Наиболее известными из них являются *BASIC256* и *Gambas*.

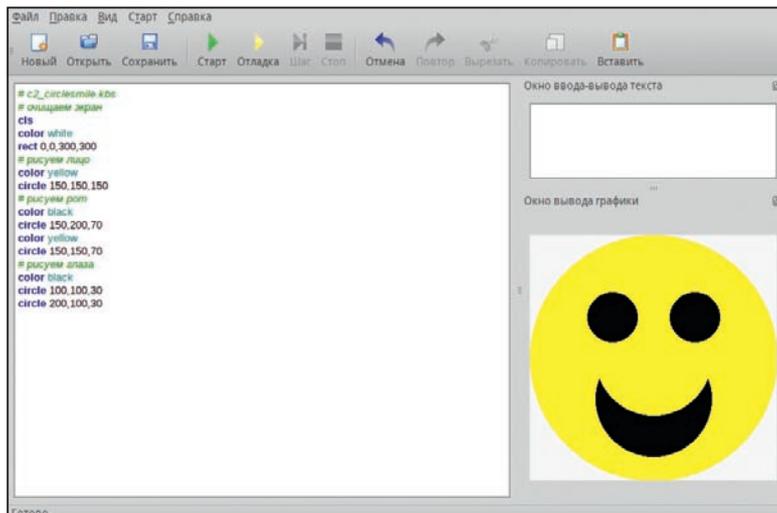
Как использовать Geany

На сайте IBM Developers Works размещен прекрасный цикл по работе в Geany с разными средами программирования:

- » Часть 1. Основы работы в Geany. Создание простого приложения – <http://www.ibm.com/developerworks/ru/library/l-geany/>
- » Часть 2. Работа в Geany с проектом – <http://www.ibm.com/developerworks/ru/library/l-geany2/>
- » Часть 3. Работа в Geany с исходным кодом на языке *Free Pascal* – <http://www.ibm.com/developerworks/ru/library/l-geany3/>

Чтобы программировать в Geany с использованием синтаксиса *QBasic* нужно, установив *FreeBASIC*, изменить строку в **Сборка** – Установить команды сборки на

```
fbс -w all -lang qb "%f"
```



» Все в одном месте. Очень удобно.

BASIC256 – это упрощенная версия языка BASIC, созданная для обучения детей младшего возраста основам программирования. Она использует традиционные управляющие структуры, такие как `gosub`, `for/next` и `goto`, что помогает детям легче понять, как происходит управление выполнением программы. Язык имеет встроенный графический режим, позволяющий в считанные минуты рисовать на экране картинки. Набор подробных и простых для повторения уроков знакомит с концепциями программирования посредством веселых упражнений. Данный язык набирает популярность и обрастает документацией и методикой (см. врезку внизу).

Gambas — объектно-ориентированный диалект языка BASIC, дополненный интегрированной средой разработки и работающий на GNU/Linux и других UNIX-подобных операционных системах. Он представляет собой средство разработки программного обеспечения, аналогичное *Visual Basic* в плане разработки программ на бейсике. *Gambas* задумывался как альтернатива *Microsoft Visual Basic* для разработчиков, которые решили перебраться на Linux.

Gambas – рекурсивный акроним от англ. *Gambas Almost Means BASIC*, что в дословном русском переводе выглядит как «Gambas Почти Означает Basic», а более привычно русскому уху «Gambas – почти Basic». Само слово *Gambas* с испанского переводится как «креветка», которая, собственно, и изображена на логотипе. *Gambas* позволяет:

- » Работать с базами данных, такими как *MySQL*, *PostgreSQL*, *Firebird*, *SQLite* и *ODBC*.
- » Создавать приложения KDE с *DCOP*.
- » Использовать 3D API *OpenGL*.
- » Портить программы с *Visual Basic* на *Gambas*.
- » Создавать сетевые решения.

Особенно стоит отметить, что *Gambas* используется как альтернатива *VisualBasic* в учебниках по информатике авторства

Методика по BASIC256

Вы можете найти и свободно скачать методику и руководства по следующим ссылкам:

- » Справочник синтаксиса *BASIC256*
<http://doc.basic256.org/doku.php?id=ru:start>
- » Джеймс М. Рено Хотите научиться программировать?
<http://www.altlinux.org/Books:Basic-256>
- » Компьютерный практикум с использованием *BASIC256*
<http://metodist.lbz.ru/authors/informatika/3/>

Почитаем про КуМир

Пройдя по данным ссылкам, вы сможете найти исчерпывающую информацию об использовании системы *КуМир* в учебном процессе.

- » Курс алгоритмизации с использованием исполнителей системы *КуМир* и автоматического тестирования
<http://server.179.ru/wiki/?page=DenisKirienko/Kumir>
- » Учебные пособия по *КуМир*
<http://www.niisi.ru/kumir/books.htm>
- » Решения задач в *КуМир*
<http://lpm.org.ru/kumir2/files/solutions.7z>

Н. Д. Угриновича, что автоматически снимает вопрос о методической составляющей.

Алгоритмический язык

Если же вы поклонник национального продукта и считаете, что использование иностранного языка при написании программ вредно (похоже, так же считают создатели 1С), то можете обратить свой взор на алгоритмический язык и среду написания программ на нем – *КуМир*.

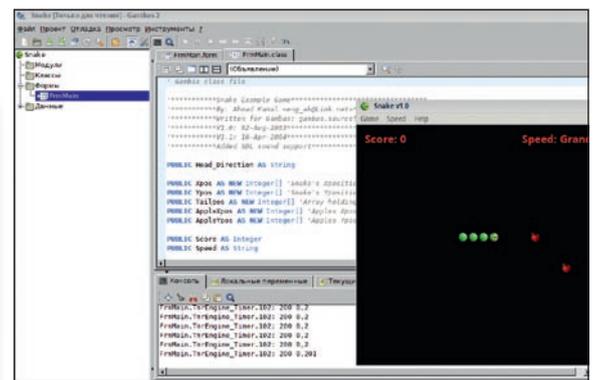
КуМир – это система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе. Основана на методике, разработанной во второй половине 1980-х годов под руководством академика А. П. Ершова. Эта методика широко использовалась в средних школах СССР и России. В системе *КуМир* используется придуманный А. П. Ершовым школьный алгоритмический язык – простой алгоподобный язык с русской лексикой и встроенными командами управления программными исполнителями (Робот, Чертежник).

Кроме этих исполнителей, доступны Водолей (задачи на переливание между сосудами), Кузнечик (аналог машины Поста) и Черепаха (замена Logo Turtle), а также Рисователь. По всем исполнителям доступна исчерпывающая справочная информация, а система команд за счет русского языка понятна всем.

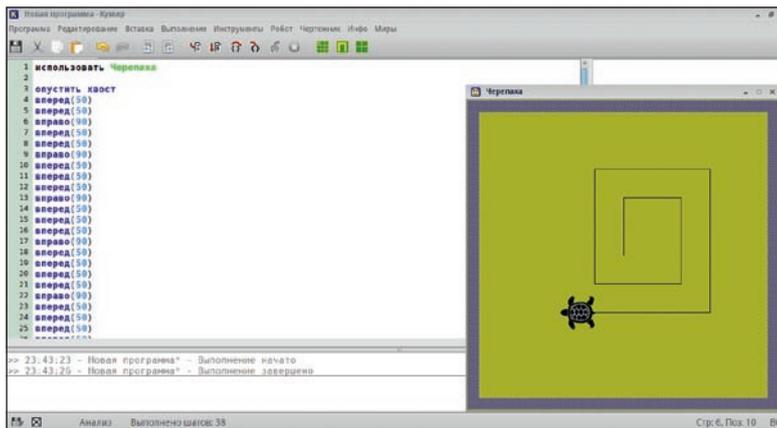
Хотим формы

Мы уже упомянули одну среду, предназначенную для изучения объектно-ориентированного принципа программирования. Но таких сред много, и они в основном представляют собой RAD IDE. Если на Windows вы привыкли к Delphi, то в Linux стоит обратить внимание на Lazarus.

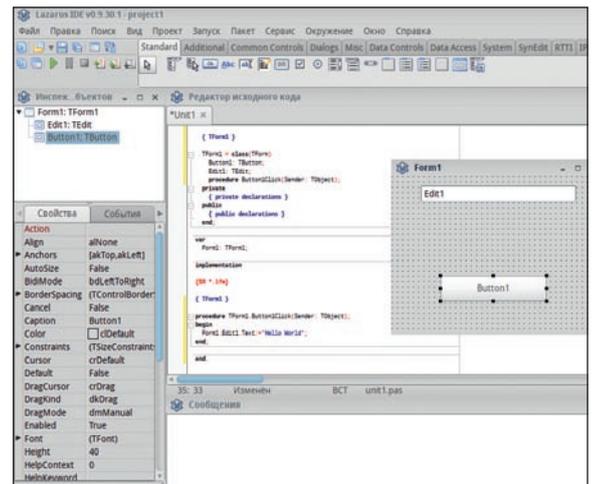
Lazarus представляет собой свободную среду разработки программного обеспечения для компилятора *Free Pascal*. Интегрированная среда разработки предоставляет возможность кросс-платформенной разработки приложений в Delphi-подобном



» *Gambas* не так прост, как кажется.



➤ Черепашка есть и в КуМире.



➤ Ну чем не Delphi?

окружении и на первый взгляд неотличима от 7-й версии Delphi. Те же визуальные компоненты, такой же принцип построения приложений. Более того, вы можете импортировать в Lazarus проекты из Delphi, и если не использовались какие-нибудь экзотические примочки, это сработает. Вам останется только поправить ОС-специфические вещи – такие, к примеру, как различие задания путей файлов в Windows и Linux – и перекомпилировать проект (особенно в последних версиях, начиная с 0.9.30). Дьявол, как говорится, таится в мелочах. В нашем случае «мелочью» будет работа с базами данных. В Lazarus применяется отличный механизм, достаточно сложный для неподготовленного пользователя. С другой стороны, разобравшись в нем, вы не будете привязаны к *Interbase*, а сможете работать с *MySQL*, *PostgreSQL*, *Firebird* и другими СУБД. Также стоит отметить наличие портов под Lazarus таких известных компонентов, как *Indy*, *GLScene* и других. Много полезной информации по использованию Lazarus можно найти на сайтах <http://lazarus.su/> и <http://freepascal.ru/>, а также на русской Вики http://wiki.lazarus.freepascal.org/Main_Page/ru.

Поклонников C# (да и VB) обрадует тот факт, что их программы, написанные на данном языке, можно запускать в Linux, при минимальной переделке. И помогает в этом проект Mono и среда разработки *Monodevelop*. Также любителям *VisualBasic*

стоит обратить внимание на *KBasic* (<http://www.kbasic.com>) – открытый аналог VB на Qt, который очень похож на его шестую версию.

И это только начало списка. Linux хорош тем, что вырос из Unix. А следовательно, большинство языков программирования имеют реализацию под Linux. Хотите перейти на C? Тогда можете установить *GCC*, и воспользоваться простотой *Geany* или мощью *Anjuta* в совокупности с графическими возможностями *GTK2* (а еще лучше – современного *GTK3*). Хотите изучить C++? Тогда, начав опять же с *Geany*, впоследствии можете перейти

«Большинство языков программирования реализовано под Linux.»

к связке с прекрасным графическим фреймворком Qt (на нем основана среда KDE4, к примеру) и открыть для себя *QtCreator* или *Code::Blocks*. Нравится

Python – *Geany* или *Eric* помогут вам погрузиться в язык. Ruby, PHP, Perl, Java, JavaScript – список можно продолжать до бесконечности, и я уверяю вас, что скорее всего для каждого языка найдется подходящая среда, пригодная для обучения программированию. Напишите нам, если вам интересна данная тема: какие языки программирования стоит рассмотреть в рамках школьной рубрики, какие проблемы вас волнуют; или оставьте сообщение на форуме linuxformat.ru. **LXF**

26 сентября на сайте Сообщества педагогов **Pedsovet.su** стартовал конкурс статей об использовании свободного программного обеспечения в образовательных учреждениях.

Цель конкурса — сделать свободное ПО доступным каждому педагогу.

Поделитесь своим опытом с коллегами – и получите шанс выиграть ценные подарки

от журнала **Linux Format** и образовательного проекта **EduMandriva**.



Pedsovet.su
СООБЩЕСТВО ВЗАИМОПОМОЩИ УЧИТЕЛЕЙ

Подробности – на странице конкурса <http://pedsovet.su/publ/40-1-0-1219>