НОУ «ИПС-Университет г. Переславля им. А. К. Айламазяна»
Институт Программных Систем РАН
АНО «Институт логики, когнитологии и развития личности»
ALT Linux

Четвёртая конференция «Свободное программное обеспечение в высшей школе»

Переславль, 30 января — 1 февраля 2009 года

Тезисы докладов

Москва,
ALT Linux,
2009

В книге собраны тезисы докладов, одобренных Программным комитетом четвёртой конференции «Свободное программное обеспечение в высшей школе». (Переславль, 30 января — 1 февраля 2009). Тезисы печатаются на основе электронных форм, представленных авторами, которые несут ответственность за содержание и оформление текста.

Программа конференции

30 января

8.30: Отъезд участников конференции от гостиницы «Космос», м. ВДНХ

12.00: Обед в Переславле

13.00: Отъезд участников в ИПС РАН от гостиницы «Переславль»

Дневное заседание
13.30–16.30

13.30–13.45: А. Е. Новодворский. Информация оргкомитета

13.45–14.00: С. М. Абрамов. Приветственное слово

14.00–14.40: А. А. Якушин

Вечернее заседание
16.50–18.50

31 января

Утреннее заседание
10.00–13.30

Дневное заседание
14.30–16.10

Вечернее заседание
17.00–18.40

1 февраля

Утреннее заседание
10.00–12.40

А. А. Якушин Москва, ALT Linux

Результаты разработки и апробации пакета свободного программного обеспечения (ПСПО) для общеобразовательных учреждениях Российской Федерации в 2007–2008 годах

Аннотация

В докладе изложены основные итоги работ по разработке и апробации пакета свободного программного обеспечения (ПСПО) для общеобразовательных учреждениях Российской Федерации в 2007–2008 годах. Рассматриваются количественные и качественные итоги реализации проекта, а также основные проблемы, возникающие при переходе общеобразовательных учреждений на использование свободного программного обеспечения в учебном процессе.

В соответствии с Распоряжением Правительства Российской Федерации от 18 октября 2007 г. №1447-р, Федеральным Агентством по образованию в конце 2007 года был проведен открытый конкурс на выполнение работ по проекту «Разработка и апробация в пилотных субъектах Российской Федерации пакета свободного программного обеспечения для использования в общеобразовательных учреждениях Российской Федерации в 2007–2008 годах».

В настоящее время следует отметить, что основная часть работ, запланированных условиями конкурса, выполнена.

По состоянию на 15 января 2009 года ПСПО установлено:

	ОУ, в которых установлено ПСПО	
Наименование пилотного субъекта РФ	Количество	%
Республика Татарстан	623	100
ОУ Казани	100	
Городские ОУ	159	
Сельские ОУ	364	

Пермский край	**350**	104
ОУ Перми	117	
Городские ОУ	100	
Сельские ОУ	133	
Томская область	134	108
ОУ Томска	63	
Городские ОУ	4	
Сельские ОУ	67	
Всего по пилотным регионам	**1107**	102
Другие регионы	996	
Всего	**2103**	

В соответствии с условиями конкурса работы по проекту проводились в два этапа. На первом этапе необходимо было создать собственно пакет свободного программного обеспечения (ПСПО), а на втором этапе провести его апробацию, доработку и внедрение.

В результате выполнения работ первого этапа, направленных на создание ПСПО, был выявлен ряд проблем, потребовавших быстрого решения:

1. При анализе состояния аппаратного обеспечения ОУ пилотных регионов было выявлено, что при первоначально заявленном требовании по объему оперативной памяти компьютеров в 128 Мб реально около трети имеющейся вычислительной техники обладает вдвое и более низким объемом. На таких слабых компьютерах была принципиально невозможна установка программного обеспечения с требованиями, изложенными в конкурсной документации, что могло привести к срыву работ по проекту в целом.

   Данная проблема была решена путем создания в кратчайшие сроки терминального решения на основе LTSP (англ. Linux Terminal Server Project) — пакета дополнений для GNU/Linux, позволяющего подключить большое количество низкопроизводительных тонких клиентов к серверу под управлением ОС Linux. Приложения выполняются на сервере, получая входные данные от тонкого клиента и отображая результат на его экране.

Данное решение в дальнейшем себя полностью оправдало на этапе внедрения.

2. При подготовке дистрибутивов ПСПО были выявлены значительные проблемы в части русификации целого ряда программных продуктов и их адаптации к условиям конкурса. В ходе этой работы значимые изменения, направленные на улучшение русскоязычного интерфейса, поддержки требуемых форматов файлов данных, были внесены в 25% программных пакетов, входящих в разработанные дистрибутивы.

После создания комплекта ПСПО была проведена его апробация в различных ОУ пилотных регионов, по результатам обратной связи с преподавателями и с учетом их пожеланий в дистрибутивные комплекты был внесен ряд существенных изменений и дополнений, касающихся улучшения программ установки, документации, взаимодействия различных компонентов.

В итоговом варианте комплект ПСПО состоит из четырех дистрибутивов:

Легкий Линукс — оптимизирован для установки и работы на компьютерах от 128 до 256 МБ памяти и процессор от PI 233 МГц на 2CD.

Линукс Юниор — для компьютеров с объемом памяти от 256 МБ до 1 ГБ на 2 CD.

Линукс Мастер — дистрибутив на DVD, требует память от 1ГБ.

Линукс Терминал — Terminal Server. Специальное решение, разработанное в дополнение к основным конфигурациям. Этот дистрибутив предназначен для использования в компьютерных классах с сервером.

Кроме того, по настоятельной просьбе пользователей, к комплекту ПСПО добавлены два диска с дистрибутивами LiveCD и LiveDVD, дающие возможность ознакомится с комплектом без установки на жесткий диск.

На втором этапе работ были выполнены следующие мероприятия:

- Создан сайт проекта `http://linux.armd.ru`, на котором размещены все материалы по проекту, доступно скачивание ПСПО, организовано взаимодействие с участниками проекта;

- Определен список пилотных ОУ по всем пилотным регионам (623 ОУ Республики Татарстан, 337 ОУ Пермского края, 124 ОУ Томской области, всего 1084);
- Разработан регламент технической поддержки на период апробации;
- Выпущен тираж ПСПО для апробации в 3-х пилотных регионах (весь тираж передан в регионы) и тираж для ознакомления общим количеством в 1500 экземпляров; для ознакомления в пилотные регионы направлено 1215 комплектов ПСПО;
- Развернуты службы технической поддержки в пилотных регионах;
- Закончена установка во всех школах;
- В связи со значительным интересом к проекту со стороны ОУ непилотных регионов Федеральное Агентство по образованию разрешило образовательным учреждениям присоединяться к данному проекту. Получено более 900 заявок от школ других регионов на присоединение к проекту, в школы, присоединяющиеся к проекту, направлены комплекты ПСПО.

В процессе внедрения ПСПО отмечено, что для успешной работы со свободным программным обеспечением в ОУ необходимо создание инфраструктуры образовательного учреждения на основе СПО, а также создание единого государственного репозитория свободных программ для нужд образовательных учреждений, позволяющих оперативно и единообразно управлять программным обеспечением компьютеров ОУ. В связи с острой необходимостью данных работ ФАО было объявлено два дополнительных конкурса, по результатам которых до конца 2008 года были разработаны: школьный серверный дистрибутив и прототип репозитория для нужд ОУ.

В целом реализацию данного проекта можно считать успешной и многообещающей. И дело в данном случае не только в том, что требуемые конкурсом работы были выполнены с превышением. Главным показателем успешности проекта можно считать добровольное присоединение к нему более 990 ОУ по всей территории РФ, которые абсолютно самостоятельно приняли подобное решение. Подобное движение по столь массовому присоединению к пилотному проекту отмечается впервые и показывает, что идеи использования свободного ПО в ОУ востребованы на местах.

В ходе внедрения ПСПО в пилотных регионах накоплен значительный опыт не только по использованию свободного программного обеспечения, но и по преодолению целого ряда системных проблем информатизации ОУ РФ. Отмечен ряд факторов, существенно сдерживающих массовый переход на СПО. К ним можно отнести:

Крайне низкую материально-техническую базу ОУ и общую изношенность оборудования. В ходе внедрения ПСПО, как уже отмечалось выше, около трети ОУ имеет компьютеры с объемом оперативной памяти менее 128 Мб, что примерно соответствует уровню десятилетней давности. Во многих случаях отмечалась неработоспособность дисководов лазерных дисков или их отсутствие, малый объем и существенная изношенность жестких дисков, устаревшее периферийное оборудование. Одним из путей существенного улучшения материально-технической базы ОУ при минимальных затратах является массовый переход на использование терминальных классов. Разработанное в процессе реализации конкурса терминальное решение показало свою жизнеспособность; при условии использования современных терминалов и некоторой доработке, Линукс Терминал может с успехом использоваться в ОУ, решив, помимо проблем устаревшего оборудования, проблемы массового администрирования школьных компьютеров в условиях нехватки квалифицированных кадров за счет простоты настройки и обновления.

Небольшое количество отечественных образовательных программных продуктов для использования с ПСПО. Долгое время образовательные, обучающие и тестовые программы создавались исключительно для платформы Windows с использованием несвободных компонентов. Некоторая часть данных продуктов пригодна для использования с ПСПО, однако в целом необходимо проведение работ по переносу накопленно контента на свободную платформу, анализ лицензионного отягощения имеющихся программ, избавление государства от затрат по их массовому тиражированию. Особенно это касается программного обеспечения по сдаче ЕГЭ. Без перевода программного обеспечения по сдаче ЕГЭ на ПСПО массовое рапространение последнего просто невозможно.

Отсутствие достаточного количества образовательной литературы. В рамках выполнения проекта создано значительное количество учебно-методических материалов, существенно облегчающих миграцию на свободное ПО. Однако для полноценного решения данной проблемы необходимо создание нового современного

учебника по курсу «Информатика и ИКТ», всех необходимых учебных и методических материалов для данного учебника. Кроме того, необходима постоянная разработка новых учебно-методических материалов по использованию ПСПО в преподавании других предметов, организации информационного пространства образовательного учреждения.

Отсутствие тематического сообщества. Реализация проекта показала, что для его успешного развития необходимо постоянное взаимодействие с учителями; энтузиастами СПО на местах, во многих случаях готовых оказывать безвозмездную помощь в продвижении свободных продуктов; с преподавателями и студентами ВУЗов. Обычной для СПО практикой является постепенный процесс формирования тематического сообщества, заинтересованного в продвижении проекта и оказывающего ему существенную, зачастую неоценимую помощь. Для формирования такого сообщества необходимо формирование так называемой «точки входа», то есть создание портала СПО. За основу подобного портала может быть взат сайт проекта, весьма успешно используемый в течение всего срока внедрения ПСПО. При минимальных затратах со временем подобный портал может вырости в тематическую социальную сеть, оказывающую существенное влияние на ИКТ в образовании.

Решение вопросов информационного пространства образовательного учреждения. Как показал опыт внедрения, для решения задач полноценной организации сетевого информационного пространства ОУ, его взаимосвязи с другими образовательными учреждениями необходимо создание принципиально новых программных продуктов, основанных на свободных решениях и открытых стандартах, не имеющих каких бы то ни было ограничений срока использования. На сегодняшний день задача организации информационного пространства ОУ решается путем внедрения разнородных программных продуктов, работающих под управлением семейства операционных систем MS Windows, что требует значительных затрат на приобретение лицензий как на сами программные продукты, так и на технологическую платформу, что существенно сдерживает массовое внедрение современных технологий в ОУ.

Вопросы поставок средств вычислительной техники в ОУ. Общеупотребительной на сегодняшний день является поставка в ОУ компьютеров с уже предустановленными проприетарными программными продуктами, чаще всего с семейством операционных си-

стем Windows. Необходимо обеспечить поставку оборудования с уже предустановленным ПСПО (как самостоятельно, так и в качестве второй операционной системы), а также обеспечить наличие драйверов устройств для ПСПО в поставлемом периферийном оборудовании. Данное требование должно стать обязательным при проведении конкурсов и тендеров на поставки средств вычислительной техники в ОУ.

Необходимость привлечения учреждений высшей школы. В процессе работы по реализации проекта ПСПО отмечен существенный интерес со стороны ВУЗов к вопросам использования СПО в высшей школе. Работа по взаимодействию с ВУЗами является весьма существенной, она позволит не только перейти к массовому использованию СПО в высшей школе, но и решить проблемы, связанные с подготовкой квалифицированных кадров, апробацией новых методик. Во многих случаях, как показывает мировой опыт, студенты в состоянии создавать весьма перспективные разработки как в рамках учебного процесса, так и при самостоятельной работе.

Необходимость создания специализированных дистрибутивов. Опыт проекта показывает, что на базе свободного ПО можно создавать в короткие сроки не только дистрибутивы общего назначения, но и специализированные программные продукты, охватывающие очень широкий спектр применения — от специализированных решений для лиц с ограниченными способностями до углубленного преподавания различных дисциплин.

Необходимость концептуального подхода. Использование СПО в образовании открывает новые возможности по обеспечению ОУ современным, качественным и доступным ПО. Однако данный проект требует комплексного подхода к решению разнородных задач, скоординированных усилий представителей различных ведомств. Для успешного решения данной проблемы необходима разработка «Концепции разработки и использования СПО в российском образовании», описывающей полно и комплексно цели и задачи проекта и пути их реализации.

А. Г. Кушниренко, А. Г. Леонов, А. В. Карпов, М. А. Ройтберг, Н. М. Субоч, Д. В. Хачко, В. В. Яковлев Москва, Пущино,
Научно-исследовательский институт системных исследований РАН,
Институт математических проблем биологии РАН,
Пущинский госуниверситет

Проект: Кумир `http://www.infomir.ru`, `http://lpm.org.ru/kumir/`

КуМир вернулся: обучение основам программирования с помощью системы КуМир

Аннотация

КуМир (Комплект учебных Миров) — 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 простой алголоподный язык с русской лексикой и встроенными командами управления программными исполнителями (Робот, Чертежник).

В докладе представлена новая версия системы КуМир для Windows, Linux, а также других распространенных Unix-вариантов (например FreeBSD).

КуМир — практикум по основам алгоритмизации — хорошо известен в современной школе благодаря тому, что целое семейство практикумов КуМир в 90-х годах прошлого столетия было спроектировано и реализовано практически на всем «зоопарке» вычислительной техники, которая использовалась в СССР (потом России) и странах СНГ. Такой подход, гарантирующий определенную мобильность разрабатываемого программного обеспечения, требовал специальной дисциплины — программирования. Действительно, ПЭВМ «АГАТ» на процессоре 6502, Zilog-ориентированные машины (в том числе и СР/М-машины на процессоре Intel8080), серия мини и микрокомпьютеров на процессорах фирмы DEC, IBM PC-совместимая техника, — это далеко не полный список вычислительной техники, на которой функционировал практикум КуМир.

На всем ряду разнообразной вычислительной техники КуМир предоставлял идентичный интерфейс, так как основная задача КуМира — сопровождение школьных (и ВУЗовских) учебников по ин-

форматике. В основе системы КуМир лежал школьный алгоритмический язык, предложенный академиком А. П. Ершовым, содержащий русскоязычные основные синтаксические конструкции алгоритмических языков высокого уровня.

Распространение PC-подобных клонов в конце XX века вытеснило с рынка или свело к минимуму практически все не Windows-ориентированные системы. Разработанная в 1993–96 годах версия КуМир была практически лишена программных ошибок и так близка к идеалу школьного ПО, что просуществовала в школьных классах достаточно долго (более 10 лет) для короткой жизни ПО. Однако DOS-ориентированный интерфейс потребовал (даже в дань моде) провести некоторые, казалось бы, косметические изменения в интерфейсе системы КуМир. Но на этапе обсуждения было принято решение о начале новой мультиплатформенной разработки системы КуМир, чтобы не только внести изменения в интерфейс, но и упростить механизм поддержки мультиплатформенности (Linux, Windows, MacOS X, и т. д.).

К 2005 году в НИИСИ РАН по заказу Российской Академии Наук был начат проект по созданию новой системы КуМир. Система Кумир разработана и распространяется свободно на условиях лицензии GNU GPL 2.0. Данная лицензия разрешает бессрочно использовать КуМир на любом количестве компьютеров в любых целях без оформления каких-либо дополнительных документов.

В системе КуМир используется школьный алгоритмический язык с русской лексикой и встроенными исполнителями Робот и Чертежник.

При вводе программы КуМир осуществляет постоянный полный контроль ее правильности, сообщая на полях программы обо всех обнаруженных ошибках.

При выполнении программы в пошаговом режиме КуМир выводит на поля результаты операций присваивания и значения логических выражений. Это позволяет ускорить процесс освоения азов программирования.

Также мультиплатформенная учебная система КуМир предоставляет учителю возможность использовать тот же язык программирования для подготовки задач, который используется для обучения школьников.

Кроме того, аппарат программных исполнителей можно использовать для создания обстановки, задания данных, а также, напри-

мер, для проверки правильности программы ученика. Так, в КуМире имеется встроенный исполнитель с фиксированными набором предписаний-программ, одна из которых исполняется перед выполнением программы ученика, а другая после ее окончания. Таким образом, педагог может задать различные данные для групп учащихся или индивидуально для каждого ученика, при этом обучающийся не только не сможет изменить заданные данные, но и даже увидеть их. Чтобы эффективно использовать время урока, по окончанию выполнения программы, можно автоматически проверить результат работы ученика, если написать на КуМире программу проверки и добавить ее к заданию.

В предыдущем поколении системы КуМир эта функциональность инициировалась учеником по нажатию `Ctrl+T` (тест, проверка). В настоящей версии такая проверка может проходить автоматически по окончанию выполнения программы. При этом для верификации результатов работы ученика можно использовать все функциональное богатство системы КуМир, создав программу проверки непосредственно в задании, или, например, переслать результаты работы ученика, включая программу и результаты, по сети учителю или внешней системе для оценки.

Система КуМир позволяет не только использовать готовые исполнители и обучаться программированию, составляя алгоритмы управления ими, но и создавать в программе новые внутренние исполнители, имеющие свои локальные и глобальные величины и предоставляющие доступ к алгоритмам исполнителя другим исполнителям.

Авторы благодарят Я. Н. Зайдельмана за многочисленные обсуждения, полезные советы и апрбацию предварительных версий системы КуМир.

Литература

[1] *Кушниренко А. Г., Лебедев Г. В., Зайдельман Я. Н.* Информатика. 7–9 класс: Учебник для общеобразовательных учреждений. — 3-е изд., стереотип. — М.: Дрофа, 2002.

Н. Н. Непейвода Ижевск, Удмуртский Государственный Университет

Проблемы открытого софта для Pitecantropus informaticus и Australopitecus informaticus

В последние два года многие ВУЗы России столкнулись с двумя проблемами: падение на 70% уровня подготовки абитуриентов и появление на 1-ых курсах поколения, которое обычно называется Homo informaticus. Анализируя встречающиеся в литературе характеристики Homo informaticus и сравнивая их с реальным уровнем абитуриентов даже на специальности «Информатика» в отнюдь не худшем российском университете, можно отметить следующее:

- До уровня Homo эти особи не доросли. Они работают с компьютерами, как обезьяны. Мышление их убито, они пытаются угадать желаемый ответ, как и требуется в тестах.
- Столкновение с открытым софтом вызывает у них шок. Командная строка воспринимается как изощренная пытка: «Ты не умствуй, ты пальцем покажи, куда ткнуть мышой!»
- Полное неумение читать и клиповое мышление. Ввиду привычки к «копипасту» и «языку падонков», они часто даже не могут перепечатать без ошибок в командную строку строку из учебного пособия, а тем более модифицировать ее другими параметрами.
- Неумение писать.
- Неумение искать релевантную информацию.

Не учитывая все эти особенности, преподавать невозможно. Лишь 1-ый курс — то время, когда «обезьянолюдей» можно превратить хотя бы в неандертальцев. Поэтому открытый софт и только открытый софт с самого первого дня — решение в данной критической ситуации.

Но преподавателям необходимо перестроиться психологически и понять, что «обезьянье» поведение — не столько вина этих молодых ребят, сколько их беда. Практически весь первый семестр уходит на то, чтобы в глазах появились искорки разума, а в решениях — проблески осмысленности.

И. А. Хахаев Санкт-Петербург, ГОУ ВПО СПбТЭИ

Свободное ПО и лицензирование ВУЗа

Аннотация

Обсуждаются результаты использования программных средств для сбора данных о показателях деятельности ВУЗа и проверки знаний студентов в режимах on-line и off-line, распространяемых Национальным аккредитационным агентством, в среде СПО ALT Linux 4.1.

Каждые 5 лет каждый ВУЗ (а также каждая школа и каждое среднее специальное учебное заведение) проходит процедуру лицензирования и аккредитации для получения лицензии и свидетельства о Государственной аккредитации, позволяющих выдавать выпускникам дипломы государственного образца.

Для сопровождения процедуры лицензирования используются бесплатные программные средства, распространяемые через сайт Национального аккредитационного агентства (`http://www.nica.ru/`), рассчитанные на работу в среде MS Windows. Для вузов имеется следующий набор программ:

1. Модуль комплексной оценки;
2. Модуль образовательных программ ВУЗа;
3. Модуль сбора данных о дипломах;
4. Программа анкетирования студентов по внеучебной деятельности;
5. Система тестирования базовых или остаточных знаний студентов.

Проверялась работоспособность этих программ в среде ALT Linux 4.1 (branch) с использованием входящей в него открытой версии WINE. Выяснилось следующее:

1. Модули *«комплексной оценки»*, *«образовательных программ»* и *«сбора данных о дипломах»* являются локальными и работают корректно после инсталляции под WINE.
2. *Программа анкетирования студентов* по внеучебной деятельности состоит из собственно модуля анкетирования и модуля

сбора результатов. *Модуль сбора результатов* является локальным и после инсталляции под WINE работает корректно. *Модуль анкетирования* является «сетевым» (штатный режим запуска — с «сетевого диска» MS Windows); он не требует инсталляции, но при попытке запуска в WINE выдается сообщение об ошибке подключения к базе данных (базе вопросов анкеты). Эта ошибка наблюдается как при запуске программы с сетевого ресурса (samba), смонтированного в локальный пользовательский каталог и сконфигурированного как «сетевой диск» в WINE, так и при локальном запуске программы из `wine_c`.

3. *Система тестирования* знаний студентов использует тесты Федерального экзамена профессионального образования (ФЭПО — `http://www.fepo.ru/`, `http://att.nica.ru/`) и может использоваться в режимах on-line и off-line. В режиме on-line требуется только браузер (по соответствующим «Техническим требованиям» нужен Internet Explorer, однако Firefox и SeaMonkey вполне подходят). При тестировании в режиме off-line текущая версия модуля тестирования должна запускаться с «сетевого диска» Windows. В таком варианте при запуске в WINE не подключается база вопросов без сообщений об ошибке. Эффект отсутствия подключения к базе вопросов имеет место как при штатном запуске модуля тестирования (с «сетевого диска», смонтированного в локальный каталог и сконфигурированного как логический диск в WINE), так и при запуске с `wine_c`.

4. Программа анкетирования и модуль тестирования без всяких проблем запускаются и корректно работают, используя в качестве «сетевого диска» Windows ресурс samba.

Нужно отметить, что при подготовке к аккредитационному тестированию большую роль сыграла LMS Moodle, в которую занесены вопросы, максимально близкие к вопросам ФЭПО (сайт ГОУ ВПО СПбТЭИ с Moodle — `http://msdl.spbtei.ru/msdl`). Очень полезны оказались возможности анализа ответов на вопросы тестов и выявления тем самым проблемных дидактических единиц (при большом количестве вопросов в тесте и большом количестве попыток сервер Moodle перестаёт отдавать результаты).

Выводы

- Локальные версии программ для обеспечения процедуры лицензирования могут запускаться в открытой версии WINE без ущерба для функциональности;
- Сетевые версии (в случае анкетирования и тестирования знаний в режиме off-line) не работают должным образом в открытой версии WINE;
- Тестирование знаний студентов в режиме on-line на сайтах `http://www.fepo.ru/` (добровольное) и `http://att.nica.ru/` (аккредитационное) возможно при использовании браузеров Mozilla Firefox и SeaMonkey, работающих в ALT Linux;
- Полностью обеспечить процедуру лицензирования и аккредитации с помощью открытых и свободных программных средств пока не удаётся. В любом случае требуется либо MS Windows (с полной гарантией работоспособности анкетирования и off-line-тестирования) или коммерческая версия WINE (с негарантированной работоспособностью анкетирования и off-line тестирования).

Е. Д. Патаракин
Переславль-Залесский,
Институт Программных Систем
Проект: Летописи.ру
http://letopisi.ru

Развитие сообщества языка Scratch в России

Развитие языка Scratch в России тесно связано с развитием Викисообщества Летописи. Истоки образовательного использования вики и Scratch тесно связаны с конструкционизмом и отправляют нас на 40 лет назад, когда Сеймур Пейперт и его коллеги создали первый язык для обучения детей математике. Эта мощная педагогическая идея была оттеснена в сторону первой волной развития средств мультимедиа. Развитие сети Интернет и Всемирной Паутины способствовало пассивному использованию компьютеров, поскольку основная активность за компьютером оказалось смотровой — просмотр сайтов и

презентаций. В скором времени в головах учителей возникла склейка, что компьютерные технологии в образовании и PowerPoint — это одно и то же. В начале 21 века эта склейка распространилась повсеместно, и если в педагогической среде произносилось слово «гипертекст», то имелась в виду презентация. Изменение ситуации связано с распространением вики-культуры, в которой принята радикальная модель коллективного гипертекста, когда возможность создания и редактирования любой записи предоставлена каждому из членов сетевого сообщества.

Мы пошли по пути создания одного крупного образовательного проекта, и в феврале 2006 года появился проект Летописи (`http://letopisi.ru`). Сегодня, когда со старта проекта прошло три года, мы можем использовать наработанный материал для построения на его фундаменте учебных проектов нового сетевого типа. В сообществе Летописи мы очень любим метафору кирпичиков. Мы можем собирать статьи из готовых блоков так же, как собирается программа из кирпичиков Лего. Кирпичики Лего или статьи Wiki полезны и сильны не тем, что они просты, а тем, что для них определен точный стандарт и они всегда совместимы друг с другом. Культура сетевого соучастия делает свои первые шаги, и внутри нее пока обмениваются простыми объектами: текстами, рисунками, фотографиями, закладками. Это пока стадия цифрового детского сада. Для того чтобы успешно развиваться и подниматься на новый уровень творчества, нам необходимы более сложные учебные объекты, которые мы могли бы придумывать, создавать и которыми мы могли бы обмениваться. По мере развития технологий в сферу построения значимых продуктов попадают все новые маленькие кирпичики — цифровые учебные объекты, пригодные для повторного использования в образовательных целях. Это общая игра по построению новых знаний, в которую с одинаковым интересом могут играть и школьники, студенты, учителя, и преподаватели. Чем большие возможности открывает учебная среда для самостоятельного построения, конструирования новых объектов, тем с большим интересом к ней относятся пользователи. Конструктивизм заметно повлиял на педагогический дизайн и воплотился в следующих педагогических средах: Лого и его производные NetLogo, StarLogo, Сквик и его производные Scratch, Sophie, Alice. Из перечисленных наибольшее распространение в России получил Scratch. С весны 2008 года сайт `http://scratch.mit.edu/` поддерживает русский язык, а в сентябре 2008 года мы подготовили полностью русифицированную

версию Scratch 1.3. В этой простой среде уже видны мощные возможности параллельных действий множества исполнителей. Scratch позволяет детям создавать собственные анимированные и интерактивные истории, игры и другие произведения. К каждому создаваемому объекту можно записать несколько блоков команд, и все блоки будут выполняться параллельно. Действующих объектов может быть сколько угодно. В результате выполнения простых команд может складываться сложная модель, в которой будут взаимодействовать множество объектов, наделенных различными свойствами. Изучение поведения и построение моделей более не ограничивается отдельным компьютером. Мы можем предложить свое произведение другим членам сообщества, и они смогут не только посмотреть, как работает модель, но и разобраться в том, как сложены кирпичики модели, смогут взять эти кирпичики и строительные блоки и построить из них свое собственное здание. Можно не только посмотреть готовые работы, но и скачать их код. После этого любой желающий может видоизменить связь блоков программы и построить свою собственную. Цель, которую ставят перед собой создатели Скретч, — построение творческого сообщества. Участвуя в развитии сообщества Скретч, мы стремимся наполнить эту среду объектами, которые были знакомы российским школьникам. И здесь происходит крайне полезное одомашнивание и освоение цифровых коллекций педагогических университетов. Например, мы использовали для объектов и сцен коллекции астрономического музея (НКЛФА) Нижегородского педагогического университета. Множество растений и животных попало в российские Скретч-библиотеки из определителей и цифровых Красных книг, созданных нижегородскими экологами. Огромное спасибо Сергею Шустову, который передал множество своих работ в открытый доступ с правом дальнейшего видоизменения в детских проектах. Это возможность не просто посмотреть, прочитать, но и поиграть с объектами, сделать с ними свой собственный проект. В настоящее время на базе Летописи.ру собрано множество иллюстрированных статей с учебными материалами и примерами проектов на языке Scratch. Эти материалы доступны по адресу `http://letopisi.ru/index.php/Scratch`.

Ю. В. Катков, Б. Б. Ярмахов Санкт-Петергбург, Н. Новгород,
Open Source Linux Lab СПб ГЭТУ «ЛЭТИ», Медиалаборатория НГПУ

Проект: OLPC `http://osll.spb.ru/projects/show/xosupport`

Свободное ПО в проекте OLPC в России: результаты и перспективы

Аннотация

One Laptop Per Child — некоммерческая американская организация, собравшая вокруг себя огромное сообщество волонтеров в большинстве стран мира. Главный продукт OLPC — детские ноутбуки XO и периферия для них. В докладе произведен разбор некоторых особенностей программного обеспечения ноутбука, рассмотрены существующие проекты российских волонтеров OLPC, а также определены студенческие проекты, посвященные XO-1.

Как уже рассказывалось на этой конференции [3] , технологические инновации в ноутбуке XO как нельзя лучше подходят для реализации модели 1:1, в которой компьютер является не предметом изучения, а инструментом для познания мира. Это обеспечивается как нестандартностью аппаратных решений, так и инновационным подходом к разработки ПО для него. Его ключевой особенностью является открытость исходного кода всех компонентов — от BIOS (Open Firmware) до пользовательских приложений-активностей. Каждое приложение Sugar может использоваться детьми совместно, как общая доска для рисования. Лучше всего этот принцип демонстрируется средами для набора текста, создания музыки и графики. Интерфейс Sugar работает на иных, по сравнению с традиционными, построенными на метафорах файла, папки и рабочего стола, принципах работы с файловой системой. Доступ к ФС осуществляется с помощью Активности «Журнал». Факт запуска каждой запущенной программы записывается в журнал, туда же охраняются созданные программами документы, скриншоты и пр. Отказ от иерархического строения файловой системы объясняется в [2].

Все эти свойства ноутбука OLPC XO делают его исключительно привлекательным для использования в образовательной практике, что и было проверено сообществом OLPC-Россия летом 2008 года в рамках пилотного проекта.

Пилотный проект

Возможность внедрения XO в школьный учебный процесс была исследована при проведении детского образовательного лагеря «Цифровая экология 2008». Организованный сотрудниками Медиалаборатории НГПУ при технической поддержке OSLL «ЛЭТИ», лагерь продемонстрировал жизнеспособность модели «1 ученик : 1 компьютер», показал пригодность OLPC XO при работе с детьми среднего школьного возраста, что может быть использовано в учебном процессе в средней школе. В ходе 10 дней работы лагеря с пятьюдесятью ноутбуками XO, были сделаны существенные наблюдения за особенностями использования ноутбуков OLPC XO в обучении. Нами были сделаны следующие выводы:

- Использование ультрапортативных ноутбуков в детском летнем лагере позволяет создать качественно новую модель учебного процесса (цифровой лагерь), которая строится в русле движения «один ученик: один компьютер» и принципов образовательного конструкционизма.
- Обязательный набор оборудования для цифрового лагеря включает в себя: по одному ультрапортативному ноутбуку для каждого ребёнка, серверный компьютер для разворачивания беспроводной локальной сети, wifi-маршрутизатор. Для подключения к Интернету вполне достаточно одного GPRS-модема. Все остальные технические средства не более чем дополнения и могут варьироваться в зависимости от рабочих и обучающих активностей в лагере.
- Изучение основных операций работы с компьютером, даже в случае нового компьютерного интерфейса, построенного на отличных от традиционных парадигмах (в нашем случае это был Sugar от OLPC), происходит крайне быстро и занимает у детей 11—12-летнего возраста не более 2—3 дней.
- В качестве универсальной среды для «сборки» объектов цифровых коллекций, собираемых школьниками, может использоваться язык Scratch.
- В условиях летнего лагеря можно создать ситуацию безопасного использования дорогостоящей компьютерной техники.
- Централизация учащихся вокруг общего ресурса позволяет им эффективно взаимодействовать как с преподавателями, так и

между собой. В случае «ЦЭ08» таким ресурсом стала среда MediaWiki, доступ к которой имел каждый участник лагеря.

- Оптимальная форма работы летнего цифрового лагеря — разумное сочетание мероприятий, проводимых на открытом пространстве (походов в лес, лодочных экскурсий, спортивных мероприятий, вечеров у костра), и занятий, предполагающих использование цифровых технологий в лабораториях-пещерах.

Образовательный лагерь позволил сформулировать список задач, связанных с XO. На данный момент решаемые в России задачи можно разделить на следующие категории:

- портирование приложений на Sugar;
- организация летних учебных лагерей;
- освоение программных средств ноутбука и сервера XS;
- интернационализация программ ноутбука, Sugar и вики-портала OLPC;
- работа над протоколом mesh-сетей 802.11s, создание собственных прошивок:
- поддержка работы с ноутбуков в режиме графического планшета.

Портирование приложений на Sugar

«Активности» Sugar должны отвечать определённому формату, поэтому приложения на Sugar необходимо портировать. Портирование заключается в переписывании элементов управления на PyGtk/PyQt, настройке DBUS Tubes для совместной работы, настройке регистрации Активности в Журнале, а также сборке xo-bundle.

Сейчас на XO портируется система программирования КуМир и редактор интеллект-карт Vym. Необходимость для российских школьников наличия этих программ объясняется просто: для КуМир давно создана мощная методическая база. То же касается mindmap-редакторов (например Vym) — исследования в этой области ведутся достаточно давно, а интеллект-карты давно зарекомендовали себя в инженерии знаний и в педагогике [1]. Задачи портирования программ на Sugar и интернационализации являются, пожалуй, наиболее при-

ориетными, поскольку позволяют интегрировать ноутбук в систему российского школьного образования.

Mesh

Другой важной задачей является развитие технологии mesh-сетей [4]. Две антенны ноутбука служат для поддержки сетевой работы. Mesh — это протокол передачи данных 802.11s. Сейчас эта версия протокола находится на стадии разработки. Особенностью является самоорганизация таких сетей и отсутствие необходимости в точке доступа. В СПбГЭТУ «ЛЭТИ» сейчас проходит студенческий проект установки связи в гетерогенной mesh-сети между Nokia N8x0 и OLPC XO. Также для студентов есть возможность поучаствовать в проекте моделирования протокола mesh для симулятора NS. Моделирование позволит выявить ограничения протокола и, в конечном итоге, повлиять на конечную версию стандарта от IEEE.

Touch Pad

Неудобство тачпадов для проведения мелких операций вроде рисования или создания музыки в нотном редакторе. Тачпад ноутбука поддерживает режим работы в виде графического планшета (Tablet mode). На данный момент поддержка планшета отсутствует в ядре, но на этот режим есть возможность посмотреть в тестирующей программе. Добавив приложениям Sugar поддержку планшета, можно добиться более удобной работы с ноутбуком.

Перспективы

В самой организации OLPC на данный момент работает 32 человека, при этом сообщество свободных разработчиков, локализаторов, педагогов, участвующих в проекте на добровольной основе, исчисляется тысячами. На территории стран СНГ разработкой ПО и образовательных моделей и решений для OLPC занимаются лаборатория открытого ПО при СПб ГЭТУ «ЛЭТИ» (`http://osll.spb.ru/`) и Медиалаборатория при НГПУ (`http://www.nnspu.ru/`). В ближайшее время в работу включится РГПУ им. Герцена.

На сегодняшний момент пользователями OLPC XO в мире являются около 600 000 детей. Силами организации OLPC и волонтерского

сообщества ведется доработка и усовершенствование как аппаратной части (ведется работа над прототипом XO-2), так и программного обеспечения к нему. Кроме этого, процессы, инициированные OLPC, послужили толчком к массовому производству субноутбуков и нетбуков (на настоящий момент произведено около 17 млн. нетбуков разных моделей), которые могут быть использованы для реализации модели «1 ученик: 1 компьютер».

Литература

[1] Муромцев Д.И. Гаврилова Т.А. *Интеллектуальные технологии в менеджменте*. Высшая школа менеджмента СПбГУ, 2007.

[2] Андрей Письменный. *Почему разработчики OLPC обошлись без файлового менеджера*. Компьютерра Online, 2008.

[3] Ярмахов Б.Б. *"OLPC как модель массового внедрения свободного ПО в образование"*. 2008.

[4] Joseph D. Camp and Edward W. Knightly. *The IEEE 802.11s Extended Service Set Mesh Networking Standard*. 2006.

А. В. Хорошилов
Москва,
Институт системного программирования РАН

Практикум по аналитической верификации программного обеспечения

Аннотация

В докладе представлен опыт проведения практикума по аналитической верификации программного обеспечения для студентов высших учебных заведений на основе свободного программного обеспечения. На практических занятиях изучались классические методы верификации Флойда/Хоара применительно к последовательным алгоритмам на языке программирования Си и автоматизированное доказательство теорем, сгенерированных на основе этих методов. Практикум проводится для студентов факультета ВМиК Московского Государственного Университета имени М. В. Ломоносова в рамках курса "Методы спецификации и верификации программ".

Свободное программное обеспечение представляет собой прекрасную основу для проведения практических занятий по многим областям знаний и, в особенности, программной инженерии и компьютерным наукам. В настоящем докладе рассматривается опыт организации практикума для курса «Методы спецификации и верификации программ», который проводится на факультете ВМиК Московского Государственного Университета имени М. В. Ломоносова под руководством д. ф.-м. н. А. К. Петренко.

Курсы по методам спецификации и верификации программного обеспечения представлены во многих университетах России и зарубежных стран [1]. В основном, эти курсы либо рассматривают только теоретические аспекты верификации программного обеспечения, либо ограничиваются изучением специализированных инструментов и методов, которые не так просто связать с опытом традиционной разработки программ, который уже сформирован у студентов высших учебных заведений.

В рамках курса «Методы спецификации и верификации программ» на факультете ВМиК сделана попытка построить мостик между классической теорией верификации последовательных программ методами Флойда/Хоара [2][3] и опытом студентов по разработке последовательных алгоритмов на языке программирования Си.

Реализация данной идеи основана на использовании семейства инструментов верификации why [4] и системы автоматизированного доказательства PVS [5], распространяемых под лицензией GPL.

Рассмотрим один из основных видов задач, решаемых при помощи этих инструментов и служащих цели построения вышеуказанного мостика. Студент получает задание разработать реализацию несложного последовательного алгоритма в виде функции на языке программирования Си. Затем ему требуется формально описать требования к данной функции при помощи предусловия и постусловия, оформленных в виде обычных комментариев языка Си, но в соответствии с синтаксисом языка ACSL (ANSI/ISO C Specification Language) [6]. На следующем шаге к каждому циклу программы приписывается инвариант и вариант цикла, как этого требуют методы Флойда/Хоара. Из получившегося кода на языке Си с комментариями ACSL инструмент why генерирует набор лемм, из доказательства истинности которых следует полная корректность данного алгоритма относительно данной спецификации. Инструмент why поддерживает генерацию лемм в различных нотациях и, в частности, в нотации PVS Definition Language. Поэтому доказательство лемм в PVS является естественным путем для завершения аналитической верификации алгоритма. Пример простейшей спецификации на Си с комментариями ACSL представлен ниже.

```
//@ logic int fact(int n)
//@ axiom fact1_def: fact(1) == 1
//@ axiom fact_def: \forall int n; (n > 1) => fact(n) == n*fact(n-1)
/*@ requires  n > 0
  @ ensures  \result == fact(n)
  @
 */
int factorial(int n)
{
  int r = n;
/*@ label TOP */
/*@ invariant  (n > 0) &&
        (r == fact(\at(n,TOP))/fact(n-1))
     variant n
 */
  while(n>1)
    r = r*(--n);
```

```
  return r;
}
```

Опыт проведения практикума показал, что используемая связка инструментов предоставляет студентам возможность увидеть один из сценариев применения методов верификации в уже привычном контексте и осознать ряд сложностей, возникающих на этом пути. В результате, у студентов складывается более глубокое понимание материала по сравнению с вариантом проведения практики исключительно на специализированных языках спецификации.

Для проведения практикума на основе операционной системы Ubuntu 8.04 был подготовлен LiveDVD с уже установленным программным обеспечением, необходимым для выполнения заданий. При этом рекомендуемым способом использования диска являлась установка операционной системы в виртуальную машину VirtualBox [7], и большинство студентов избрали именно этот вариант, так как при достаточной производительности аппаратного обеспечения он является наиболее удобным. Во-первых, в этом случае сокращается время запуска системы, а во-вторых, остаются доступными все возможности и документы привычного рабочего окружения.

В заключении отметим, что связка свободного программного обеспечения why и pvs показала себя эффективным инструментом для изучения классических методов верификации Флойда/Хоара, и мы рекомендуем ее для использования в учебном процессе. Основным ее недостатком на настоящий момент является отсутствие более-менее полной документации на русском языке, но работа в этом направлении ведется в рамках подготовки методических материалов по проведению представленного практикума.

Также хотелось бы отметить, что использование готовых образов свободных операционных систем в совокупности с технологиями LiveDVD и виртуальных машин является очень удобным механизмом для проведения практических занятий, так как позволяет студентам и преподавателям иметь под рукой готовый инструментарий, не ограниченный учебным классом образовательного учреждения.

Литература

[1] *dddds.*
http://www.cs.indiana.edu/formal-methods-education/courses/.

[2] *Floyd R. W.* Assigning meanings to programs. — Vol. 19. — Providence: American Mathematical Society, 1967. — Pp. 19–32.

[3] *Hoare C. A. R.* An axiomatic basis for computer programming. — 1969.

[4] Семейство инструментов верификации why. `http://why.lri.fr/`.

[5] Система автоматизированного доказательства pvs. `http://pvs.csl.sri.com/`.

[6] Acsl: ansi/iso c specification language. `http://frama-c.cea.fr/download/acsl{\_}1.4.pdf`.

[7] Виртуальная машина virtualbox. `http://www.virtualbox.org`.

С. В. Знаменский Переславль Залесский, НОУ Институт программных систем — «Университет города Переславля»

Проект: КАИС «Ботик» `http://wiki.botik.ru/IS4UGP`

Контекстно-автономная информационная система

Аннотация

Описывается контекстно-автономная информационная система[1], в которой структуры и взаимосвязи данных и организация доступа к ним могут быть произвольно изменены в любом контексте без последствий для остальной части системы.

Система реализуется отрыто на свободном программном обеспечении.

Проблема организации сотрудничества в новаторской образовательной деятельности

Содержание образования непрерывно пересматривается, цели уточняются, возникают принципиально новые технические средства, и поэтому оптимизация совместного управления такой деятельностью является алгоритмически неразрешимой проблемой, требующей творческого поиска.

[1]Проект заявлен в РФФИ под номером 09-07-00470-а.

При этом любая информация об успешных или не вполне успешных педагогических экспериментах может в какой-то момент оказаться исключительно ценной.

Полноценная информационная поддержка такой деятельности должна изменять структуру внутренних данных намного быстрее, чем сами данные теряют актуальность. То есть в системе должны соседствовать различные модификации структур данных, а конфликты и противоречия должны разрешаться по мере обнаружения.

Базовая функциональность и основные структуры и взаимосвязи (онтология) этой информационной системы не могут закладываться на этапе ее создания. Система должна гибко эволюционировать, а не устаревать, и заменяться обновленной версией.

Принципы построения системы «Ботик»

Мы строим систему на следующих принципах:

Контекстная автономность: любой организационный контекст (т. е. вся организация, любое подразделение, комиссия, совет, рабочая группа, проект, конкретное дело или мероприятие, или его часть, требующая ролевого взаимодействия различных сотрудников) допускает произвольные изменения структуры и обрабатывающего программного кода. Побочные эффекты на результатах обработки запросов, не связанных с этим контекстом, недопустимы.

Полная ретроспективность: любой запрос на чтение может быть дан с дополнительным указанием (прошедшего) момента времени, и ответ на него должен быть ровно таким, каким он был бы на указанный момент. Разумеется, в контекстах с ограниченным доступом ретроспективный запрос также ограничивается в соответствии с сегодняшним статусом пользователя. Более того, все изменения в любом контексте до указанного момента можно отменить (и, соответственно, вернуть, отменив изменения за время отмены).

Требование полной ретроспективности облегчает поддержание системы в рабочем состоянии.

Выбор платформы

Проект в духе Web 2.0 и [1] предусматривает развитие системы Nadmin с приданием ей следующих черт:

Персонализация бизнес-процесса: более важные текущие дела каждого сотрудника, вплоть до участия в открытых обсуждениях, должны быть более на виду. Индикация дел должна немедленно отражать изменения приоритетов руководства, отмены заданий, завершение работ, изменение состава рабочей группы или роли в ней пользователя и изменение приоритетов конкретного контекста пользователем.

Стандарты качества ISO 9000-9001 и ГОСТ Р ИСО 15489: ролевое обсуждение и корректный учет всех мнений при принятии решений. Любому решению предшествует протоколируемый обмен открытыми и порой конфиденциальными сообщениями в ходе ролевой дискуссии и прозрачное автоматическое сведение всех выраженных оценок в новый статус документа.

Трекинг вклада каждого сотрудника: общая динамика активности и результативности участников, а также точный учет и формальная оценка, кто и насколько задерживает обсуждение и другие дела.

Сложность задачи делает крайне проблематичной масштабируемость до межвузовских ресурсов при опоре на традиционные SQL/XML/RDF технологии. Придерживаясь принципа открытости разработки, мы сегодня опираемся на связку Apache2 + mod-perl2 + BerkeleyDB + jQuery.

Практическая реализация

Разработка системы ведется открыто на `http://wiki.botik.ru/IS4UGP` силами студентов 2–4 курсов при поддержке *ООО «Ботик-технологии»*. Руководят разработкой студентки 4 курса УГП Лена Титова и Надя Живчикова, прошедшие *зимнюю школу Linux для преподавателей* и уже разработавшие прошедшим летом первое контекстно-автономное хранилище структурированной информации [2]. Недавно в постоянную эксплуатацию введен модуль трекинга учебной и производственной практик и курсовых и дипломных проектов с процедурами утверждения/изменения темы (организации),

выбора/смены руководителя и других параметров, а также контроля регулярности и содержания периодических отчетов о проделанной работе.

С февраля на контекстно-автономное ядро `http://edu.botik.ru` перейдет и подсистема поддержки конференций [3], в частности, *Молодежная научная конференция «Наукоемкие информационные технологии» Переславль-Залесский, 22–25 апреля 2009 г.*, и юбилейная конференция ИПС РАН будут проходить с перекрестным рецензированием и согласованием мнений рецензентов.

Литература

[1] *Знаменский С. В.* Хорошо масштабируемое автономное администрирование доступа. Труды Международной конференции «Программные системы: теория и приложения», Переславль-Залесский, октябрь 2006, Наука, — Физматлит, М. Т. 1, с. 155–169. — `http://skif.pereslavl.ru/psi-info/psi/psi-publications/e-book-2006/index.html`

[2] *Живчикова Н. С., Титова Е. В.* Логическая модель изменчивых организационных структур. Тезисы международной конференции «Системы проектирования, технологической подготовки производства и управления этапами жизненного цикла промышленного продукта (CAD/CAM/PDM — 2008)»

[3] *Коряка Ф. А.* Автоматизированная система управления вузом — UPIS. XI научно-практическая конференция «Университета г. Переславля». Переславль-Залесский, апрель 2007, изд.-во «Университет города Переславля», Т. 1, с. 59–63. — `http://wiki.botik.ru/up/pub/IS4UGP/StudConf/1-2/03-koryaka-p-59.pd`

Д. В. Сподарец Одесса (Украина), UAFOSS, журнал RootUA

Всеукраинская инициатива использования свободного программного обеспечения в образовании и науке

Аннотация

Со свободным программным обеспечением представители образования и науки Украины знакомы уже более пяти лет. Знакомство это в разное время носило различный характер: от пассивно-информационного в начальной стадии до активно-делового в последние два года. Возможно, сплеск активности связан с известными лицензионными событиями в России и участившимися проверками у нас, в Украине, но как бы то ни было, многие поняли реальную пользу и перспективность перехода на СПО.

В 2007 году на конференции OSDN, в рамках секции «СПО в образовании и науке», впервые удалось собрать людей, занимающихся этим делом. А уже через год в Одессе на конференции FOSS Sea 2008 взяла старт «Всеукраинская инициатива использования свободного программного обеспечения в образовании и науке».

Инициаторами выступили Украинская Ассоциация пользователей и разработчиков свободного и открытого программного обеспечения (UAFOSS) и всеукраинский журнал RootUA.

Рабочей площадкой стал специально созданный сайт инициативы — `http://www.edu.root.ua/`. Все обсуждения проходят на форуме по адресу `http://forum.root.ua/viewforum.php?f=31`.

Linux уже прошёл первичную апробацию и используется в:

- Одессе: школа №84, школа №85, экономический лицей, Одесский национальный университет им. И. И. Мечникова (используется OpenOffice.Org, Firefox...);
- Полтаве: гимназия №17;
- Киеве: КПНЛ №145;
- Львове: школа №80, Львовский национальный университет им. Ивана Франко.

И это только начало и далеко не полный перечень тех учебных заведений, где используется СПО. Реально, по нашим предположениям, СПО используется чуть ли не во всех учебных и научных заведениях Украины, но некоторые из пользователей просто не знают о том.

Разработан план первоначальных действий, который сейчас находится в активной фазе исполнения.

Первый этап:

- Провести информирование учебных и научных заведений о старте инициативы и познакомить их со свободным программным обеспечением.
- Подготовить предложения для учебных и научно-исследовательских заведений по присоединению к инициативе; провести мониторинг заведений, которые уже используют открытое программное обеспечение.
- Подготовить пакет свободного программного обеспечения для ОС Windows (включая документацию к программам), чтобы разослать всем тем, кто присоединится к инициативе.
- Подготовить пакет информационных материалов для различного рода СМИ.
- Подготовить программный комплекс для оказания технической поддержки учреждениям, подключившимся к проекту.

На втором этапе будет проведен сбор данных, которые позволят сформулировать конкретные требования к будущей учебной программе и операционной системе на открытом коде. Будет проведён отбор учебных заведений для пилотного запуска созданной ОС.

Инициатива на марше: недавно создан портал `http://www.root.ua/`, призванный объединить украинское сообщество, направить его работу, организовать обмен опытом.

В становлении и развитии портала помогают два украинских издания, посвященные свободному программному обеспечени: всеукраинский журнал RootUA и газета FOSS News.

Инициативе способствует техническая мощь `http://www.OSDN.org.ua/`, обеспечивая интернет-пространство и все необходимые сервисы для работы.

Основным центром взаимодействия выступает UAFOSS. Задействовав ее сеть и подключив к ней новые инициативные группы, мы надеемся организовать мощную техническую поддержку на местах.

Для общения и обмена опытом в неформальном режиме на портале `http://www.root.ua/` была развернута социальная сеть `http://www.blog.root.ua/`.

Мы открыты для всех, кто готов присоединиться к доброму, нужному и перспективному делу. Ждем новых участников инициативы.

Будем рады интересным идеям и предложениям. До встречи на нашем форуме.

Э. В. Хайруллов Димитровград, Управление образования администрации города Димитровграда

Адаптивное восприятие и социальные аспекты внедрения ПСПО

Аннотация

Вот уже практически два года мы ведем политику активного внедрения и использования СПО в образовании; за период с 2007 года по нынешнее время были достигнуты значительные результаты в плане его продвижения; пакеты СПО есть в каждой школе, проводятся семинары, мастер-классы. Сейчас ни один из директоров школ не представляет возможность использования Интернета и внутришкольного контроля без интернет-сервера, работающего на СПО. Но активной пользовательской позиции все же не наблюдается. Поэтому в ноябре 2008 года нами было решено провести обширное социологическое исследование на предмет выявления сдерживающих факторов внедрения СПО.

Аудитория была разделена на несколько категорий

- Школьники младшего звена;
- Школьники среднего звена;
- Выпускники;
- Учителя информатики;
- Преподаватели ВУЗов;
- Бухгалтерия;
- Библиотекари;
- Администрация школы (завучи, директора);
- Администрация Управления образования.

Для каждой категории был составлен свой опрос и анкета. После этого были подведены результаты по каждой группе в отдельности. В итоге были сделаны выводы и определены направления для дальнейшей работы.

1. Разработка проекта по обеспечению доплаты работникам образовательных учреждений, использующим СПО, за счет стимулирующей части ФОТ в связи с введением отраслевой системы оплаты труда.
2. Адаптация комплекса бухгалтерских программ под СПО.
3. Введение координационной должности по обеспечению учителей-предметиков материалами и курсами по СПО.

В результате, формальным промежуточным итогом в каждой группе стал практически один и тот же вывод: «Пользователю не нужна операционная система как таковая, ему нужен адаптивный набор сервисов, позволяющий ему выполнять повседневные задачи. Таким образом, внедрение СПО заключается в адаптации пользовательских задач на новом рабочем месте».

С формами анкет и результатами опросов вы можете ознакомиться по адресу `http://heap.altlinux.org/engine/Emil_V.Hairullov`.

А. А. Панюкова Москва, ALT Linux
Проект: ALT Linux Children `http://www.altlinux.ru`

Дистрибутив ALT Linux Children: опыт и перспективы

Аннотация

Специализированный дистрибутив для детского творчества ALT Linux Children успешно развивается и применяется уже два года. В докладе рассматриваются технические вопросы формирования такого дистрибутива, обосновывается выбор ПО, описываются существующие варианты дистрибутива и перспективы его развития.

Историческая справка

Летом 2008 года была выпущена бета ALT Linux 4.0 Children в формате Live CD[2]. Live CD («Живой CD») — это возможность полноценной работы в Linux-окружении на любом компьютере без установки операционной системы на жесткий диск. Единственное требуемое действие — загрузка компьютера с самого Live CD. Большая часть файлов системы находится на диске, значительно меньшая — в

оперативной памяти, а сохранять результаты работы можно на флэш-носитель или дискету. Содержимое жесткого диска в используемом варианте Live CD доступно в качестве отдельных каталогов для того, чтобы можно было задействовать при работе имеющийся банк изображений.

В первую очередь, диск был выпущен для раздачи детям по окончанию занятий в детском оздоровительном лагере «Березка» и на базе отдыха «Наука» при Южно-Уральском государственном университете[4].

В декабре 2008 года была выпущена очередная бета ALT Linux Children на базе бранча 4.1. Основные отличия от дистрибутива на 4.0, помимо более современной системной части, — дополнен курс, обновлены исходные материалы для детей, несколько уточнена пакетная база. Кроме того, ALT Linux Children 4.1 напечатан большим тиражом по сравнению с 4.0.

ALT Linux 4.1 Children сегодня

Программное наполнение ALT Linux 4.1 Children составлялось с расчетом не только на непосредственное проведение курса, но и на последующую самостоятельную работу детей дома. В дополнение к программным продуктам, необходимым для самого курса, в дистрибутив входят инструменты для более требовательных или любознательных пользователей: диспетчер фотографий цифровой фотокамеры, редакторы фрактальной и ASCII графики, мощный аудиопроигрыватель, программы по астрономии и географии. По сравнению с предыдущей версией дистрибутива значительно обновлены графический редактор GIMP и редактор нелинейного видео Kdenlive.

Для дошкольников от 4 лет и младших школьников предусмотрен развивающий центр GCompris, который содержит множество модулей, начиная с освоения клавиатуры и мыши и заканчивая логическими играми и заданиями, помогающими изучать окружающий мир. Для цели общей тренировки добавлен клавиатурный тренажер.

Стоит отдельно отметить, что в состав дистрибутива не входят офисные программы, почтовые клиенты, средства манипуляции контактной информацией и прочие продукты, не отвечающие поставленным задачам. Практика преподавания показала, что эти программы создают значительный отвлекающий фон и засоряют информационное пространство при проведении занятий. По той же причине под-

ключение к сети Интернет переведено в «ручной» режим и по умолчанию не используется. Для многопрофильной работы на компьютере рекомендуется использовать дистрибутивы ALT Linux 4.1 с другим спектром решаемых задач.

Перспективы и пути развития

Live CD

По окончании работ по курсу для детей [3], планируется релиз Live CD на бранче 5.0 с выпуском книги по курсу.

Кроме того, рассматривается возможность выпуска Live DVD с более подробным набором исходных материалов (в том числе с видео), большим количеством игрушек и, конечно, возможностью настроить себе Интернет, об отсутствии которой очень сожалеют родители. Возможно наличие «профилей загрузки» на Live CD: сейчас по умолчанию после загрузки ребёнок получает систему, в которой все носители информации примонтированы на запись. Такая схема идеальна для того, чтобы ребёнок мог сохранять созданные файлы на любые носители, чтобы не потерять их к моменту завершения работы. Однако не все родители уверены в том, что имея доступ на запись ко всем жестким дискам, ребенок не повредит информацию. Поэтому рассматривается возможность создания нескольких вариантов загрузки:

- со всеми носителями, доступными на запись;
- с жесткими дисками, доступными только на чтение; при этом все съемные носители доступны на запись;
- только съемные носители доступны на запись.

Установочный вариант

Домашний для детей

Есть некий класс детей, которые утверждают, что компьютер в доме используется только ими, что они отдают отчет в своих действиях и т. п. Есть некий класс родителей, которые были бы не против, чтобы ALT Linux Children стоял на компьютере, не было бы необходимости постоянно грузиться с Live CD. Поскольку такие вопросы задаются довольно часто, логично предположить, что такой вариант дистрибутива тоже имеет место быть. С другой стороны, такое решение не

может быть универсальным, если компьютер в доме используется не только и не столько ребенком, сколько родителями. Ставить отдельный дистрибутив для ребенка довольно неудобно: разница между загрузкой с Live CD и установленной второй системой (особенно если один Linux уже стоит) с точки зрения удобства процесса работы невелика. Ставить для ребенка Children, а родителям потом доставлять для себя все недостающие программы тоже довольно неудобно. Доводить Desktop под пользователем ребенка до состояния, близкого к Children тоже можно, но по количеству усилий эквивалентно предыдущему варианту.

Очевидно, отдельный дистрибутив довольно неудобен с точки зрения конечного пользователя, как и любое размножение сущностей. В качестве варианта решения можно предложить вариант создания при установке более одного пользователя, но с разными профилями. Так, например, в случае Children, можно было бы по выбору устанавливающего создать дополнительного пользователя во время установки, отличного от пользователя по умолчанию некоторыми настройками, с доставлением недостающих пакетов.

Так, например, в качестве таких «небольших» настроек могут быть ссылки с рабочего стола на некоторые ресурсы, специфичные для Children, может быть несколько изменена структура меню (специфичные для Children пункты выведены на передний план, а те, которые вряд ли пригодятся детям, — наоборот) и т. п.

Боекомплект преподавателя

Другой вариант «Устанавливаемого Children», который может пригодиться, — это установочные диски для быстрого разворачивания класса на неизвестной территории. Как показала практика, изначальный вариант с разворачиванием всего с чистого листа может применяться практически в любых условиях. Проблема, с которой можно столкнуться, очень похожа на проблему по школьному проекту: железо везде разное, для разного железа оптимально подходят разные решения. Одно условие всегда примерно одинаково: есть некий класс с примерно равными по характеристикам компьютерами. Возможно, есть какая-то сеть (а, возможно, и нет, или ее можно организовать). В этом случае довольно универсальный вариант — установка на все компьютеры, при этом при наличии сети (или возможности ее организовать) один компьютер выделяется под «сервер». Под «серве-

ром» в данном случае понимается некий компьютер, не отличимый от других (один из класса, возможно, самый мощный), за которым будет работать ребенок (или преподаватель, если будет возможность не использовать его детьми), но который будет отличаться от всех остальных только тем, что на нем будет работать некоторое количество служб, а также будут храниться все работы всех детей (для того, чтобы у них была возможность работать за разными компьютерами, а не «драться» за место).

В качестве комплекта для установки предполагается 2 CD или 1 DVD, с которых будут ставиться и «сервер», и клиентские системы.

Например, изначально будет ставиться «сервер». Когда становится известен IP или хостнейм сервера, ставятся системы на остальные компьютеры, при установке прописываются необходимые значения с учетом установки сервера (для того, чтобы не нужно было производить последующую настройку). Итого, после установки с минимумом действий будет получаться класс, в котором:

- будет один samba-сервер. На клиентских компьютерах на рабочем столе будет ссылка на соответствующий ресурс, куда дети смогут соханять свои работы. Исходные материалы также будут раздаваться по samba, однако на всякий случай на каждом компьютере будет локальная копия, без ссылки с рабочего стола (если не был выбран безсетевой вариант установки).
- jabber-сервер. На клиентских компьютерах — установленные и настроенные под работу клиенты.
- возможно управление всеми компьютерами (в т. ч. и сервером) одновременно с любого компьютера класса (правда, пока через командную строку).

К сожалению, надеяться на то, что установленные системы будут единственными на компьютерах, не приходится. Кроме того, не редки задачи, когда работы детей могут понадобиться на компьютере с Windows (например, поделиться с соседним классом, работающим под Windows, скопировать файлы на ноутбук с Windows, подключенный к сети, и т. п.).

Заключение

Представленный вариант дистрибутива в формате Live CD, основанный на бранче ALT Linux 4.1, не является окончательным вари-

антом. Доработке подлежит и сам курс, и техническая база (переход на ALT Linux 5.0). С каждым годом применения курса на практике появляются новые идеи, новые приемы и задания, которые находят отражения в новых версиях.

Литература

[1] *Панюкова А. А.* Создание обучающего курса для детей на базе Linux // Третья конференция «Свободное программное обеспечение в высшей школе», Переславль, 2–3 февраля 2008 г. — М., ALT Linux: 2008

[2] *Панюкова А. А.*, *Якшин М. М.*, *Панюкова Т. А.* Методика проведения учебных занятий с использованием свободно распространяемого программного обеспечения // Роль и место самостоятельной работы студентов в образовательном процессе вуза. Юбилейная региональная научно-методическая конференция (4–6 февраля 2008 г.): Сб. науч. тр. — Челябинск, Издательство ЮУрГУ: 2008. — Т. 1. — ISBN 978-5-696-03714-1

[3] *A. Panyukova*, *M. Yakshin*, *T. Panyukova.* Organization and methodics for realization of computer graphics studying using free software // Proceedings of the 10th International Workshop on Computer Science and Information Technologies, Antalya, Turkey, September 15–17, 2008: Сб. науч. тр — Уфа, Редакционно-издательский комплекс УГАТУ: 2008. — Т. 1. — С. 238 — ISBN 978-5-86911-787-8

М. М. Якшин Москва, МГТУ им. Н. Э. Баумана

Проект: ALT Linux Children http://www.altlinux.ru

Свободное ПО для внешкольных занятий с детьми

Аннотация

В докладе рассматриваются вопросы применения свободного ПО в рамках внешкольных творческих занятий с детьми (в детских лагерях, кружках, студиях и т. п.). По результатам внедрений обобщается полученный опыт ведения курса на базе дистрибутива ALT Linux Children, делаются выводы о целесообразности таких занятий, полученных результатах, встреченных вопросах и проблемах.

В последние годы актуальными становятся вопросы применения свободного ПО в образовании. Использование свободного ПО в образовании, в частности в школьной практике, активно изучается и поддерживается на государственном уровне программами внедрения[1]. Тем не менее, кроме школьного образования и воспитания, существует еще сфера внешкольных занятий — как правило, творческих. Дети занимаются с преподавателями в детских лагерях, кружках, студиях и т. п. В сферу внешкольных программ, кроме традиционных занятий спортом, танцами, актерским мастерством, живописью, скульптурой, декоративно-прикладными искусствами, с недавних пор добавились еще и занятия на компьютере — как правило, тоже с творческим уклоном.

Идея использования свободного ПО для проведения внешкольных компьютерных занятий с детьми сравнительно молода, но, тем не менее, авторами накоплен некоторый опыт при проведении подобных занятий с 2006 года [2], [3].

С помощью специально подготовленного дистрибутива ALT Linux Children были проведены циклы занятий в различных детских лагерях. Можно отметить следующие внедрения и определенные результаты, достигнутые при выполнении каждого из них:

- Занятия с детьми сотрудников базы отдыха «Наука» при Южно-Уральском Государственном Университете[4].
- Занятия с детьми, отдыхающими в Детском Оздоровительном Лагере «Березка» при Южно-Уральском Государственном Университете[4].
- Занятия с детьми во всесоюзном лагере информационных технологий «Страна КОМПЬЮТЕРиЯ».
- Прочие разовые занятия по приглашениям во Дворцах детского творчества, школах и т. п.

На основании проведенных работ можно выделить следующие типовые классы вопросов и проблем, возникающих при проведении подобной программы:

1. Типичные проблемы, с которыми сталкиваются практически все детские педагоги.

1.1. Дети — все очень разные, в том числе в отношении получения практических навыков ведения творческой деятельности с помощью компьютера. Особенно при больших размерах групп всегда чувствуется, что часть схватывает материал быстрее и требует идти вперед,

а часть — медленнее и требует подольше оставаться на месте. Необходимо структурировать курс таким образом, чтобы внутри одного занятия все обучаемые шли в примерно одинаковом темпе. Опыты варьирования продолжительности занятий показывают, что оптимальное время занятия играет в этом не последнюю роль и, в зависимости от коллектива, в котором проводятся занятия, по возможности стоит варьировать длину занятий от 40–45 минут (если разброс навыков детей значительный) до полутора-двух часов (в старших группах, где все дети достаточно быстро усваивают материал).

1.2. Один из основных педагогических приемов — «повторение» — далеко не всегда адекватно воспринимается детьми при обучении навыкам работы с ПО. Условно можно разделить детей на тех, кто повторяет за преподавателем, тех, кто повторяет за своими сверстниками, и тех, кто ничего ни за кем не повторяет. Основную проблему представляют как раз такие дети, которые принципиально не хотят повторять то, что им предлагается, переоценивают свои силы и тратят много сил, пытаясь придумать что-то свое, но в итоге ничего не успевают. Если подобных детей немного, то часто помогает персонализация обращений к ним и уделение им лично большего внимания.

1.3. Важную роль играет правильное разделение детей по возрастным группам. В идеале можно выделить 3 возрастных группы:

Младшая (7–9 лет) — наиболее эмоциональная и легко обучаемая группа; в курсе с ними проводятся, как правило, занятия по TuxPaint, покадровой анимации в GIMP и созданию видеоклипов в kdenlive. Все указанные направления требуют минимальных навыков, являются максимально красочными, эффектными и результативными. При необходимости, курс может быть растянут на 10–12 занятий — потенциал глубинного изучения у предлагаемых программ высок.

Средняя (9–12 лет) — предлагаются занятия по обработке фотографий в GIMP, работе с несложной векторной графикой (созданию коллажей из готовых картинок в Inkscape) и созданию видеоклипов.

Старшая (12–18 лет) — наиболее сложная группа — в ней дети уже с большой долей вероятности проходили какие-то предыдущие занятия и имеют значительный опыт общения с Windows-системами, который может оказывать негативное влияние и вызывать резко отрицательный эмоциональный настрой. На за-

нятиях требуется показать наиболее эффектные и привлекательные особенности предлагаемого свободного ПО (зачастую по сравнению с проприетарными аналогами) — спецэффекты в GIMP, продвинутые возможности Inkscape (тени, градиенты, размытие, фотореалистичные изображения), создание видеоклипов на продвинутом уровне. Определенную сложность представляет и то, что старшие дети, как правило, уже определились с некоторыми жизненными приоритетами и интересами — они могут совпасть или не совпасть с предлагаемыми материалами. Это наиболее открытая часть курса, которая требует от преподавателей максимальной отдачи и эрудиции за рамками обозначенной изначально программы курса.

В случае, когда приходится работать со смешаными группами, необходимо реструктурировать курс таким образом, чтобы захватывать за один поток наиболее эффектные и интересные части творческих задач, решаемых в рамках одного и того же ПО, но на разных уровнях.

2. Соседство с Windows-системами. Курсы с использованием свободного ПО часто идут параллельно курсам с использованием Windows и проприетарного ПО. Это создает определенный круг технических, психологических и организационных проблем.

2.1. Техническая проблема: зачастую организаторы не в состоянии предоставить отдельные компьютеры/лабораторию для проведения занятий и приходится проводить занятия на тех же компьютерах, где установлен Windows. Это возможно либо с помощью установки второй ОС (dual boot) на компьютеры (инсталлятор должен иметь возможность быстрого и надежного уменьшения размера разделов с другой системой и установки загрузчика для dual boot), либо с помощью использования Live CD (менее предпочтительный вариант, т. к. создает сложности с сохранением работ детей).

2.2. Психологическая проблема: дети, «привыкшие» работать с Windows или имеющие какие-то негативные предпосылки по отношению к свободному ПО и Linux в частности. Особую сложность представляют лицензионные вопросы, поднимаемые детьми, в том числе о лицензионности проприетарного ПО, используемого ими (при параллельном ведении нескольких курсов).

2.3. Соседство с другими курсами: часто в организациях, занимающихся проведением подобных курсов для детей, уже есть поставлен-

ные на поток занятия и обученные преподаватели, которые работают по собственным программам и используют проприетарное ПО. Достаточно сложно сочетать как просто программы обучения (в том числе, если эти программы эксплуатируются не первый год и дети имеют определенный опыт прохождения по программам в прошлых годах), так и проводить параллели между используемым проприетарным и свободным ПО — так, чтобы дети понимали разницу между ними, понимали, для чего это необходимо и т. п.

3. Общая проблема многих лагерей — изначальная работа с постановкой цели получения «результата» от детей, в первую очередь для отчетности (среди родителей) и популяризации места проведения занятий (некие коммерческие интересы). Как правило, это практически всегда противоречит интересам детей и процесса обучения. В вырожденных вариантах в «конечном продукте» полностью убивается творческая составляющая: все дети делают тривиальное повторение действий преподавателя без изменений.

При всем этом стоит отметить положительные моменты проведенных занятий:

- Многократно показано, что дети на творческих занятиях с успехом могут использовать в качестве инструментов как проприетарное, так и свободное ПО.
- В случаях, когда детям копировался Live CD для продолжения занятий дома, отклик достаточно высок[4], но наблюдаются определенные проблемы технического плана, связанные с запуском Live CD (в первую очередь из-за низкой компьютерной грамотности у родителей). Предложенная в 2008–2009 годах и реализованная в ALT Linux Children 4.1 схема позволяет существенно снизить прецеденты возникновения этих проблем.

Литература

[1] Распоряжение Правительства Российской Федерации от 18 октября 2007 г. №1447-р

[2] *Панюкова А. А.* Создание обучающего курса для детей на базе Linux // Третья конференция «Свободное программное обеспечение в высшей школе», Переславль, 2–3 февраля 2008 г. — М., ALT Linux: 2008

[3] *Панюкова А. А.*, *Якшин М. М.*, *Панюкова Т. А.* Методика проведения учебных занятий с использованием свободно распространяемого программного обеспечения // Роль и место самостоятельной работы студентов в образовательном процессе вуза. Юбилейная региональная научно-методическая конференция (4–6 февраля 2008 г.): Сб. науч. тр. — Челябинск, Издательство ЮУрГУ: 2008. — Т. 1. — ISBN 978-5-696-03714-1

[4] *A. Panyukova*, *M. Yakshin*, *T. Panyukova.* Organization and methodics for realization of computer graphics studying using free software // Proceedings of the 10th International Workshop on Computer Science and Information Technologies, Antalya, Turkey, September 15–17, 2008: Сб. науч. тр — Уфа, Редакционно-издательский комплекс УГАТУ: 2008. — Т. 1. — С. 238 — ISBN 978-5-86911-787-8

Р. В. Криваковская Днепропетровск, Днепропетровский государственный аграрный университет

Проект: Short-term courses management
`http://sourceforge.net/projects/short-termcourc`

Свободное программное обеспечение в управлении учебным процессом

Аннотация

В докладе будут рассматриваться предпосылки и возможности применения свободного программного обеспечения в управлении учебным процессом (в основном, в составлении расписания занятий), текущие проблемы, связанные с внедрением свободного ПО, и пути их решения. Будут рассмотрены некоторые свободные программы для управления учебным процессом (FET, Orario-elettronico, возможно, UniTime), а также мои разработки по теме.

В настоящее время наблюдается тенденция к укрупнению ВУЗов, что увеличивает нагрузку на диспетчерскую. Составление расписаний вручную в этом случае приводит к большому количеству ошибок, а сам процесс составления расписаний является очень трудоемким.

Помочь решить эту проблему могли бы программы автоматизации управления учебным процессом.

В настоящее время большинство программ в этой области составляют коммерческие и закрытые разработки. Программное обеспечение с открытым исходным кодом находится в тени.

Развитие open source могло бы помочь продвижению хороших идей в этой области, а именно обеспечить обсуждение и решение задач силами сообщества, что особенно ценно для создания сложных систем. Создатели новых алгоритмов могли бы получить более широкую аудиторию для апробации своих результатов.

Следует отметить, что наибольшая нагрузка падает на первых разработчиков, так как именно они должны пересилить себя и выложить свои продукты в открытый доступ.

Путем поиска на серверах, где размещаются свободные программы (`sourceforge.net`, `freshmeat.net` и подобные), было обнаружено, что имеется ряд свободных решений для автоматизации составления расписаний. Мной были протестированы более 10 программ. Мне бы хотелось рассказать о трех из них.

Orario-elettronico

Orario-elettronico — программа для составления расписаний, разработанная Alessandro Brunelli. Написана на PHP, работает через веб. Работает во всех основных ОС (в Windows в течение 10 дней).

В настоящее время доступна на итальянском и английском языках.

Возможности:

- Позволяет вводить информацию о преподавателях, классах, нагрузках.
- Есть возможность назначать занятия по определенным предметам в определенные аудитории и возможность гибко настраивать нагрузки по предметам (например, есть возможность ставить некоторые предметы только в определенное время или с определенным промежутком между занятиями).
- Есть возможность составления расписания вручную для преподавателей и групп.
- Есть автоматическая проверка расписаний на ошибки и поиск нераспределенных часов.

- Официально неанглийские кодировки не поддерживаются, но они работают почти на всех страницах программы.
- Можно блокировать некоторые пары для преподавателей и студентов.

Ограничения:

- Количество предметов для каждого класса не более 20.
- Количество часов для одного преподавателя на один предмет — не более 24 в неделю.

Выявленные недостатки:

- Есть возможность автоматического составления расписания, но работает она очень плохо.
- Интерфейс не совсем понятен.
- Деление классов на подгруппы и двухнедельный рабочий график не предусматриваются.
- Сообщения об ошибках нечеткие.
- Документация только на итальянском языке.

Сайт программы: `http://sourceforge.net/projects/orarelettr/`.

FET

FET — Free Evolutionary Timetabling. Разработана Liviu Lalescu.

Переведена на 17 языков (русский в их число не входит). Работает для всех основных операционных систем (в UNIX и Linux с использованием библиотеки Qt).

Возможности:

- Очень гибкая система ввода ограничений (можно поставить ограничения по всем основным параметрам расписания).
- Хорошая справка.
- Работает автоматическая генерация расписаний.
- Расписание сохраняется на диск в формате xml.

Ограничения:

- В справочнике написано, что, кроме прочего, есть возможность ручного ввода расписания, но я ее не нашла.

Выявленные недостатки:

- Не реализована печать расписаний из программы.
- При автоматической генерации расписания в аудитории ставятся только те нагрузки, для которых задана предпочитаемая аудитория в ограничениях.

Сайт программы: `http://lalescu.ro/liviu/fet/`.

UniTime

По заявлениям разработчиков, эта программа обеспечивает полноценную реализацию системы управления в образовании. Работает под Java. На сайте существует англоязычная документация и демо-ролик.

Сайт программы: `http://www.unitime.org/`.

Мои разработки

В данное время я разрабатываю систему для проверки расписаний для курсов повышения квалификации. Основной упор делается на проверку корректности составленных в ручном режиме расписаний. Сейчас есть пилотная версия программы, но она нуждается в доработке.

Сайт программы:
`http://sourceforge.net/projects/short-termcourc`.

Выводы

В настоящее время потенциал свободного программного обеспечения в управлении учебным заведением не используется в полной мере. Как мне кажется, это происходит из-за психологических причин. Выходом из этой ситуации, как мне кажется, могли бы быть следующие действия:

1. Создание собственных свободных программ, находящихся в открытом доступе.
2. Популяризация использования свободного ПО в управлении учебным процессом в той мере, в которой это возможно в каждом конкретном ВУЗе.

В. В. Яковлев, Н. М. Субоч, М. А. Ройтберг, А. Г. Кушниренко
Пущино, Институт математических проблем биологии РАН, Пущинский госуниверситет, Научно-исследовательский институт системных исследований РАН

Проект: Кумир `http://www.infomir.ru`, `http://lpm.org.ru/kumir/`

Синтаксический разбор программ, содержащих ошибки

Аннотация

Описание языка программирования определяет поведение компилятора только на правильных программах; поведение компилятора при разборе программ, содержащих синтаксические ошибки (например, диагностика ошибок) обычно никак не регламентировано и оставляется на усмотрение разработчика. В то же время для учебных систем программирования удобная диагностика ошибок имеет первостепенное значение.

Мы предлагаем способ описания поведения компилятора в случае программ, имеющих ошибки в структуре, с помощью подходящей контекстно-свободной грамматики; этот способ был реализован в системе программирования КуМир. Наш опыт показал, что предложенный подход является достаточно гибким и позволяет улучшить качество диагностики.

Постановка задачи

В настоящей работе мы рассматриваем разбор структуры программ на языке КуМир. КуМир-программа может быть представлена в виде последовательности «канонических» строк, каждая из этих строк — это либо элементарный оператор (присваивание, вызов алгоритма и т. п.), либо компонент операторной скобки (`алг`, `нач`, `кон`, `если`, `то`, `иначе`, `все` и т. д.).

Под структурой КуМир-программы мы понимаем ее представление в виде последовательности канонических строк, в котором все строки с элементарными операторами считаются эквивалентными (например, заменены специальной строкой простая команда). Структуры правильных Кумир-программ образуют контекстно-свободный язык с несложной грамматикой. В компиляторе системы КуМир анализ структур программы является отдельным модулем.

Нашей целью было разработать язык, который описывает и классифицирует ошибочные структуры программ (по возможности детально) и модуль анализа структуры программы, такие, что:

1. модуль анализа распознает не только правильные программы, но и ошибочные программы в соответствии с разработанным описанием, диагностика ошибок при этом определяется описанием;

2. модуль анализа имеет открытую архитектуру, что позволяет расширять и уточнять классификацию ошибочных ситуаций без вмешательства в исходный код системы КуМир.

Реализация последнего требования позволяет добавление типовых ошибок учеников в систему диагностики непосредственно в ходе учебного процесса.

Реализация

Грамматики, описывающие как сам язык КуМир $L(G_0)$, так и ошибочные ситуации $L(G_i)$, $i > 0$, описаны во внешних текстовых файлах в виде набора правил вывода. Данные правила упорядочены по приоритету применения (приоритет указывается в имени файла), поскольку в общем случае множества, порождаемые различными грамматиками $G_i(\mathcal{H})$ над одним и тем же алфавитом $\mathcal{H}$, могут пересекаться. Грамматика, описывающая язык КуМир, то есть все множество правильных программ, имеет наивысший — нулевой приоритет.

С каждым правилом вывода, содержащим терминалы (т. е. символы алфавита $\mathcal{H}$, соответствующие типам канонических КуМир-строк) может быть связана некоторая нагрузка, описывающая выполняемые МП-автоматом действия при достижении данного терминального символа. Эти действия представляют собой инструкции на языке ECMAScript, более известном как JavaScript, которые выполняются в случае применения данного правила и могут выдавать сообщения об ошибках, устанавливать ранги отступов, и т. д.

Благодаря такому техническому решению можно легко программировать поведение анализатора в различных ситуациях разбора программ.

Результаты

В текущем варианте системы КуМир (на момент данной публикации — текущая версия из SVN-репозитория) реализован принципиально новый метод разбора структуры пользовательких программ. Новизна метода заключается в одновременном использовании правил вывода из различных грамматик, порождающих разные языки, в то время как существующие средства генерации анализаторов КС-грамматик, такие как `yacc` или `bison`, могут работать только с одной грамматикой.

Этот метод позволяет, во-первых, корректно диагностировать программы, содержащие более одной ошибки в своей структуре, а во-вторых, распознавать типовые и наиборее распространенные ошибки более крупным планом, так как это сделал бы человек, а не машина.

Открытая система правил вывода ошибочных ситуаций позволяет еще на этапе бета-тестирования «обучить» КуМир-систему распознавать различные ошибочные ситуации, которые допускают учащиеся, для того чтобы впоследствии выдавать соответствующие диагностические сообщения применительно к конкретным ситуациям.

Тем не менее данная работа является экспериментальной, и у разработанного метода нашелся существенный недостаток, а именно длительное время разбора, преодолеть который удалось только ценой ухудшения качества диагностики. Данный метод легко распараллеливается, а тенденция развития современных процессоров ведет к увеличению числа одновременно выполняемых инструкций. Однако экспоненциальная зависимость времени работы данного алгоритма от длины входной цепочки не позволяет надеяться на технический прогресс даже в отдаленной перспективе, поэтому необходимо продолжать исследования в области альтернативных способов разбора программ. Возможным альтернативным подходом является отказ от явного перечисления ошибочных ситуаций и переход к построению правильной программы, наиболее похожей на данную ошибочную.

Литература

[1] *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. В двух томах. М.: Мир, 1978. пер. с англ.

[2] *Кушниренко А. Г., Лебедев Г. В., Зайдельман Я. Н.* Информатика. 7–9 класс: Учебник для общеобразовательных учреждений. —

3-е изд., стереотип. – М.: Дрофа, 2002.

Н. М. Субоч, А. В. Карпов, Е. А. Святушенко, М. А. Ройтберг
Пущино, Институт математических проблем биологии РАН, Пущинский государственный университет, Гимназия Пущино

Проект: Кумир `http://www.infomir.ru`, `http://lpm.org.ru/kumir`

Методы тестирования в разработке системы обучения программированию КуМир

Аннотация

В докладе обсуждается методика тестирования, использовавшаяся при разработке системы программирования КуМир. Описана структура корпуса тестов, методика создания этого корпуса и средства поддержки тестирования. Использованная методика может быть полезна при разработке других обучающих систем.

Введение

В ходе разработки системы КуМир (как и при разработке всех достаточно сложных систем) мы столкнулись с двумя проблемами. Во-первых, хотя для системы было разработано достаточно подробное описание, не все функциональные возможности оказывались реализованными. Во-вторых, при развитии системы могут перестать правильно работать уже отлаженные компоненты. Чтобы преодолеть эти трудности, нами был создан корпус тестов и разработан протокол тестирования, включающий как автоматизированную, так и интерактивную составляющие. Идея примененного подхода — в подготовке максимально подробного описания системы и создании корпуса тестов в соответствии с этим описанием. Такой подход не гарантирует отсутствие ошибок, но, как показал наш опыт, существенно повышает эффективность разработки.

Автоматизированное тестирование

В КуМире предусмотрен так называемый пакетный (*batch*) режим выполнения, при котором графический интерфейс системы не загру-

жается, а результаты выполнения программы выводятся в файлы. При этом создаются три файла вывода, хранящие содержимое трех областей главного окна системы КуМир:

- *Поле с текстом программы.* Оно также содержит информацию о подчёркивании синтаксических ошибок в программе.
- *Поле диагностики.* Содержит тексты сообщений о синтаксических ошибках.
- *Поле ввода-вывода.*

Нами был создан набор тестовых программ для выполнения в пакетном режиме и набор эталонных файлов вывода каждого теста. Наличие несоответствий между полученными выходными файлами, полученными при выполнении теста, и эталонными выходными файлами, говорит о найденных ошибках. Различия в выходных файлах первого типа означают ошибку в подчеркивании синтаксических ошибок; в файлах второго типа — неверную диагностику ошибок; в файлах третьего типа — нарушение процесса выполнения программы.

Для того чтобы автоматизировать процесс тестирования, был написан *скрипт тестирования*, который последовательно обрабатывает все тесты и сравнивает полученные данные с эталонными. Найденные несоответствия записываются в файл отчета. Файл отчета скрипта тестирования представляет собой последовательность пар различающихся строк из выходных файлов. Для каждой пары указывается имя выходного файла и номера строк.

Как создаются эталонные тесты

Мы разделяем тестовые примеры на три типа:

Выполнимые — не содержат синтаксических ошибок и ошибок выполнения, выполняются нормально.

С ошибками разбора — содержат синтаксические ошибки.

С ошибками выполнения — не содержат синтаксичесих ошибок, но их выполнение завершается аварийно.

При разработке тестов мы рассматриваем все формальные правила описания языка и создаем примеры, следуя этим правилам. Составляя правильные конструкции, мы получаем выполнимый тест. Перебирая различные варианты ошибочного написания рассматриваемого правила, мы получаем набор тестов с ошибками. Таким образом, мы можем покрыть всю функциональность языка программирования.

Тестирование интерфейса

Описанный выше метод позволяет тестировать только разбор и выполнение программ, в то время как необходимо также выполнять тестирование интерфейса. Для этого была написана система так называемых *интерфейсных тестов*, охватывающая функциональность всего интерфейса системы. Каждый такой тест представляет собой инструкцию для тестера с алгоритмом действий, которые ему необходимо совершить. Тестер должен выполнить требуемые действия и сравнить результат с ожидаемым.

Для создания системы интерфейсных тестов нами было разработано по возможности формальное описание интерфейса системы КуМир. Мы старались описать все состояния системы, действия пользователя в этих состояниях и реакции системы. Каждый раздел описания стал основой для написания интерфейсного теста или группы тестов. Такое описание, естественно, оказалось неполным, поэтому система интерфейсных тестов расширялась по мере отладки системы.

Выводы

Даже в производственных системах, которые используются много лет, находятся ошибки[2]. То есть долговременная эксплуатация не может быть заменена никакими специальными системами тестирования. Тем не менее, при разработке системы обучения программированию КуМир хорошо себя зарекомендовали следующие методы тестирования.

- Создается корпус тестов, состоящий из тестов автоматического выполнения и интерфейсных тестов. Тесты обеих групп создаются на основе формальных описаний. Тесты автоматического выполнения создаются на основе описания языка КуМир, включая описания диагностики ошибок. Интерфейсные тесты создаются на основе специально подготовленного описания интерфейса.
- Тесты автоматического выполнения делятся на три группы: 1) Тесты синтаксических ошибок; 2) Тесты ошибок выполнения; 3) Тесты правильных программ. Результатами выполнения каждого такого теста являются текстовые файлы, содержащие диагностику синтаксических ошибок, диагностику ошибок выполнения, а также вывод программ. В перспективе будет преду-

смотрена возможность протоколирования всех промежуточных действий. Эталонные значения файлов прилагаются к каждому тесту автоматического выполнения. Для выполнения этих тестов в системе КуМир предусмотрен специальный пакетный режим и созданы средства тестирования. Эти средства позволяют выполнить все тесты в пакетном режиме, отметить тесты, результат выполнения которых отличается от эталонных, и указать, в чем состоит расхождение с эталоном. Это тестирование запускалось регулярно в течение последнего года разработки системы КуМир.

- Для тестирования интерфейса было создано описание возможных состояний системы КуМир с точки зрения пользователя, действий пользователя в этих состояниях и реакций системы. На основе этого описания был создан корпус интерфейсных тестов. Каждый тест по возможности включает максимально точные инструкции для человека, который ведет тестирование.

  Такой протокол тестирования позволил избежать повторного внесения ошибок при развитии системы и существенно ускорил ее отладку.

Литература

[1] *Кушниренко А. Г., Лебедев Г. В., Зайдельман Я. Н.* Информатика, 7–9 классы.

[2] *Дейкстра Э.* Дисциплина программирования.

Г. В. Курячий Москва, ALT Linux

Проблемы и методы командной разработки свободных учебных материалов

Аннотация

Ключевые слова: ПСПО, СПО, документация, командная разработка, учебный материал, wiki.

Приведена история создания рабочей группы по документированию ПСПО, организация процесса создания учебного материала, структура получившегося продукта и поддерживающей его технологии. Делается попытка истолкования положительных и отрицательных результатов работы и обобщения полученного опыта.

Тезисный план

Постановка частной задачи

Немного истории пилотного внедрения ПСПО в российских школах. Одна из задач: обучение взрослых (учителей, в т. ч. предметников, администраторов, директоров и пр.). Непосредственная задача: 10 отчуждаемых курсов.

Условия и обстоятельства

- Неблагоприятные:
  - Проблема выбора исполнителя: педагог не знает Linux, программист не знает, как учить, педагог-программист занят 23–25 часов в сутки.
  - Лимит ресурсов (и людских, и временных): в идеале разработка курсов должна предшествовать внедрению.
  - Быстрое устаревание имеющихся материалов.
- Компромиссные:
  - Отказ от написания 100% лекционного материала в пользу «двухуровневой» структуры.
  - Отказ от полного методического оснащения «курсов».
  - Баланс между «необходимым» (для предполагаемой аудитории) и «интересным» (для участников проекта).

  - Модульная структура курсов.

- Благоприятные:
  - Нашелся человек с опытом преподавания и широкими знаниями Linux.
  - Подобралась команда, готовая «помогать» в реализации проекта (на самом деле — сделавшая львиную долю работы).
  - Стояло лето, и участники команды — студенты и преподаватели — могли выделить заданное время под проект.

Организация рабочего процесса

Технически: использование wiki (MoinMoin) с дополнительным программированием и установленной дисциплиной работы.

Also Sprach Zaratustra

1. «Лекции» (16 ак. ч. в неделю);
2. Конспектирование online;
   - критерий качества: не упускать ключевых моментов лекции.
3. «Расшифровка»: воссоздание структуры лекции;
   - условие: расшифровщик присутствует на лекции;
   - критерий качества: «переводчику» должно быть понятно, даже если он не был на лекции.
4. «Перевод на русский»;
   - критерий качества: не потерять в процессе литредактуры значимых утверждений и формулировок.
5. «Научное редактирование».
   - критерий качества: связность конечного текста.

За каждый фрагмент отвечает конкретный исполнитель каждого этапа.

Содержательное редактирование сведено к минимуму (иногда с небольшой потерей качества), организаторская работа и т. п.

Наполнение курсов материалами

1. Архитектор составляет «Лекционный минимум» — развернутый тематический план будущего фрагмента курса (модуля);
2. За каждый модуль отвечает конкретный исполнитель;
3. Исполнитель из команды подбирает ссылки на материалы, содержащие информацию по каждой теме;
4. Все использованные внешние материалы импортируются.

Результат: продукт

Структура получившегося корпуса материалов:

- Материал = Паспорт + Файлы; внутренний и внешние материалы;
- Модуль = Лекционный Минимум * Материалы + Комментарии;
- Курс = сумма Модулей;
- Побочный объект: Книг = самодостаточная сумма Материалов.

Результат: технология

- Использование Wiki MoinMoin;
- Доработка MoinMoin;
- Дисциплина работы: инварианты во внутренних материалах, правила импорта внешних и т. п.

Приятная неожиданность: отчуждаемость контента, в т. ч. в виде автономного www-сервера на CD.

Задние мысли

- От каждого — по способностям;
  - Только один род задач на каждого исполнителя;
  - Разумная нагрузка.
- Высокая роль внутренней мотивации;
- Необходимость автоматического разделения «готовой» и «разрабатываемой» частей при отчуждении;
- Необходимость отслеживать устаревание.

Можно ли таким способом написать книгу? Сделать многокомпонентную документацию?

Е. Д. Сыромятников Москва, факультет ВМК МГУ им. М. В. Ломоносова

Проект: семинар UNИХ http://uneex.ru/

MoinMoin: обзор, опыт использования и администрирования

Аннотация

Рассматривается wiki-движок MoinMoin, процесс его установки, настройки, использования и администрирования, а также приводятся результаты опыта использования.

Что такое MoinMoin и что такое wiki

MoinMoin — сервер для совместной работы с гипертекстовыми документами, написанными в специальной разметке (wiki-разметке), или, как его часто называют, wiki-движок.

Wiki-разметка является одним из способов разметки форматированного текста, который отличается намеренным упрощением элементов синтаксиса. Например, для обозначения полужирного текста в большинстве языков wiki-разметки используется обрамление тремя кавычками, а для форматирования таблицы — разделение ячеек двумя вертикальными слешами (ср. с HTML, где для первого используется `<strong></strong>`, для второго — `<table><tr><td></td></tr></table>`). Слова в `CamelCase` рассматриваются как ссылки на другие страницы.

В спектр возможностей, предоставляемых wiki-движком, входят отображение (рендеринг) страниц, их изменение и управление ими. Кроме того, одной из важной особенностей любого wiki-движка является поддержка управления ревизиями.

Обзор MoinMoin

MoinMoin является реализацией wiki-движка на языке Python. Он использует в своей работе следующие виды сущностей:

- Пользователь.
- Страница — основная сущность, с которой взаимодействует wiki-движок. В MoinMoin существует ряд специальных страниц:
  - Системные страницы;
  - Страница шаблона;
  - Страница группы. На данной странице указывается список пользователей, входящих в данную группу.
- Прикрепленый файл.

Среди особенностей MoinMoin можно выделить следующие:

- Иерархичность пространства имен страниц;
- Поддержка ACL;
- Механизм категорий;
- Механизм шаблонов (emplate);
- Механизм макросов;
- Механизм действий (action);
- Механизм переменных (variable);
- Механизм поиска.

Установка

В простейшем случае можно скачать tarball с `http://moinmo.in/MoinMoinDownload`, развернуть его в любой директории и запустить `wikiserver.py`.

Можно использовать MoinMoin как совместно с имеющимся HTTP-сервером (apache, nginx, lighttpd), так и в как самостоятельный HTTP-сервер.

Настройка

Настройка отдельной moin instance производится посредством файла `wikiconfig.py`, в котором содержатся глобальные настройки конкретной moin instance (название wiki, пути к тем или иным файлам, ACL, ...).

MoinMoin поддерживает wiki farm, то есть несколько wiki, использующих одну инсталляцию MoinMoin и единый конфигурационный файл.

Администрирование

Благодаря механизму ACL, достаточно добавить необходимых пользователей в соответствующие группы и указывать ACL у страниц.

- ACL по умолчанию настраивается при помощи `acl_rights_default`, `acl_rights_before`, `acl_rights_after` в `wikiconfig.py`.
- Существует механизм AutoAdmin, который позволяет настраивать выдачу прав на администрирование отдельных страниц некоторым группам и пользователям.

Опыт использования

Далее рассматривается опыт использования MoinMoin в рамках проекта документирования ПСПО, где было необходимо организовать совместную работу по обработке и подготовке конспектов лекций. Частично данная задача была решена имеющимися средствами MoinMoin, частично — добавлением функциональности путем написания дополнительных макросов и действий.

В рамках поставленной задачи, среди прочего, были необходимы следующие возможности:

- Организация процесса обработки фрагментов лекций: просмотр статуса по фрагменту конспектов лекций, просмотр статистики по нескольким фрагментам.
- Хранение иерархии файлов, связанной с определённой страницей.
- Создание ссылок на сущности различного вида и формирование текста ссылки в зависимости от содержимого сущности.

Создание макросов и действий

Можно выделить следующие основные подходы при добавлении функциональности:

- Создание нового макроса. Используется, когда необходимо дополнить возможности wiki-разметки.
- Создание пары макрос-действие. Используется при добавлении новой функциональности, действие которой локально.
- Создание действия и модификация используемого внешнего вида (skin). Используется при добавлении новой функциональности, действие которой глобально.

Выводы

MoinMoin изначально обладает богатой функциональностью, а также предоставляет широкие возможности по собственному расширению.

Ряд вещей можно сделать как встроенными средствами MoinMoin (при некоторых дополнительных соглашениях), так и посредством дополнительного программирования (что, возможно, является лучшим, но иногда достаточно ресурсоёмким вариантом).

В. Г. Маняхина Москва,
Московский государственный педагогический университет

О некоторых возможностях использования LMS Moodle в учебном процессе педагогического ВУЗа

Аннотация

В докладе дается общая характеристика системы управления обучением Moodle и рассматриваются возможности использования этой системы в учебном процессе педагогического ВУЗа.

На математическом факультете МПГУ ведется работа по созданию информационно-образовательной среды (ИОС) факультета. Сейчас разрабатывается ИОС кафедры ТИДМ. Важным фактором в успешной реализации среды является выбор программной оболочки. Проанализировав соответствующее программное обеспечение, мы остановили свой выбор на Moodle.

Moodle — это система управления обучением, или LMS (Learning Management System), в нашей стране подобные программы также

называют системами дистанционного обучения (СДО). Moodle является СПО с лицензией GPL. Moodle — аббревиатура от Modular Object-Oriented Dynamic Learning Environment (модульная объектно-ориентированная динамическая обучающая среда). Благодаря своим функциональным возможностям система приобрела большую популярность и успешно конкурирует с коммерческими LMS. Официальный сайт проекта: `http://www.moodle.org/`.

Moodle дает возможность проектировать, создавать и в дальнейшем управлять ресурсами информационно-образовательной среды. Система имеет удобный интуитивно понятный интерфейс, используется WYSIWYG HTML-редактор; кроме того, существует возможность ввода формул в формате TEX или Algebra. Можно вставлять таблицы, схемы, графику, видео, флэш и др.

Можно использовать как тематическую, так и календарную структуризацию курса. При тематической структуризации курс разделяется на секции по темам. При календарной структуризации каждая неделя изучения курса представляется отдельной секцией, такая структуризация удобна при дистанционной организации обучения и позволяет обучающимся правильно планировать свою учебную работу.

В электронный курс легко добавляются различные элементы: лекция, задание, форум, глоссарий, wiki, чат и т. д. Для каждого электронного курса существует удобная страница просмотра последних изменений в курсе.

Достаточно хорошо продумано и администрирование системы.

Moodle обладает большим набором средств коммуникации. Это не только электронная почта и обмен вложенными файлами с преподавателем, но и форум (общий новостной на главной странице программы, а также различные частные форумы), чат, обмен личными сообщениями, ведение блогов.

В LMS Moodle имеется обширный инструментарий для создания тестов и проведения обучающего и контрольного тестирования. Поддерживается несколько типов вопросов в тестовых заданиях. Moodle предоставляет много функций, облегчающих обработку тестов. В системе содержатся развитые средства статистического анализа результатов тестирования.

С помощью LMS Moodle нами создан электронный курс дисциплины «ПО ЭВМ». Мы выбрали тематическую структуризацию курса (по количеству модулей курса). Каждый модуль охватывает определен-

ный раздел и включает в себя теоретический и практический материал, контрольные задания, тесты, а также методическое обеспечение самостоятельной внеаудиторной работы.

Электронные лекции являются для студентов, особенно для первокурсников, большим подспорьем, так как восприятие теоретического материала на слух с его одновременным конспектированием для них представляет большую сложность. Имея электронный текст лекций, студенты могут исправлять ошибки в своих конспектах, сверяя их с электронным вариантом.

Особое внимание мы уделили разработке учебно-методических материалов для проведения практических занятий. Система электронных практических занятий построена таким образом, чтобы студенты могли самостоятельно освоить и проработать материал занятий. Преподаватель выступает в качестве консультанта и обращает внимание студентов на особенности работы с тем или иным ПО, и предупреждает их о типичных ошибках, допускаемых при работе.

Использование таких элементов LMS Moodle, как форум и Wiki, позволяет подключить студентов к созданию нового образовательного контента. Например, преподаватель создает форум, на котором обсуждаются программы, относящиеся к классу СПО. Студенты самостоятельно знакомятся с рекомендуемыми программами и обсуждают возможности их использования на форуме. При помощи Wiki совместно создается справочное руководство по работе с наиболее интересными программами. Эта работа осуществляется во внеаудиторное время, взаимодействие в группе и координация проекта преподавателем осуществляется дистанционно при помощи средств, предоставляемых Moodle. Созданный таким образом образовательный контент, размещенный в ИОС, безусловно, будет полезен многим.

Использование LMS Moodle в образовательном процессе педагогического ВУЗа не только позволяет эффективно организовать аудиторную и внеаудиторную самостоятельную работу студентов, но и помогает будущим учителям приобщиться к инновационным образовательным технологиям и в дальнейшем использовать их в своей профессиональной деятельности.

Н. Ю. Иванова Москва,
Московский Государственный Педагогический Университет

Опыт использования OpenOffice.org в курсе «Программное обеспечение ЭВМ» на математическом факультете МПГУ

Аннотация

В докладе освещается опыт преподавания курса Программное обеспечение ЭВМ с использованием OpenOffice.org, рассматриваются методические аспекты преподавания этого курса с использованием ПСПО, опыт адаптации учебных материалов, разработанных для MS Office, к использованию с OpenOffice.org

Место и структура курса ПО ЭВМ

Курс Программное обеспечение ЭВМ преподается на 1 курсе в течение двух семестров. На аудиторные занятия отводится 126 часов, из них 54 — лекции и 72 — лабораторные работы.

Задача курса состоит в том, чтобы сформировать у студента целостное представление о принципах построения и функционирования современных операционных систем, о месте и роли современных технологий в решении прикладных задач с использованием компьютера.

В рамках курса ПО ЭВМ выделяются следующие разделы:

1. Программное обеспечение ЭВМ. Классификация.
2. Операционные системы. Операционные оболочки.
3. Прикладное программное обеспечение. Классификация.
4. Обработка текстовой информации на ЭВМ. Текстовые редакторы.
5. Обработка табличной информации на ЭВМ. Табличные процессоры.
6. Базы данных. Системы управления базами данных.
7. Решение математических задач на ЭВМ. Математические пакеты.
8. Обработка графической информации на ЭВМ. Системы машинной графики.

9. Вспомогательные системные программы.
10. Сжатие данных. Архиваторы.
11. Компьютерные вирусы. Антивирусные программы.
12. Системы программирования.

Этапы перехода на свободное программное обеспечение

1. Подготовительный этап. Использование СПО фрагментарно, в рамках курса Программное обеспечение ЭВМ.
2. Основной этап. Полный переход к использованию СПО в рамках курса ПО ЭВМ.

На подготовительном этапе в рамках курса Программное обеспечение ЭВМ в разделе Операционные системы, кроме изучавшихся ранее MS DOS и MS Windows, стали знакомить студентов с ОС Linux. В основном, с командами для работы с файловой системой.

На следующем этапе внедрения СПО был осуществлен полный переход на Linux и ПСПО. Это потребовало изменения разработанных ранее лабораторных работ и лекций, хотя в целом структура курса осталась неизменной.

Адаптация курса к использованию ПСПО

Наш лекционный курс и до перехода на ПСПО разрабатывался максимально независимым от конкретного программного обеспечения, в основном мы старались говорить об общих принципах обработки информации, например, текстовой, а не о конкретных приемах. Поэтому в той части курса, которая относилась к прикладному программному обеспечению, особых изменений не потребовалось. Наибольшие изменения произошли в разделе Операционные системы, т. к. ранее в этом разделе основной акцент был сделан на MS Windows, а остальные ОС рассматривались обзорно. Теперь подробно рассматривается Linux, а MS Windows — обзорно.

Упражнения и задачи, предлагаемые ранее студентам для выполнения в MS Office, могут быть решены и в OpenOffice.org. Поэтому изменения, вносимые в практическую часть курса, носили в большинстве своем «редакторский» характер. Они коснулись названий пунктов меню, расположения тех или иных пунктов, изображений кнопок панелей инструментов и т. д.

В некоторых случаях потребовалось изменить описание приемов работы с некоторыми объектами, например оглавлением или библиографией докумета в редакторе Writer.

Потребовалось отдельно описать работу с редактором Math, тогда как ранее работа с формулами рассматривалась в рамках изучения MS Word. Необходимо отметить, что возможность работы с формулами важна для наших студентов, т. к. они обучаются по специальности математика и информатика.

Выводы

Подводя итоги, можно сказать, что переход на СПО в рамках изучения курса ПО ЭВМ требует подготовительной работы и решения не только технических, но и методических проблем. В частности, со строны преподавательского состава требуется переработка существующих методических материалов, их редактирование и частичное изменение. В то же время все задачи, которые ставятся перед курсом, могут быть решены с помощью СПО.

М. А. Гусаров Новосибирск, ALT Linux

Открытые проекты как место практики студентов

Аннотация

В докладе рассматривается возможность участия в проектах по разработке свободного ПО в качестве замены традиционным курсовым и годовым студенческим работам.

В докладе предлагается вместо модельных заданий, обычно составляющих семестровые и годичные студенческие работы на профильных факультетах, предлагать студентам участие в *уже существующих* открытых проектах по разработке свободного ПО.

Такая форма обучения выгодна для разработчиков свободного ПО:

- Любой свободный проект нуждается в дополнительных разработчиках. Большинство проектов имеет длинный список задач (от тривиальных до сложных), которые не получают решения из-за недостатка рабочих рук.

Также этот вариант привлекателен для (определенного типа) студентов:

- Участие в разработке какой-нибудь настоящей программы, особенно если студент сам ей пользуется, заведомо интереснее выполнения искусственной задачи;
- студент может выбрать один из множества проектов по заданной тематике.

Кроме того, такая форма интересна с точки зрения преподавателя:

- Участие в открытом проекте позволяет попробовать «настоящую» работу без проблем, связанных с практикой в коммерческих фирмах;
- такая форма обучения позволяет проиллюстрировать возможность коммерческой работы над свободным софтом (в данном случае платой за работу является зачет практики);
- студент обучается работе в команде, в том числе распределенной;
- показывается практическое применение обычно лишь теоретически изучаемых инструментов совместной работы (wiki, системы контроля версий, багтрекеры, списки рассылки);
- не требуется искусственное создание примеров и заданий для практики, свободные проекты всегда имеют огромное количество готовых задач, требующих разрешения;
- в силу публичности проектов полностью исключается плагиат;
- в силу культуры открытых проектов у студентов начисто отбивается привычка к велосипедостроению, прививается практика грамотного распоряжения уже наработанными в сообществе ресурсами: библиотеками, сетевыми протоколами, языками.

Кроме непосредственной разработки СПО, возможно участие в смежной деятельности: переводе документации и интерфейса программ, которое позволяет избавиться от переводов никому не нужных текстов в курсе иностранного языка.

Участие в открытых проектах уменьшает контроль преподавателя за выполнением задания по сравнению с модельным примером, но это компенсируется высокой документированностью процесса разработки в открытых проектах (публичные системы контроля версий, архивы списков рассылки и т. д.).

Доклад не претендует на звание методической разработки, а представляет видение проблемы участником открытых проектов и бывшим студентом.

М. В. Быков Москва,
Российский Государственный Гуманитарный Университет

Проект: http://diglossa.org

Многоязычная библиотека diglossa.org

Аннотация

`http://diglossa.org` — проект свободной электронной многоязычной библиотеки. Написана на Ruby on Rails с интенсивным использованием JS-библиотеки JQuery.

`http://diglossa.org` — проект свободной электронной многоязычной библиотеки. Идея проста — есть два окошка: слева — исходный текст, справа — соответствующие ему переводы. Переводы можно переключать и сравнивать между собой. Для текстов, сложных для перевода, это абсолютно необходимая и классическая процедура. Непонятно, почему это не было до сих пор реализовано в Сети, поскольку технически задача не сложна.

Приложение написано в классическом стиле на RoR [1], есть административный интерфейс, позволяющий добавлять в библиотеку названия текстов в виде дерева (используется плагин acts_as_tree [2]). Каждому названию текста соответствует два или больше названий файлов переводов. Переводы хранятся в плоских текстовых файлах в формате utf-8 [3] и находятся под управлением системы контроля версий git [4]. При обновлении файла текста или перевода автоматически происходит переиндексация базы данных полнотекстового поиска, для которой используется свободный поисковый движок Sphinx [5] и RoR-плагин Thinking Sphinx[6].

Хитрость (и ограничение данного подхода) состоит в том, что тексты и оригинала, и всех переводов отредактированы таким образом, чтобы каждому абзацу оригинала соответствовал в точности один абзац перевода. Для ускорения работы тексты подгружаются не целиком, но чанками по нескольку абзацев. Также чанками происходит

и процесс переключения части текста на другой перевод. Для отображения текста на странице, выравнивания абзацев и переключения переводов используется js-фреймворк Jquery [7].

Таким образом, движок приложения можно считать очень простым, построенным «на коленке» из стандартных, очень распространенных и очень популярных сейчас средств, простых, надежных и удобных.

Приложение и все тексты переводов опубликованы под свободной лицензией и доступны по адресу: `git://github.com/mbykov/diglossa.org.git`.

Список литературы

1. Ruby on Rails — `http://rubyonrails.org/`
2. acts_as_tree — `git://github.com/rails/acts\_as\_tree.git`
3. utf-8 — `unicode.org`
4. git — `http://book.git-scm.com`
5. Sphinx — `http://www.sphinxsearch.com/`
6. Thinking Sphinx — `http://ts.freelancing-gods.com/`
7. JQuery — `http://jquery.com/`

И. С. Игнатьев, А. Б. Грунау Москва,
Московский Государственный Институт Электроники и Математики

Применение Open Source в курсе «Методы и Средства Анализа данных»

Аннотация

В середине 2008 года потребовалось разработать курс, освещающий различные аспекты анализа данных. Для его разработки были выбраны средства FLOSS — движок MediaWiki для хранения лекций и методических указаний и среда Weka для выполнения лабораторных работ. В результате курс был проведен уже в первом семестре 2008–2009 учебного года.

В начале учебного года встала задача создать курс, который давал бы студентам знания по методам анализа данных и навыки использования и создания средств анализа данных. При этом курс должен быть достаточно легким для усвоения в силу неудовлетворительной подготовки студентов по базовым математическим дисциплинам, в то время как все источники характеризовали анализ данных как достаточно сложную дисциплину и рекомендовали прежде тщательно изучить сопутствующие разделы математической теории[1].

В соответствии с общей политикой кафедры для реализации данной задачи были выбраны средства, распространяемые под свободной лицензией. Лекции и методические указания писались и распространялись при помощи движка MediaWiki, доступного под лицензией GNU GPL v2 и выше. Для лабораторных работ была в результате обзора выбрана среда анализа данных Weka — Waikato Environment for Knowledge Analysis — распространяемая под лицензией GNU GPL v2.

Использование движка MediaWiki дало студентам и преподавателям возможность:

- быстро писать и исправлять лекции;
- писать и исправлять лекции удаленно;
- при помощи специальной математической разметки удобно представлять формулы, которые часто используются в лекционном материале;
- при нахождении неточности или недостаточно полной информации в материале оперативно исправить этот недостаток;
- заинтересовать студентов в изучении лекции с целью исправления опечаток и фактологических ошибок;
- при возникновении вопросов у студентов выносить ответы на них в отдельную часть материала, не отделяя вопросы от материала (как это произошло бы при ответе на них на форуме) и делая ответы общедоступными (чего не происходит при ответе в частной беседе или при переписке). Это значительно снизило нагрузку на преподавателя, которому не приходилось по 10 раз объяснять одно и то же.

Интересные возможности использование MediaWiki дало в лабораторных работах. Благодаря фиксации времени внесенных исправлений удалось отразить посещаемость студентами лабораторных (при

начале лабораторных они должны записывать свой вариант в список в wiki).

Полученный текст лекции и методических указаний к лабораторным работам хорошо приспособлен для дальнейшей предпечатной обработки и печати.

В результате выбора MediaWiki были оперативно написаны общедоступные лекции и указания к лабораторным работам.

Для лабораторных работ использовалась открытая среда анализа данных Weka. Она состоит из библиотеки алгоритмов и нескольких графических интерфейсов к этой библиотеке, реализующих ее референсное использование в разных целях. Открытость ПО дало возможность:

- использовать реализованные в библиотеке алгоритмы как примеры в ходе лабораторных и лекций;
- использовать уже реализованные в библиотеке алгоритмы для обучения использования библиотечных функций для анализа данных;
- студентам реализовывать и применять различные алгоритмы к имеющимся у нихв данным не с нуля, а создавать короткие и четкие программы, написанные при помощи внутренних функций weka;
- научится самим создавать классификаторы и кластеризаторы, не отвлекаясь на различные технические нюансы, которые уместны к рассмотрению в курсе программирования, а не анализа данных, благодаря указаниям по написанию расширяющих среду weka классификаторов и их интеграции в среду weka.

Одним из непрямых результатов использования weka стало более легкое прогружение студентов в язык Java, на котором написана среда. Сначала они изучили среду, потом — API среды, использующий Java-подобный синтаксис, затем — при исправлении некоторых ошибок и защите лабораторных — слегка более глубоко перешли к языку, а при написании классификатора уже использовали сам язык Java. При этом примеры использования были у них перед глазами.

Выбор FLOSS позволил в коротчайшие сроки составить и провести новый сложный курс.

Литература

[1] Клекка У.Р. Олдендерфер М.С. Блешфилд Р.К. Ким Дж.-О, Мьюллер Ч.У. *Факторный, дискриминантный и кластерный анализ.* Финансы и статистика, 1989.

Е. Р. Алексеев

Донецк,
Донецкий национальный технический университет

Переподготовка преподавателей и сотрудников Донецкого национального технического университета на факультете повышения квалификации в рамках курса «Использование свободного программного обеспечения в учебном процессе»

Аннотация

В докладе представлен опыт чтения курса «Использование свободного программного обеспечения в учебном процессе» на факультете повышения квалификации в Донецком национальном техническом университете.

В конце 2007–2008 учебного года в Донецком национальном техническом университете (ДонНТУ) было принято решение о широком использовании свободного программного обеспечения в учебном процессе и научной деятельности. В связи с этим на факультете повышения квалификации были организованы курсы «Использование свободного программного обеспечения в учебном процессе» для преподавателей и сотрудников университета. Программа курсов предусматривает 54 часа аудиторных занятий (18 часов лекций и 36 часов лабораторных занятий) в течении 6 недель. По окончанию занятий слушатели защищают выпускную работу. Занятия проходят в специализированной аудитории кафедры «Вычислительная математика и программирование». На компьютерах в этой аудитории установлены операционные системы: Ubuntu 8.04.1 и ALT Linux 4.03 Lite. После успешного окончания курсов слушателям выдается свидетельство государственного образца.

Программа курсов включает в себя следующие основные разделы:

1. Знакомство со свободным программным обеспечением.
2. Использование ОС Linux в качестве рабочего стола (desktop).
3. Работа во всемирной сети Интернет под управлением ОС семейства Linux.
4. Свободно распространяемое офисное программное обеспечение.
5. Решение инженерных задач с использованием свободно распространяемого ПО.

На первых занятиях слушатели узнали о принципах и истории развития свободного программного обеспечения (СПО).

Изучение ОС Linux начинается со знакомства с файловой системой. Далее слушатели узнают о том, что из себя представляет дистрибутив современной ОС семейства Linux. Одна из лекций посвящена обзору современных дистрибутивов. На последующих занятиях слушатели знакомятся с установкой операционных систем Ubuntu Linux и ALT Linux Lite на ПК. Далее изучается настройка рабочего стола ОС, принципы установки и удаления программ.

Использование Интернета на компьютерах, работающих под управлением ОС ALT Linux (Ubuntu Linux), начинается с изучения особенностей настройки ПК для работы в локальной сети университета и во Всемирной Паутине. Далее слушатели знакомятся с программами для работы в Сети: Mozilla FileFox, Mozilla SeaMonkey, Mozilla Thunderbird, Pidgin, Kompozer и др.

При изучении офисного программного обеспечения преподаватели и сотрудники ДонНТУ знакомятся с офисным пакетом OpenOffice.org, графической программой dia, словарем StarDict, программами просмотра документов в форматах PDF и DjVu.

В завершении курса слушатели знакомятся со свободными программами для решения прикладных задач: математическими программами Scilab и Maxima, а также со средствами разработки программ на языках C++, Pascal, Basic. В качестве среды для изучения программирования на Basic слушатели используют OpenOffice.org Calc и Gambas. Для программирования на языке Pascal — IDE Free Pascal, Lazarus или Geany. При программировании C (C++) в качестве среды программирования также можно использовать Geany.

В качестве методической литературы на курсах используются лекции автора по основам работы в ОС Linux, программированию на Free Pascal, C++, использованию пакета Scilab (`http://www.teacher.`

dn-ua.com/), работы Ивана Хахаева по офисным и графическим пакетам (http://heap.altlinux.org/engine/AppsLinks), а также методическая литература, расположенная на сайте «Пакет программного обеспечения для образовательных учреждений России» (http://linux.armd.ru/ru/documentation/metod/).

По окончанию курсов каждый слушатель получает собранный в ДонНТУ DVD-диск «Свободное программное обеспечение для высших учебных заведений», на котором представлены:

- Windows-версии кроссплатформенных свободных программ;
- методическая литература;
- образ дистрибутива Runtu 3.

После выпуска каждой группы слушателей на каждом факультете появляется еще один специалист, который прошел обучение на курсах и может участвовать во внедрении свободного ПО в учебный процесс. Для широкого внедрения свободных программ в учебный процесс в Донецком национальном техническом университете в первую очередь предстоит решать проблему модернизации компьютеров.

Е. А. Чичкарев, К. Е. Чичкарев Мариуполь, Украина,
Приазовский государственный технический университет

Проект: S.A.G.E. http://www.sagemath.org

Интегрированный пакет математических расчетов S.A.G.E.: использование в преподавании

Аннотация

Пакет SAGE обеспечивает унифицированный интерфейс взаимодействия различных систем символьных и численных расчетов, что позволяет организрвать их совместное использование для решения разнообразных задач в области алгебры, дифференциального и интегрального исчисления, теории чисел, криптоанализа, теории групп, комбинаторики и других разделов математики. Наличие веб-интерфейса открывает широкие возможности для использования пакета в учебном процессе для организации дистанционного обучения, унификации лабораторного практикума по различным дисциплинам. Приведены примеры решения различных задач в среде SAGE, тест скорости выполнения ряда тестовых задач.

S.A.G.E. — это свободный пакет для математических расчетов. Он объединяет множество существующих свободных пакетов в объединенной платформе на Python. Конечная цель SAGE состоит в том, чтобы создать жизнеспособную, свободную, с открытым исходным кодом альтернативу проприетарным пакетам — Maple, Mathematica, Magma, MatLaB и др.

Основным приемуществом пакета SAGE является простая интегрируемость — используются стандартные инструменты Python, которые могут быть вызваны из любой программы (например, из редактора таблиц OpenOffice.org Calc вызвать макрос на Python), а также возможность вызова функций других распространненных математических пакетов, таких как MatLab, Mathematica, Maple, GNU Octave и др.

Программа SAGE может работать в веб-интерфейсе, что позволяет использовать ее в научных и учебных заведениях с малыми вычислительными мощностями на рабочих станциях, а также для дистанционного обучения. Sage позволяет решать задачи в области алгебры, дифференциального и интегрального исчисления, теории чисел, криптоанализа, теории групп, комбинаторики и других разделов математики. Для сравнения — цена лицензии популярных проприетарных математических программ Mathematica и Magma составляет примерно $2.5 тыс. и свыше $1 тыс. соответственно. Существуют версии SAGE для GNU/Linux (300 Mb) и Microsoft Windows (700 Mb), доступные для скачивания на сайте проекта (см. [1]). Программа доступна в виде веб-сервиса SAGE Notebook на сайте разработчиков [2] или на сайте Сообщества математического моделирования [3].

SAGE доступен для различных программных и аппаратных платформ: Windows, Mac, и Linux (32-и 64-битных).

Для отображения графиков в собственно SAGE используется библиотека matplotlib. Однако при обращении к Maxima или Octave из SAGE можно сформировать иллюстрации и при помощи интерфейсов gnuplot или openmath.

При работе с SAGE реализовать нелобходимый алгоритм можно неколькимии способами:

- в виде скрипта на макроязыке SAGE;
- в виде скрипта на Python (с использованием библиотеки SAGE);
- в виде функции C/C++ (с возвратом результатов SAGE с использованием cython);

- в виде скрипта cython;
- в виде последовательности команд или функции на языке одного из пакетов, для которых имеетмя интерфейс SAGE (Maxima, Octave, GAP и др.).

Преимущества Python в качестве средства интеграции систем компьютерной математики:

- хорошо поддерживается сохранение на диск почти любых объектов;
- поддержка автоматического извлечения документации из исходного текста, а также автоматического тестирования примеров;
- наличие в реализации Python автоматического управления памятью и «сборкой мусора»;
- наличие большого числа пакетов, написанных на Python, реализующих методы численного анализа и линейной алгебры, 2D и 3D-визуализацию; распределенные вычисления и работу в сети, поддержку баз данных и др.;
- переносимость Python на различные платформы;
- эффективное управление исключениями и ошибками;
- наличие эффективных отдадчика и профайлера кода на Python.

Таким образом, в рамках SAGE не столько создаются новые функциональные возможности, сколько обеспечивается ясный, систематический и последовательный путь получения доступа к большому количеству алгоритмов символьной и численной математики, в последовательном построении математического каркаса.

Однако SAGE — довольно большой по объему пакет, построенный достаточно специфично: в состав дистрибутива SAGE входят целиком ряд стандартных пакетов (NumPy, SciPy, R, Maxima и др.). Поэтому компиляция SAGE из исходных текстов может быть достаточно длительным мероприятием. Кроме того, версии компонентов SAGE могут не совпадать с установленными в операционной системе (например, в стандартные репозитории Ubuntu 8.10 входит Maxima 5.13.0, в состав SAGE 3.2.3 — Maxima 5.16.3).

Анализ скорости выполнения различных тестов и вычислительных задач различной сложности показал, что их выполнение в среде SAGE или при помощи индивидуально установленных пакетов существенно не различается. Существенным достоинством SAGE по

сравнению с использованием индивидуальных пакетов являются достаточно широкие возможности обмена данными между различными приложениями (например Maxima, Octave и R).

В Приазовском государственном техническом университете пакет SAGE опробован для выполнения семестровых заданий по ряду математических курсов (математический анализ, функциональный анализ), части лабораторных и курсовых работ по курсам численных методов и моделирования на ЭВМ, идентификации и моделирования. Использование SAGE в перспективе позволит организовать сквозной лабораторный практикум по ряду дисциплин, читающихся последовательно. Кроме того, наличие веб-интерфейса, возможности построения анимированных иллюстраций, возмож ности автоматизированного и пошагового решения задач из различных разделов математики делает SAGE привлекательным средством дистанционного обучения.

Литература

[1] http://www.sagemath.org

[2] http://www.sagenb.org

[3] http://www.sagemath.ru:8000

А. Н. Гороховский Донецк,
Донецкий национальный технический университет

Проект: Использование СПО в ВУЗах
http://peooc.donntu.edu.ua/olimp/

Опыт программно-информационного обеспечения межвузовских олимпиад по дисциплине «Экология»

Аннотация

Рассматриваются возможности СПО при проведении дистанционных межвузовских интернет-олимпиад.

Разработка выполнена и применяется в течение 2003–2008 г. г. на базе дистрибутивов ОС ALT Linux (Master, Desktop) с использованием пакетов свободно-распространяемых программ — Web-сервера Apache и языка Perl.

В настоящее время наблюдается стремительный рост интереса образовательных заведений к новой форме обучения — дистанционной. Активизации этого процесса в полной мере способствует развитие web-технологий и Internet [1, 2, 3].

Ярким примером применения технологий Internet в учебном процессе является проведение тестирования и дистанционных олимпиад, которые призваны стимулировать активность, инициативность при проверке знаний, умений и полученных навыков по изучаемому предмету.

Организационная структура олимпиад

На базе кафедры «Прикладная экология и охрана окружающей среды» Донецкого национального технического университета в соответствии с приказом МОН в течение 2004–2006 г. г. проводились Международные (с участием студентов из России), а в 2003–2008 г. г. — внутривузовские олимпиады по дисциплине «Экология» (`http://peooc.donntu.edu.ua/olimp/`).

Организационный комитет, методическая комиссия и жюри формируются из ведущего профессорско-преподавательского состава университета и различных ВУЗов, которые обеспечивают экспертный уровень подготовки тестовых заданий олимпиады и делают ее максимально открытой.

Тестовые задания и условия проведения олимпиады ежегодно изменяются, а результаты проверки работ и их анализ оформляются в виде итоговых таблиц, которые размещаются для ознакомления.

Традиционно олимпиада проводится в два тура, т. е. 1-й тур — отборочный. Его цель — проверка уровня общей эрудиции и отсев заведомо слабых участников. Победители 1-го тура (на основе проходного количества правильных ответов-баллов) приглашаются к участию в более сложном — 2-ом туре олимпиады.

В итоговой таблице окончательный рейтинг участников составляется в соответствии с результатами двух туров.

Основными недостатками «бумажного» проведения олимпиад (до 2003 г.) были:

1. отсутствие одновременной поддержки нескольких языков заданий (русский, украинский) для многонациональной аудитории;
2. человеческий фактор во время проверки работ;

3. низкая оперативность обработки ответов участников (100–110 чел.) и ограниченный анализ при подведении итогов.

Все это, безусловно, должно было сказываться и на объективности конечных итогов олимпиад.

Понимая всю ответственность этих мероприятий, было принято решение об их проведении в виде компьютерного тестирования по схеме «клиент — сервер». В качестве платформы для сервера была выбрана ОС ALT Linux Master.

Концептуальная схема проведения интернет-олимпиад

Оргкомитетом участники олимпиад (≈ 105–110 чел.) разделялись на группы, каждой группе определялось время проведения тестирования в 7 компьютерных аудиториях университета с ПК, подключенными к сети Internet (через сеть университета) и работающих под управлением различных ОС (Windows 95–98, ALT Linux). Доступ к серверу (ALT Linux Master, Web-сервер Apache) происходил посредством браузера с графическим интерфейсом (Opera).

В дальнейшем тестирование проходило по схеме «клиент — сервер»:

1. С рабочего места через html-форму участник регистрируется на Web-сервере `http://peooc.donntu.edu.ua/olimp/`: задает ФИО, выбирает Область, ВУЗ, язык общения.

2. Каждый участник в ответ получает html-формы со случайно выбранной последовательностью открытых тестовых заданий. Общее количество заданий (вопросов) — 100, общее время тестирования — 100 минут.

3. Результаты проверки ответов автоматически отображаются на мониторе ПК участника олимпиады.

Функциональные возможности ресурса проведения олимпиад на Web-сервере реализованы с помощью оригинального программного кода (язык Perl), защищенных баз данных и дружелюбного пользовательского интерфейса.

Количество принявших участие в олимпиадах постоянно растет (табл. 1), поэтому данный проект проведения дистанционных олимпиад планируется продолжать.

Таблица 1: Статистика состава участников Всеукраинских студенческих олимпиад 2004–2006 г. г.

Год	Областей Украины	ВУЗов	Выбор языка (Ru/Ua)	Всего участников
2004	19	39	65 / 39	104
2005	21	51	68 / 41	109
2006	21	45	56 / 54	110

Литература

[1] Научно-практический журнал «Открытое образование» `http://www.lib.ru/CTOTOR/BRUKS/mithsoftware.txt`

[2] Журнал «Вопросы Интернет образования» `http://center.fio.ru/vio/vio_07/cd_site/Articles/archive.htm`

[3] Журнал «Компьютерные инструменты в образовании» `http://www.ipo.spb.ru/journal/`

М. Э. Кушнир Москва, ГОУ Гимназия №45

Проект: Электронный классный журнал РУЖЭЛЬ www.rujel.net

Психология инициативной разработки

Аннотация

Чтобы успешно развивать СПО, нужно учитывать мотивацию как пользователей, так и разработчиков. Для этого полезно проанализировать нетипичные для технарей аспекты — психологические.

Меня будут интересовать 4 типичные роли:

- пользователь-ламер;
- продвинутый пользователь;
- разработчик проприетарного ПО;

- разработчик СПО.

Основу рынка ПО составляют пользователи-ламеры, которые ничего не понимают и не хотят понимать в компьютерных нюансах. Их интересует максимально удобное и надежное ПО за приемлемые деньги и время на освоение. Конечно, им больше нравится бесплатный сыр, но они знают о мышеловке, поэтому предпочитают расплатиться. Некоторые расплачиваются знакомством с продвинутыми пользователями — их можно не считать ламерами в нашем обсуждении. Остальным спокойнее купить такое ПО, которое требует от них минимальных издержек по трудозатратам. Поэтому ламер — основа проприетарного ПО.

Продвинутый пользователь — бесплатный сотрудник маркетинга тех вендоров, которых он предпочитает. Он является референтным советчиком для окружающих его ламеров и часто весьма активно агитирует их применять понравившееся ему ПО. Популяция таких пользователей более эффективна в рекламе, чем штатные сотрудники, но их пропагандистская деятельность остается за кадром. Именно такой пользователь важен миру СПО как популяризатор.

Теперь рассмотрим разработчика СПО.

При этом я хочу сконцентрироваться на разработках среднего уровня сложности, т. е. тех, которые:

- уже не маленькие поделки, которые могут оставаться в качестве хобби разработчика и не отнимать у него «хлеб насущный»;
- и еще не большие знаковые продукты, за которыми стоят монстры СПО, конкурирующие с солидными проприетарными игроками, часто при поддержке других коммерческих фирм, выпускающих не менее проприетарное ПО.

Ниша «среднего» ПО более проблематична. Ее продукты или кормят, или должны быть оставлены без развития. Если разработка обеспечивает решение общеупотребительных задач, вероятность внимания к ней коллег и потребителей существенно выше, чем в случае специфических направлений. Соответственно, выше вероятность выживания и развития проекта. Однако именно специфические направления определяют корпоративный рынок.

Участников конференции больше занимает ниша специфического ПО. Как минимум, это определяет тематика — образование. Интерес к этому направлению вызван сегодня, прежде всего, тем, что «запахло деньгами». Это видно по массовому предложению известных (и не

очень) фирм, да и разработчики СПО не остаются без надежд на «свою долю пирога».

Наша разработка — электронный классный журнал РУЖЭЛЬ — тоже попадает в разряд специфического ПО.

Обычно разработка начинается как попытка решить конкретную проблему в сфере деятельности специалиста. Он на основе своих профессиональных навыков и подручных средств ищет решение задачи. При этом подбирается некоторый набор готовых средств, а самостоятельно создается только небольшой фрагмент целостного решения.

Достойно решив задачу, разработчик оказывается под давлением собственного тщеславия — он хочет получить более весомое подтверждение своего профессионализма. Лучшее подтверждение — признание более широкого круга пользователей, чем предполагалось изначально: внешних коллег-программистов и сторонних ламеров. Хочу обратить внимание на возникновение в этот момент серьезного конфликта между двумя совершенно разными задачами:

- программистской, которая ставилась для одной конкретной ситуации;
- маркетинговой, в которую превратилась исходная — распространение решения для других.

Стоит подчеркнуть, что программист, как правило, совсем не специалист в маркетинге.

Подробнее этот конфликт анализируется в полной версии статьи по адресу `http://sites.google.com/site/kmedwru/thinks-1/2009-psyspo`.

Вывод

Для разрешения проблемы развития СПО, на мой взгляд, необходим активный маркетинговый узел, который мог бы составить достойную конкуренцию стабильности и маркетинговой квалификации коммерческих фирм. Такая структура должна быть центром формирования банка СПО, его интеграции, координации, популяризации, обучения, поддержки... маркетинга во всех его проявлениях.

Создание и достойное существование таких центров — серьезная проблема, в том числе, психологическая, т. к. разработчики — люди заведомо великие, а менеджеры такого центра — «хапуги и крючкотворы, жирующие на теле (или деле) истинных творцов».

А. В. Щеткина Донецк,
Донецкий национальный технический университет

Проект: Свободное программное обеспечение для технических ВУЗов
http://allopensoft.ucoz.ru

Свободное программное обеспечение для технических ВУЗов

Аннотация

Целью работы является создание образовательного ресурса, который посвящен свободному программному обеспечению для преподавателей, студентов и специалистов в области естественных и точных наук.

Перед пользователем, осваивающим свободное программное обеспечение (ПО), стоит проблема поиска информации. В Интернете существует множество разнообразных русскоязычных сайтов, посвященных отдельным свободным программам. Целью работы является разработка образовательного ресурса, посвященного свободному программному обеспечению. Мы попытались на одном сайте собрать информацию, о свободном ПО, которой могли бы пользоваться преподаватели, студенты, а также специалисты в области естественных и точных наук.

Ресурс разработан на базе DVD-диска, который был собран в Донецком национальном техническом университете на кафедре «Вычислительная математика и программирование». На сайте представлены следующие разделы:

1. Основы работы на ПК под управлением ОС Ubuntu Linux;
2. Офисные приложения;
3. Графические программы;
4. Приложения для работы в Интернете;
5. Математические программы;
6. Средства разработки программ.

Каждая программа из этих разделов содержит краткое описание, ссылку на официальный сайт и страницу загрузки, а также документацию с указанием источника информации.

В разделе, посвященном офисным приложениям, представлена информация о таких программах, как Abiword, Gnumeric, OpenOffice.org, StarDict, TEX. В разделе «Графические приложения» находится информация о наиболее используемых техническими специалистами свободных программах: Dia, Gimp, а также о программе Scribus. Среди программ для работы в Интернете основное внимание уделяется программам семейства Mozilla (Firefox, Thunderbird и др.) и средствам мгновенного общения в Интернете — Pidgin и Miranda. Среди математических программ сайт содержит информацию о таких пакетах, как Maxima, Scilab, Freemat. Раздел «Средства разработки программ» поможет пользователям в освоении компилятора gcc, написании программ на языке Free Pascal, использовании Lazarus.

Разработанный сайт поможет пользователям быстро найти необходимую информацию о свободном ПО. Сайт только начинает развиваться, мы приглашаем всех желающих к наполнению сайта информацией.

Е. В. Андропова, Т. Н. Губина, М. А. Губин Елец,
Елецкий государственный университет им. И. А. Бунина

Проект: Центр свободного программного обеспечения
`http://www.fosscenter.elsu.ru`

Образовательное пространство и свободное программное обеспечение

Наибольшее распространение в нашей стране получили операционные системы семейства Windows и ориентированные под эту систему прикладные программные продукты: MS Office, PhotoShop, CorelDraw и др. Но все это программное обеспечение относится к проприетарному и требует от пользователей немалых денежных затрат на покупку программ и соответствующих лицензий.

Учитывая экономические, технические и юридические аспекты использования лицензионного программного обеспечения, в 2008 году руководством Елецкого государственного университета им. И. А. Бунина было принято решение о создании Центра свободного программного обеспечения (СПО), целью которого является апро-

бация и постепенный перевод учебного процесса ВУЗа на свободное программное обеспечение.

Руководствуясь этой целью и работая на перспективу, сотрудники Центра СПО осуществляют следующую работу:

- организация и проведение научно-практических семинаров;
- выступления на конференциях;
- разработка учебно-методических пособий;
- участие в профильных выставках и образовательных форумах;
- организация и проведение мастер-классов.

В современных условиях рынка программного обеспечения считаем необходимым показать тенденции в его развитии студентам физико-математического факультета. Так, в текущем учебном году некоторые дисциплины информационного профиля на физико–математических специальностях ведутся с использованием свободного программного обеспечения: OpenOffice.org, Maxima, Scilab, Gimp, Free Pascal, Lazarus. Кроме того, студенты выполняют курсовые и выпускные квалификационные работы с использованием свободного программного обеспечения. Таким образом, у будущих специалистов вырабатывается понимание того, что современные IT-технологии могут строиться не только на основе проприетарного программного обеспечения.

Сотрудниками Центра СПО составлен перечень программных продуктов, распространяющихся под лицензией GPL, способных заменить проприетарное программное обеспечение, используемое для обеспечения учебного процесса в ВУЗе, при этом не нарушая требований ГОС ВПО.

Также немаловажным является применение свободного программного обеспечения в учебном процессе средней школы, т. к. в дальнейшем знания, полученные во время занятий, могут понадобиться ученикам при обучении в ВУЗе и работе в организациях и предприятиях с любым видом деятельности.

На основании письма из Рособразования «Об использовании ПСПО для общеобразовательных учреждений Российской Федерации» от 17 июня 2008 №15–51–450/01–09, в котором всем желающим было предложено «принять участие в апробации ПСПО и выработке рекомендаций по использованию ПСПО в УО», был заключен договор о сотрудничестве с гимназией №11 г. Ельца Липецкой области на 2008–2009 учебный год.

Суть данного проекта — сотрудничество Центра СПО и школы с целями:

- отработки процедуры внедрения ПСПО в средних учебных заведениях;
- проведения оценки целесообразности, успешности применения и степени готовности программного состава дистрибутива ПСПО к использованию в учебном процессе;
- разработки и внедрения учебного курса «Информационные технологии на базе ПСПО» в учебный процесс школы.

Однако в рамках одного Центра справиться с требуемым объемом работ очень сложно. Считаем необходимым привлекать другие заинтересованные учебные заведения и создавать региональные Центры СПО.

Таким образом, требуется развитие и внедрение в каждой ступени образования нескольких методик перехода и построения образовательного процесса на базе свободного программного обеспечения с использованием разных программных продуктов. Необходимо ускорение темпов обновления стандартов, планов и методик, организации курсов переподготовки специалистов в области свободного программного обеспечения, проведение конкурсов методических разработок, научно-квалификационных работ в области применения СПО.

М. О. Карташов, М. А. Губин Липецк, Елец,
МОУ СОШ №49 г.Липецк, Центр СПО ЕГУ им.И.А.Бунина г.Елец

Проект: Центр СПО ЕГУ им. И.А.Бунина, Lipetsk *nix Association
`http://www.fosscenter.elsu.ru`, `http://lna.org.ru/`

Об опыте быстрого развертывания системы ALT Linux в компьютерном классе

Аннотация

Рассматривается один из методов быстрой установки и настройки ОС ALT Linux Junior на школьные компьютеры с предустановленной ОС Windows. Необходимым условием являются: максимальная экономия времени инженера компьютерного класса и сохранение всех настроек и документов учителей и учеников школы.

По планам Правительства РФ, в 2009 году предполагается внедрение пакета свободного программного обеспечения во всех образовательных учреждениях России с целью создания условий для альтернативного выбора программного обеспечения.

В рамках приоритетного национального проекта «Образование» в апробации ПСПО изъявили желание участвовать и некоторые школы Липецкой области, например МОУ СОШ №49 г. Липецка и гимназия №11 г. Ельца. После получения пакета СПО встал вопрос развертывания системы Linux на компьютерах общеобразовательных учреждений в максимально сжатые сроки.

На школьных компьютерах уже был установлен комплект СБПО «Первая Помощь» и требовалось сохранить уже установленную систему неповрежденной. Потребовалось найти оптимальный способ установки, занимающий как можно меньше времени. Один из таких способов — клонирование системы.

Сам способ состоит из 2-х этапов:

1. настройка одного эталонного экземпляра полностью рабочей системы;
2. тиражирование этого экземпляра на рабочие станции.

Для начальной настройки использовалась виртуальная машина SUN VirtualBox 2.0.6 (`http://www.virtualbox.org`). На нее был установлен дистрибутив ALT Linux School Junior 4.0 с графической системой KDE.

Настройка сводилась к созданию 2-х учетных записей — учительской и ученической, настройке внешнего вида, меню для учеников по принципу «ничего лишнего» и написанию сценария для автоматического восстановления всех настроек ученика «по умолчанию» при перезапуске машины.

Тиражирование происходило следующим образом.

Сначала образы корневого и домашнего разделов были записаны на внешний носитель (жесткий диск) командой

```
dd if=/dev/hda1 of=/mnt/usb/root.img bs=1M
dd if=/dev/hda2 of=/mnt/usb/home.img bs=1M
```

Затем на рабочих станциях посредством программы Gparted, входящей в состав дистрибутива, создавалось 3 раздела в свободном месте — для корневого и домашнего разделов, и для swap. После чего к каждой машине подключался внешний жесткий диск, и выполнялась

команда копирования из файлов-образов на жесткий диск рабочей станции:

```
dd if=/mnt/custom/root.img of=/dev/sda2 bs=1M
dd if=/mnt/custom/home.img of=/dev/sda7 bs=1M
```

Оставалось сделать еще 2 шага: настроить монтирование разделов и установить загрузчик.

Для настройки монтирования вносились поправки в файл `/etc/fstab` — для каждого windows–раздела необходимо внести изменения в строки:

```
/dev/hda1 /mnt/win/sys ntfs-3g defaults,umask=022,uid=0 0 0
/dev/hda5 /mnt/win/doc ntfs-3g defaults,umask=022,uid=0 0 0
```

Маска `umask=022` требуется, чтобы ограничить пользователям доступ к записи на разделы Windows, но при этом оставить его у администратора.

Для настройки загрузчика проводились следующие действия:

- правка файла `/etc/lilo.conf`;
- установка загрузчика командой `lilo`.

В результате выполнения всех действий, после перезагрузки получается рабочая система с выбором из 2-х операционных систем: Windows и Linux.

И. В. Воронин, ИТ ИПЛИТ РАН

Использование СПО на курсах системы РКЦ-ММЦ

Аннотация

Комплексный Процесс Модернизации Образования (КПМО) подразумевает использование системы РКЦ-ММЦ. В докладе освещается, как используется СПО на курсах повышения квалификации преподавателей, трудности при организации и проведении подобных курсов, предлагаются решения по внедрению СПО в общеобразовательный процесс, обсуждается мотивация участников процесса внедрения, выдвигаются предложения.

Современные тенденции развития системы образования выдвигают определенные требования, или можно сказать вызовы, на которые должно отреагировать современное общество, чтобы создавать постиндустриальную — инновационную — экономику. Присоединение России к Булонскому процессу подразумевает комплексную модернизацию образования КПМО. Соответственно требованиям руководящих органов федерального министерства образования выстраивается система РКЦ-ММЦ в субъектах федерации. Таким образом, в Московской области создана сеть ММЦ, и один из них представлен в Шатурском районе — на базе МОУ СОШ № 2 г. Шатуры. ММЦ Шатуры функционирует уже более 3-х лет. За это время на его базе прошло обучение и переподготовку более 400 учителей школ Шатурского района. В данном докладе хочется осветить положительные моменты и некоторые недостатки системы ММЦ с точки зрения руководителя и преподавателя такого центра. Заметим сразу, что характерной особенностью вновь созданной системы ММЦ является лавинообразное увеличение отчетности перед вышестоящими органами, причем в бумажном виде.

Автор данного доклада является руководителем отдела ИТ института ИПЛИТ РАН и преподает в ММЦ в свободное от основной работы время. В основной профессиональной деятельности автор использует ОС ALT Linux, поэтому и обучение преподавателей школ производится тоже в Linux. Исторически так сложилось, что, когда в школу пришел класс с ПК, то не было возможности активировать предустановленную ОС Windows, и руководство школы согласилось на установку на всех 13 рабочих местах ОС Linux. Таким образом, можно заметить, что использование ОС Linux в системе ММЦ носит случайный, а не закономерный характер.

Курс по повышению квалификации преподавателей средней школы строится на получении навыков в работе с текстовым процессором, таблицами, презентациями, поиске информации в Интернете, начальных азов по работе с графикой. Для этого используется OpenOffice.org и браузер FireFox. Курс составляет 72 часа.

Слушателями курсов являются, в основном, женщины за 50 лет — преподаватели разных дисциплин в средней школе, как правило, не имеющие никакого опыта даже с включением и выключением системного блока. Мотивацией слушателей к обучению является получение и подтверждение категории, стремление не оказаться на пенсии и без работы. Большая протяженность Шатурского района по площади за-

трудняет поездки преподавателей для традиционного повышения квалификации в Москву, поэтому наличие курсов ММЦ в районном городе воспринимается положительно.

Как правило, трудностей с освоением работы в среде OpenOffice.org не возникает. Понять, что такое файл, где его найти в папке, как сохранить файл под нужным именем, получается после нескольких занятий. Ориентация в файловой среде Linux не вызывает затруднений. Проблемы конвертации форматов сохраненных документов DOC или ODT после получения собственного опыта решаются легко.

После прохождения курсов слушатели получают документ, позволяющий им заявлять на повышение своих баллов при формировании таблиц к стимулированию в новой системе финансирования оплаты труда НФОТ.

Необходимость использования ИКТ в образовательном процессе очевидна всем участникам этого процесса, а вот использование СПО на курсах ММЦ, к сожалению, воспринимается неоднозначно. Преподаватели, получившие навыки по составлению презентаций в OpenOffice.org на курсах, могут испытывать некоторые затруднения в школах при работе с продуктами от компании Microsoft. Пока они не научатся хорошо ориентироваться в различных платформах, необходимо оказывать им консультационную помощь. Они должны иметь возможность задать вопрос хотя бы по телефону.

Кроме того, дорогостоящее ИКТ-оборудование, установленное в школах по нацпроекту «Образование», зачастую не может быть задействовано в учебном процессе, например, из-за банального неумения подключить проектор к выходу видеоплаты. Требовать от преподавателей-женщин 50-летнего возраста необходимой квалификации, не связанной с их основой деятельностью, представляется неразумным.

Вполне логичным для снятия проблем с использованием ИКТ-оборудования в школах было бы решение по организации центров компетенции на местах. Например, по одному на район. Штат таких центров должен быть минимален — два, максимум три сотрудника. Как самостоятельное юридическое лицо такой центр компетенции должен финансироваться из местного бюджета Управления образованием, например, за счет платных услуг.

В сфере его компетенции должно быть оказание услуг по консультации, развертывании и использования СПО в образовательном

процессе, поддержка и настройка локальных сетей в школах, тестирование и мелкий ремонт оборудования, поддержка пользователей по использованию ИКТ. Совершено очевидно, что без четкой команды из министерства образования субъекта федерации, а им от министерства РФ, в местном бюджете такая структура финансироваться не будет.

Таким образом, увеличение числа преподавателей, имеющих навыки в работе с продуктами СПО, позволит нам повсеместно развивать внедрение СПО в образовательный процесс, сделать процесс обучения более наглядным и ярким, привлекать на этой основе к творчеству все большее число учеников. Тем самым будет усиленно инновационно-инженерная составляющая в образовательном процессе.

И. В. Зайцев Санкт-Петербург, ГОУ ВПО СПбТЭИ

О преподавании курса по алгоритмизации на основе языка JavaScript и открытого ПО

Аннотация

На основе опыта преподавания курса по алгоритмизации с использованием свободных программных средств (ALT Linux & SeaMonkey) рассматриваются преимущества перехода с устаревших языков BASIC и Pascal на использование современного широко распространенного языка JavaScript.

В российских ВУЗах обучение программированию на специальностях, напрямую не связанных с информатикой, часто осуществляется на базе таких языков как BASIC и Pascal, в силу сложившейся традиции. Однако в настоящее время использование этих языков становится нецелесообразным по следующим причинам:

- Эти же языки часто используются и в школьных курсах информатики, из-за чего у части студентов 1-го курса (хорошо освоивших курс информатики в школе) формируется пренебрежительное отношение к преподаваемой дисциплине.
- Практическая ценность данных языков программирования крайне мала, поэтому их знание не является преимуществом при трудоустройстве, что также ухудшает отношение студентов к процессу обучения.

- Ввиду малой применяемости практически не выходит новой литературы, посвященной этим языкам программирования.

В качестве замены традиционным языкам, применяющимся для обучения программированию, в Санкт-Петербургском торгово-экономическом институте ряд групп изучают алгоритмизацию на основе современного и широко распространенного языка JavaScript. В процессе обучения используется ПО с открытыми исходными кодами в соответствии с генеральным направлением развития информационной подсистемы ВУЗа.

Язык JavaScript — это новый (в сравнении с языками Basic и Pascal) язык программирования, ставший весьма востребованным в настоящее время благодаря широкому распространению всемирной сети Интернет, где он используется в web-приложениях на стороне клиента для увеличения функциональности интерфейса и удобства взаимодействия с пользователями, а также в некоторых случаях и на стороне сервера.

Этот язык также вполне подходит для обучения основам алгоритмизации, причем его использование имеет следующие преимущества:

- Для выполнения упражнений по JavaScript нет необходимости устанавливать никакого дополнительного программного обеспечения ни в Windows (где можно использовать Internet Explorer и Notepad), ни в Linux (SeaMonkey и Kate). Это позволяет студентам самостоятельно выполнять задания на любом компьютере в дополнение к предусмотренным занятиям в компьютерных классах.
- Полученные начальные знания по программированию на JavaScript полезны в случае дальнейшего обучения студентов созданию Web-страниц и построению Web-интерфейса.
- Язык JavaScript поддерживается как один из языков для написания макросов в бесплатном офисном пакете OpenOffice.org (который широко используется в учебном процессе в СПбТЭИ).
- Язык JavaScript позволяет познакомить студентов с основами объектно-ориентированного подхода, играющего ключевую роль в современном программировании. Такое знакомство полезно для последующего углубления знаний в различных направлениях современной информатики.
- Язык JavaScript изначально создавался для быстрого написания небольших скриптов, а учебные задания студентов некомпью-

терных специальностей, по сути, как раз и являются небольшими скриптами;

- JavaScript имеет большую практическую ценность как средство, используемое при создании сайтов компаний (способность выпускника ВУЗа обновлять сайт компании во многих случаях может оказаться дополнительным преимуществом при приеме на работу).
- Благодаря большому сходству синтаксиса JavaScript с синтаксисом наиболее распространенных языков программирования Java и C++, для одаренных студентов облегчается переход к профессиональной разработке программного обеспечения, в том числе участие в open-source проектах.

Ограничения, накладываемые на функциональность сценариев JavaScript встроенными в браузеры интерпретаторами в целях безопасности (например запрет на чтение/запись произвольных файлов), никак не влияют на возможность выполнения заданий по курсу алгоритмизации в том объеме, в котором они выполнялись на языках Basic и Pascal ранее. Кроме того, способным студентам, желающим использовать полученные навыки программирования для решения учебных задач по другим дисциплинам, можно показать способы преодоления этих ограничений. Одним из вариантов является использование интерпретатора на open-source движке Rhino, включенного в состав Java Developer Kit (JDK) компании Sun MicroSystems (утилита jrunscript), предоставляющем огромные возможности для творчества и исследования, поскольку в нем обеспечивается доступ к библиотекам классов мощного универсального языка Java. Работа с JDK 6 с постепенным включением в свои скрипты экземпляров классов Java может стать для способных к информатике студентов удобным и эффективным путем в профессиональное программирование.

Таким образом, JavaScript, с одной стороны, представляется удобным современным языком для знакомства студентов некомпьютерных специальностей с основами программирования (а также с основами Интернет-технологий), а с другой стороны, оказывается весьма полезным для тех студентов, которые хотят и могут продолжить освоение программирования на более глубоком уровне.

А. Ю. Лагунов Архангельск,
Поморский государственный университет имени М. В. Ломоносова

Проект: План-конспекты уроков по элективному курсу «Технология объектно-ориентированного программирования на языке Basic в среде GAMBAS» http://freecode.pspo.perm.ru/348/index.html

Выбор среды разработки для обучения студентов программированию на языке JAVA

Аннотация

Важным элементом использования компьютеров в учебном процессе является программирование. В данной работе проводится обзор возможностей перехода на кроссплатформенные среды программирования с открытым исходным кодом (Open Source) при обучении программированию на языке Java.

В настоящее время происходит достаточно активный переход с проприоритетного программного обеспечения на ПО с открытым исходным кодом (Open Source). При подготовке на физическом факультете ПГУ имени М. В. Ломоносова инженеров по специальности 230201 «Информационные системы и технологии» (специализация «Информационные системы и технологии в бизнесе») необходимо учитывать тот факт, что выпускаемые нами специалисты чаще всего работают на небольших предприятиях и в государственных организациях. Из-за недостатка финансирования именно эти предприятия чаще всего переходят с проприоритетного ПО на ПО с открытым исходным кодом, которое во многих случаях является свободным. Выпускаемый ВУЗом специалист должен быть готов предложить предприятию проект по переходу на свободное ПО, поэтому в процессе обучения студент должен активно работать с проектами, использующими ПО с открытыми исходными текстами. Мы используем при обучению программированию язык Java, данный язык может быть использован на различных платформах. Мы являемся сторонниками кроссплатформенности в обучении: настоящая свобода появляется тогда, когда студент изучает различные операционные системы, а затем сам выбирает для себя систему, на которой будет работать в дальнейшем. Сформулируем требования, которым должна удовлетворять среда для обучения программированию:

- Кроссплатформенность среды разработки.
- Наличие удобной среды разработки как важнейшего элемента современного программирования.
- Среда разработки должна быть с открытым исходным кодом.
- Простота использования при обучении программированию.
- Возможность поэтапного перехода к более профессиональному уровню программирования с использованием того же языка программирования и той же среды разработки.
- Низкие требования к системным ресурсам.

Согласно вышеперечисленным требованиям, мы отобрали две среды разработки: Eclipse (`http://www.eclipse.org`) и NetBeans (`http://www.netbeans.org`).

Основные возможности NetBeans:

- Поддерживает языки Java, C/C++, JavaScript, HTML, CSS, Ruby;
- Развитый редактор графического пользовательского интерфейса (GUI) из сред с открытым исходным кодом;
- Хороший отладчик;
- Приятный редактор форм;
- Поддержка UML-проектирования;
- Имеется возможность подключения модулей (плагинов) от третьих производителей;
- Оперативная поддержка новых версий Java;
- Русскоязычная помощь в виде учебных карт на сайте производителя.

Недостатки NetBeans:

- Англоязычный интерфейс (возникают проблемы при обучении студентов, изучающих отличные от английского иностранные языки);
- Высокие требования к ресурсам, как следствие — замедление работы компьютера;
- Относительно медленная компиляция;
- Сохранение всех последних проектов на вкладках (при первичном обучении студенты плохо ориентируются в проекте).

Основные возможности Eclipse:

- Поддерживает разные языки программирования путем подключения различных плагинов (C/C++, Java, PHP, COBOL и т. д.);
- Хороший отладчик;
- Приятный, удобный интерфейс;
- Имеет возможность подключения модулей (плагинов) от третьих производителей. Причем уже существует большое количество таких модулей: как свободных, так и коммерческих;
- Возможность сборки собственного набора требуемых плагинов;
- Достаточно высокая скорость работы по сравнению с NetBeans.

Недостатки Eclipse:

- Платность некоторых модулей;
- Недостаточно хорошая совместимость модулей от разных разработчиков для разных версий Eclipse;
- Сложность при поддержке UML-проектирования.

Выводы:

- Среда NetBeans представляет собой полный комплекс средств для обучения основам программирования, мы рекомендуем данную среду в начале обучения программированию.
- Среда Eclipse прекрасно подходит для обучения студентов на курсах по выбору и при выполнении курсовых и дипломных проектов.

А. М. Дербень Красноярск, Сибирский государственный аэрокосмический университет им. ак. М. Ф. Решетнева

Проект: eXPine http://www.gnome-look.org/content/show.php/eXPine?content=77119

Внедрение Linux для обеспечения системы управления ВУЗом

Аннотация

В процессе перевода пользователей с ОС сеймества Windows на Linux существуют проблемы субъективного восприятия некоторых элементов оформления окон, рабочего стола, кнопок и т. д. В настоящем докладе предлагается решение, в результате внедрения которого срок адаптации пользователей при переходе сокращается с 1 мес. до 3–4 дней.

В последние два года в России проявляется активный интерес к Desktop-решениям на базе открытых систем, в т. ч. и Linux. Появились первые по-настоящему удачные внедрения, но, к сожалению, есть также и негативный опыт. В чем заключается причина неудач? Имеющийся функционал практически ничем не уступает Windows-системам, скорость работы и устойчивость (как и у всех Unix-подобных систем) также на высоте, графический интерфейс удобнее, к вирусам полная невосприимчивость. И все-таки рядовой пользователь с явной неохотой переходит на Linux. Конечно, сказывается и некоторая недоработка Linux-программ и эмуляторов, но основная причина лежит в психологической плоскости.

Как известно из теории маркетинга, только около 15% людей являются новаторами, 70% — «умеренные» консерваторы и 15% — «безнадежные» консерваторы, которые при любых условиях остаются верны даже устаревшему продукту. Таким образом, главной задачей продвижения продуктов на базе Linux является достижение успеха среди «умеренных» консерваторов, составляющих подавляющее большинство. Пользователи должны получить знакомый интерфейс, привычную и интуитивно понятную графическую «обёртку» функционала ОС. Для Linux существует несколько графических подсистем, но основные усилия разработчиков сконцентрированы на KDE и GNOME.

Как показывает практика, при выборе графической подсистемы следует рассматривать не только внешнее сходство с Windows-системами, но также и сходство на уровне т. н. идеологии управления рабочим столом.

При внедрении Desktop-систем на базе KDE отмечается наибольшее количество неудач. Причиной этого, на наш взгляд, как раз и является расхождение базовой концепции построения интерфейса с интерфейсом Windows: идеология KDE более инвариантна, в ней стандартные операции над файлами возможно выполнить едва ли не десятком разных способов. В результате главное меню KDE очень разветвленное, Konqueror обременен различными всплывающими подсказками, контекстными меню, системой перекрестных ссылок, и как результат, вчерашний пользователь Windows «благополучно» теряется в этих лабиринтах. И только по прошествии некоторого времени (от 1 до 3-х месяцев!) происходит адаптация к новой среде.

Внешне графическая среда GNOME производит не самое лучшее впечатление для использования в качестве альтернативы Windows-среде: 2 панели, три кнопки в главном меню (причем на верхней панели), пиктограммы неярких (приглушенных) полутонов. Но благодаря сходству внутренней идеологии, нам она представляется более подходящей средой для внедрения Linux на Desktop-системе. Стоит отметить, что в компании TrustVerse (производящей дистрибутив LinXP) также ориентируются на Gnome в качестве основного интерфейса.

Основными элементами графического интерфейса, существенно влияющими на производительность работы, являются:

- Кнопка «Пуск», пиктограммы «Завершить сеанс», «Выключить компьютер»;
- Системные пиктограммы, относящиеся к стандартным файловым операциям, использованию сети, настройке компьютера, электронной почты, выхода в Интернет;
- Пиктограммы приложений, а также некоторые иконки для пользовательских файлов (`*.doc`, `*.xls`, `*.rar`, `*.zip` и т. п.);
- Главная панель, содержащая кнопку «Пуск», область быстрого запуска, список задач, системный лоток;
- Рисунок фона рабочего стола.

Зависимость важности тех или иных элементов оформления рабочего стола была определена по количеству использования (вызовов окон,

щелчков «мыши») в экспериментальной группе из 35 компьютеров под управлением ОС Windows XP SP2 RUS за 1 календарный месяц нормальной работы.

Исходя из вышесказанного, необходимо, чтобы внешний вид этих элементов был максимально похожим на вид популярных ОС семейства Windows, настроенных по умолчанию. Эта задача решается созданием объединенных тем для элементов управления, рамок окна и значков. Пример простой темы в стиле Windows XP можно найти на странице проекта: `http://www.gnome-look.org/content/show.php/eXPine?content=77119`.

Результатом тщательного подбора темы и настройки рабочего стола является снижение срока адаптации конечного пользователя с 30–40 до 1–3 дней. Пропорционально уменьшается количество обращений в службу технической поддержки по вопросам, связанным с ОС Linux.

Эксперименты в СибГАУ с графическими интерфейсами пользователей регулярно проводятся с 2006 г., в общей сложности установлено около 200 компьютеров с ОС Linux.

К сожалению, в сообществе OpenSource уделяется недостаточно внимания проблемам облегчения перехода пользователей Windows на свободное ПО. Производители дистрибутивов, как правило, применяют темы, установленные в KDE и GNOME по умолчанию. Но, как показывает практика, создание специализированных дружественных тем для бывших пользователей Windows при относительно невысоких затратах на программирование дает обычно значительный эффект по экономии времени и средств на обучение и техническую поддержку.

Л. С. Яковлев
Саратов,
Поволжская академия государственной службы

Свободное программное обеспечение для высшей школы: организационные проблемы

Аннотация

В докладе рассматриваются проблемы, с которыми сталкивается руководство ВУЗов при обращении к свободному ПО. Предлагаются решения этих проблем, связанные с развитием технической поддержки дистрибутивов Linux, оптимизацией пользовательских интерфейсов, активизацией маркетинговой политики.

Происходит удивительная вещь. Самая дешевая версия Windows (предустановка на ноутбуке) стоит \$60–70, но чтобы работать, ею ведь обойтись нельзя. Нужно обеспечить безопасность, загрузить хотя бы скромный набор утилит, и, наконец, программы собственно для работы. Вот, например, что нам предлагает фирма Utinet.ru, не самая дорогая в Сети. Стартовый пакет программ: пакет офисных программ OpenOffice.org 3.x; пакет аудио и видео кодеков; архиватор 7-Zip; браузер Mozilla FireFox 3.x; Skype — итого, 990 рублей. «Правильный» пакет программ: Антивирус Касперского 2009 или ESET NOD 32; ICQ 5.1; пакет офисных программ OpenOffice.org 3.x; пакет аудио и видео кодеков; архиватор 7-Zip; браузер Mozilla FireFox 3.x; Skype — итого, 2490 рублей. Возможно, кому-то из поклонников Windows это кажется и не слишком дорогим подарком любимому ноутбуку. Но за что вообще тут платить? OpenOffice, Mozilla FireFox, разнообразные пакеты кодеков и под Windows распространяются бесплатно. Видимо, так в мире проприетарного ПО принято: платить надлежит за все, за что фирме угодно взять деньги. За установку, надо полагать, хотя интересно было бы увидеть пользователя, не сумевшего самостоятельно установить Mozilla FireFox.

Итак, нас окружат миллионы людей, готовых платить до 4 тысяч рублей единовременно (потому как софт обновляется быстро, и неправда, будто обновления бесплатны: от выхода Windows XP до Vista не прошло и пяти лет, но бесплатно вам одну на другую никто не заменит) и при этом не решающихся перейти на свободное ПО. И главное для нас сейчас: почему, несмотря на постоянный дефицит

средств, не решается на этот шаг руководство подавляющего большинства высших учебных заведений? Всему на этом свете должна быть причина.

А причин здесь много. Первая, главная — «железо». Только очень наивные люди могут, применительно к распространению Linux, толковать, что его можно поставить на любую машину. Можно, если ставить будет профессионал. А ждать этого от человека, впервые пытающегося попробовать, что такое свободный софт, просто нелепо. Между тем, на российском рынке порядка 60–65% «железа», не способного работать под Linux. Дело тут прежде всего в «оптимизации» дешевых «девайсов», которые, собственно, и обновлений Windows не переносят — их разработали, чтобы здесь и сейчас можно было запустить современные игры, а если через пару лет они безнадежно устареют и просто станут хламом, тем лучше для производителя.

Решение этой проблемы лежит на поверхности. В Сети давно дискутируется вопрос об обязанности создателей дистрибутивов публиковать списки поддерживаемого железа. Это нужно хотя бы для того, чтобы облегчить выбор железа тем, кто хочет перейти на Linux. Следующим шагом могло бы стать лоббирование комментариев к антимонопольному законодательству, которые обязали бы торгующие компьютерным железом фирмы сертифицировать свои товары применительно к их готовности к установке различных операционных систем. Это вполне возможно, поскольку абсолютное большинство из них не является дистрибьюторами Microsoft, и, следовательно, косвенное навязывание покупателю использования софта одной единственной фирмы может трактоваться как нарушение антимонопольного законодательства. Разумеется, их полное право торговать теми продуктами, какими считают нужным, в том числе не совместимыми ни с чем, кроме Windows, но при этом можно обязать их уведомлять покупателя о неполноценности продаваемых устройств.

Надуманной является «проблема», о которой пишут едва ли не больше всего, а именно необходимость «переучиваться» при переходе на Linux. Что касается пользователей, все, чем абсолютное большинство владеет, — навыки работы с графическим интерфейсом. Что под ним, они все равно не знают и, вообще-то, могут сказать, что пользуются Windows только потому, что видят логотип при начальной загрузке. Среды KDE и Gnome могут, конечно, отличаться от среды рабочего стола Windows, но лишь в меру того, сколь пользователь этого захочет и сумеет настроить. Скажем, окна могут реагировать

на поведение мыши разными способами, но по умолчанию в большинстве дистрибутивов стоит та же реакция, что и в системе от Microsoft. Что же касается сисадминов, учиться — их работа. О чем стоит задуматься, так это техническая поддержка. Могут быть выстроены сетевые структуры, в которых она будет организована при минимизации затрат.

Более серьезной представляется проблема, которую можно определить как «горе от ума», или гипертрофия fetch protection до fool prove. Предельным примером ее проявления может служить нетбук Acer Aspire One. Хорошая для своего класса машина снабжена удивительным софтом под названием Linpus, являющимся искалеченной Федорой. Пользователя здесь полагают за существо невменяемое, с крайне скромными потребностями, послушное и неспособное к обучению. Два с половиной гигабайта (плюс гигабайт на swap) дискового (в смысле флэш) пространства заняты той самой Fedora, доступ к которой получить почти нельзя. Проникнуть в консоль все же можно без особой акробатики, но вот дальше, чтобы получить права root, уже нужно «уметь держать бубен левой ногой» ради того, чтобы пользователь получил OpenOfice, Firefox и еще пару программ. В Linux полно легких дистрибутивов, которые обеспечат весь этот нехитрый набор при весе меньше чем в полгигабайта, то есть раз в пять меньшем. Пользователю запрещено все: устанавливать программы, настраивать систему, что-то удалить. Оскорбительна для человеческого достоинства и документация, где нас учат пользоваться почтой и выходить в чат. К сожалению, все это — лишь крайний пример. Есть тенденция «упрощать» дистрибутивы, делая их негодными для работы. Даже в Windows у пользователя больше прав. Безопасность должна обеспечиваться иными средствами, и, вообще-то, вполне достаточно штатных.

Один из главных вопросов, бесспорно, — установка программ. В зависимостях новичок в Linux не разберется однозначно. И решение, по сути, одно: как можно больше программ standalone (как, опять же, Firefox). Кстати, сходный по смыслу шаг сделан в последней версии PC BSD. Разумеется, писать программы интереснее подо все богатство библиотек, но при этом надо же понимать, что сами наборы библиотек различаются от дистрибутива к дистрибутиву и не могут оставаться годами неизменными. А значит, должна быть и альтернатива, независимые приложения, тем более, что некогда существенное

соображение насчет экономии дискового пространства давно утратило всякий смысл.

Администрация ВУЗов вполне может уже в ближайшее время сделать шаги к переходу на свободное ПО, что будет иметь решающее значение для его распространения в целом, потому что с чем люди станут работать во время учебы, тем они после и будут пользоваться для работы и развлечений. Но чтобы это произошло, нужны не благие призывы, а реальные шаги навстречу этому потенциальному интересу. Если у сообществ, поддерживающих свободное ПО, есть выбор, на чем сосредоточить усилия, то это именно помощь разработчикам удобных для конечного пользователя дистрибутивов: максимально широко поддерживающих современное железо, имеющих возможно полные наборы графических утилит, для которых пишется достаточное количество разнообразных standalone программ. Именно такая стратегия на период трех-четырех лет может позволить действительно завоевать ВУЗы для свободного ПО.