

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

**Седьмая конференция  
«Свободное программное обеспечение  
в высшей школе»**

Переславль, 28–29 января 2012 года

Тезисы докладов

Москва,  
Альт Линукс,  
2012

Седьмая конференция «Свободное программное обеспечение в высшей школе»: Тезисы докладов / Переславль, 28–29 января 2012 года. М.: Альт Линукс, 2012. — XX с. : ил.

В книге собраны тезисы докладов, одобренных Программным комитетом седьмой конференции «Свободное программное обеспечение в высшей школе».

**ISBN 978-5-905167-08-9**

© Коллектив авторов, 2012

# Программа конференции

**28 января**

11.30-13.00 Регистрация участников и заселение

13.00 А. Е. Новодворский. Открытие. Информация оргкомитета

13.10 С. М. Абрамов. Приветственное слово

13.40–14.10 В. А. Сухомлин

Анализ современного состояния международных  
образовательных стандартов в области ИТ..... 8

14.10–15.10 Перерыв на обед

15.10–15.40 В. Габриель

Экосистема свободного ПО на платформе Microsoft и  
развитие инноваций ..... 12

15.40–16.10 Н. Н. Непейвода

Факультет прикладной логики: информатики-аналитики... 13

16.10–16.40 Г. В. Курячий, В. А. Старых и др.

Информика и Альт Линукс: новый комплект для  
образования ..... 15

16.40–17.00 Кофе-брейк

## **Вечернее заседание**

**17.00–20.00**

17.00–17.20 Д. Ванькевич, Г. Злобин

Быстрое развёртывание учебного полигона для проведения  
лабораторных работ в курсе «Системное  
администрирование ОС Linux» в компьютерных  
лабораториях общего пользования ..... 18

17.20–17.40	Д. Костюк, С. Дереченник, А. Шитиков	
	Оценка эффективности управления окнами в современных графических оболочках .....	20
17.40–18.00	В. А. Ружников	
	Опыт обучения СПО сотрудников районных администраций Самарской области на базе ФГОБУ ВПО ПГУТИ .....	23
18.00–18.20	В. Н. Брагилевский, С. А. Гуда, Г. В. Худoley	
	СПО на мехмате Южного федерального университета ....	26
18.20–18.40	Кофе-брейк	
18.40–19.00	Г. В. Разумовский	
	Методика преподавания технологии построения параллельных и распределенных приложений на базе свободного программного обеспечения .....	29
19.00–19.20	И. А. Хахаев	
	Свободные программы в ИМБИП НИУ ИТМО .....	31
19.20–20.00	Дискуссия	

## 29 января

### Утреннее заседание

10.00–13.00

10.00–10.15	И. Б. Рогожкина, А. Г. Кушниренко	
	Пиктомир: опыт обучения азам программирования старших дошкольников .....	33
10.15–10.30	Д. В. Хачко, В. В. Яковлев и др.	
	Кумир 1.9. Практикумы и исполнители. Средства интенсификации обучения .....	36
10.30–10.45	В. В. Яковлев, Д. В. Хачко и др.	
	Кумир 2.0: новые цели — новые решения .....	37
10.45–11.00	М. А. Ройтберг	
	Новые тенденции преподавания информатики в школе ....	44

11.00–11.20	В. Н. Лукин, Л. Н. Чернышов	
	Опыт использования приложения с открытым кодом StarUML в системе дистанционного обучения MOODLE с виртуальным практикумом .....	45
11.20–11.40	Н. С. Лагутина, А. М. Васильев, И. В. Парамонов	
	Обучение студентов объектно-ориентированному программированию на основе архитектурного шаблона Модель – Вид – Контроллер .....	47
11.40–12.00	Кофе-брейк	
12.00–12.20	С. А. Фомин	
	Эффективное дистанционное преподавание нетривиальных предметов .....	50
12.20–12.40	Е. Алексеев, О. Чеснокова и др.	
	Свободные математические программы. Опыт использования в техническом университете .....	53
12.40–13.00	Е. Алексеев, И. Филь	
	Об использовании свободного программного обеспечения для решения задач электротехники .....	56
13.00–14.00	Перерыв на обед	
<b>Дневное заседание</b> <b>14.00–18.00</b>		
14.00–14.20	М. В. Быков	
	SEO-дружелюбное web-приложение одной страницы — diglossa.ru.....	60
14.20–14.40	Е. А. Синельников	
	Использование системы контроля версий Git в преподавании курсов, связанных с разработкой ПО ...	62
14.40–15.00	А. Г. Михеев	
	Использование концепции бинарных отношений в системе RupaWFE как пример интеграции математической теории в свободное ПО .....	66

15.00–15.20	Д. Пынькин, И. Глецевич	
	Открытый подход к обучению студентов технической специальности ВУЗа .....	71
15.20–15.40	В. Л. Симонов, М. В. Храпченко	
	Применение свободного программного обеспечения для управления делопроизводством кафедры с учётом компетентностного подхода в соответствии с требованиями ФГОС 3 поколения .....	76
15.40–16.00	Кофе-брейк	
16.00–16.20	А. Н. Пустыгин, Б. А. Тарелкин, А. А. Ковалевский и др.	
	Программы автоматизированного построения технических комментариев на основе промежуточного представления открытых исходных текстов программ .	79
16.20–16.40	А. Н. Пустыгин, Б. А. Тарелкин, А. А. Ковалевский и др.	
	Прототипы строителей промежуточных представлений исходных текстов программ, основанные на компиляторах с открытым исходным кодом .....	82
16.40–17.00	А. В. Малибашев	
	Использование «Комплексной системы компьютерного обучения» в образовательном процессе технического университета .....	87
17.00–17.20	В. Олоничев, Б. Староверов	
	Linux в подготовке студентов IT-специальностей .....	89
17.20–17.40	Н. Н. Сидун, Е. А. Чичкарев	
	Использование вычислительного кластера под управлением ОС Linux в высшем учебном заведении ..	92
17.40–18.00	Ф. С. Занько	
	Свободные инструменты на языке Python в исследованиях турбулентности .....	95

**Вне программы**

А. Н. Гороховский

Идентификация, разбор и конструирование уравнений  
химических реакций с помощью Chemistry::Harmonia .. 99

М. Пожидаев, Д. Жарков, В. Кирюшкина

Исследование методов извлечения данных при помощи  
СПО и материалов энциклопедии «Википедия» ..... 105

С. А. Мартишин, М. В. Храпченко

Использование LAMP и MySQL Workbench в процессе  
обучения студентов..... 108

Владимир Сухомлин

Москва, МГУ имени М.В. Ломоносова

## **Анализ состояния международных образовательных стандартов в области информационных технологий**

В условиях информационного общества, где информационные технологии (ИТ) являются главным двигателем общественного прогресса, в сфере образования аналогичную роль выполняет система ИТ-образования (подготовка профессионалов в области ИТ, обучение информатике и ИТ, информатизация образования) — самая динамичная часть современной системы образования. Необходимость массовой подготовки профессиональных кадров в области ИТ для глобального рынка труда предопределила, актуальность разработки международных рекомендаций, учитывающих достижения и тенденции в науке, технологиях, образовании, и служащих ориентиром для университетов в соответствующей образовательной деятельности. Ответственность за решение задачи формирования таких ориентиров-рекомендаций в виде типовых учебных программ или куррикулумов (*curriculum*), называемых также стандартами куррикулумов, начиная с 60-х годов 20-го столетия, несут международные профессиональные организации — Ассоциация компьютерной техники (*Association for Computing Machinery, ACM*) и Компьютерное Сообщество Института инженеров по электронике и электротехнике (*Computer Society of the IEEE или IEEE-CS*) [1].

Новое поколение стандартов куррикулумом для области ИТ (или компьтинга — *Computing*, — академическое название области ИТ) создавалось в первое пятилетие настоящего столетия в виде целостной системы куррикулумов, построенной на основе единого методологического подхода, модульности, формирования структурированных объёмов знаний по основным направлениям подготовки ИТ-профессионалов.

В последующее пятилетие (а процесс развития куррикулумов принял постоянный непрерывный характер и осуществляется на принципах консорциумной стандартизации) практически все документы этой системы куррикулумов были переработаны и вышли в новых

---

редакциях. Цель настоящего доклада — анализ современного состояния системы международных образовательных стандартов в области ИТ (или компьютеринга), а также поиск возможностей использования наработок мирового сообщества в методическом обеспечении отечественной высшей школы.

В докладе дан анализ современного стека стандартов курикулумов для подготовки бакалавров компьютеринга, включая анализ следующих документов:

- Computing Science 2001 (CS2001),
- Computer Science 2008 (CS2008),
- Information Systems 2010 (IS2010),
- Software Engineering 2004 (SE2004),
- Information Technology 2008 (IT2008),
- Computer Engineering 2004 (CE2004),

а также руководства GSwE2009 — Graduate Software Engineering 2009 (GSwE2009), предназначенного для подготовки магистров в области программной инженерии.

В частности, в докладе проанализированы важнейшие принципы построения системы курикулумов, к которым относятся:

- целостность — использование единой методологии и структуры построения, единого понятийного контекста;
- знание-ориентированность — основным содержанием курикулумов являются собственно объёмы знаний (body of knowledge) по основным профилям подготовки;
- единый способ структурирования и представления объёмов знаний в виде трёх-четырёх-уровневой иерархической структуры — на верхнем уровне иерархии располагаются предметные области (areas) — самые крупные части объёма знаний, которые подразделяются на разделы или модули знаний (units), последние в свою очередь разбиваются на темы (topics), в некоторых случаях темы делятся на подтемы (subtopics);
- концепция ядра — выделение в объёмах знаний минимально необходимого образовательного содержания, реализация которого во всех учебных программах обеспечивает единство образовательного пространства, мобильность учащихся в рамках профиля или всего направления, гарантию качества базовой подготовки;

- чёткая спецификация профессиональных характеристик профилей подготовки, системы целей обучения, итоговых профессиональных характеристик выпускников;
- включение рекомендаций методического и педагогического характера (по диверсификации программ подготовки, составлению учебных планов, компоновки курсов из модулей знаний в соответствии с выбранной педагогической стратегией реализации учебной программы, организации профессиональной практики, реализации процессов обучения);
- включение описаний учебных курсов и пакетов курсов для различных педагогических стратегий реализации курикулов.

Рассмотренная выше система стандартизованных на международном уровне курикулов создавалась для обеспечения глобализации процессов подготовки востребованных мировой экономикой ИТ-кадров, единства и однородности образовательного пространства, мобильности студентов и преподавателей, интеграции усилий академических, промышленных, коммерческих и правительственных организаций в создании и непрерывной актуализации современного методического обеспечения системы ИТ-образования. Всё это делает необходимым развитие национальной образовательной системы в области ИТ с учётом нарабатанного международного задела. В противном случае российское образование может быстро потерять даже надежды на конкурентоспособность в мировом образовательном пространстве.

В то же время анализ современного состояния методических основ отечественной высшей школы [2], в основе которых лежит навязанная системе образования административным путём порочная концепция федерального государственного образовательного стандарта третьего поколения (ФГОС), показывает невозможность адекватного переноса рассмотренных современных образовательных технологий в российское образование. В частности, изгнание из ФГОС содержания обучения, т.е. самих знаний, передача которых и есть основная суть образования, и замена этого содержания лозунгами-компетенциями, делает бесполезными ФГОСы для практики, экранирует возможность использования современных технологий управления знаниями, закладывает опасные тенденции, ведущие к разрушению единого образовательного пространства страны, сформированное на базе стандартов второго поколения. Последнее негативное явление многократно усиливается «полистандартизацией» в российском образовании — пред-

---

писанной на законодательном уровне необходимостью элитным университетам, а таких порядка 40, учить по самостоятельно устанавливаемым стандартам. В целом всё это говорит о том, что развитие высшего профессионального образования (ВПО) в России следует курсом противоположным вектору развития мирового университетского образования. И предстоит огромная работа по выправлению этого курса. Такая работа посильна только консолидированному профессиональному сообществу. Поэтому поиск организационных форм интеграции усилий профессиональной общественности в решении неотложных задач российского образования является актуальной задачей. Только общими усилиями можно развернуть и спасти этот корабль под названием ВПО!

## Литература

- [1] *В.А. Сухомлин.* Анализ международных образовательных стандартов в области информационных технологий. Современные информационные технологии и ИТ-образование / Сборник избранных трудов VI Международной научно-практической конференции: учебно-методическое пособие. Под ред. проф. В.А. Сухомлина, ISBN 978-5-9556-0129-8, — М.: ИНТУИТ.РУ, 2011. с. 6-46.
- [2] *В.А. Сухомлин.* Реформа высшей школы — анализ итогов. Современные информационные технологии и ИТ-образование / Сборник избранных трудов V Международной научно-практической конференции: учебно-методическое пособие. Под редакцией В.А. Сухомлина, ISSN 978-5-9556-0115-1, — М.: ИНТУИТ 2010, с. 3-22.

Владимир Габриель  
Москва, Microsoft

## Экосистема свободного ПО на платформе Microsoft и развитие инноваций

### Аннотация

Инновационные проекты требуют экосистемы обмена знаниями и инвестиций для быстрого развития. В сообщении будет рассказано о ресурсах для развития СПО проектов и использовании их в образовании, а также продемонстрированы примеры natural user interface (NUI) программ созданных СПО сообществом и доступных в библиотеке проектов [codeplex.com](http://codeplex.com)

Майкрософт прошел достаточно большой путь своих отношений к FOSS сообществу и способах сотрудничества с ним. Ярче всего эволюция компании может быть продемонстрирована в инновационных продуктах, когда разработка новых услуг или новых устройств позволяет Майкрософт активно взаимодействовать.

Для повышения эффективности такого взаимодействия Майкрософт создал инфраструктуру поддержки открытых проектов — <http://codeplex.com>, которая позволяет развивать свободные проекты на платформе Microsoft. В моем сообщении рассмотрена серия проектов, которые позволяют приобщить интересующихся разработчиков (в т.ч. школьников и студентов) к разработки собственного программного обеспечения в области Natural User Interface с использованием Microsoft Kinect, что позволяет приобщить начинающих разработчиков к наиболее инновационной области взаимодействия человека и компьютера.

В ходе сообщения будут продемонстрированы свободные программы для Microsoft Kinect, которые могут использоваться как прототипы новых решений или как удачные методические пособия при изучении темы естественного пользовательского интерфейса.

---

Николай Непейвода, Евгений Кочуров, Вячеслав Пупышев.  
Ижевск, УдГУ [www.udsu.ru](http://www.udsu.ru)

## Факультет прикладной логики: информатики-аналитики

### Аннотация

Факультет прикладной логики создаётся в Удмуртском государственном университете. Специальность: фундаментальная информатика (бакалавриат и магистратура) с траекторией обучения, направленной на обработку нечисленной информации. В феврале возобновляется олимпиада по информатике «Реализация нереализуемого».

### Информатик-аналитик

В чём заключается его работа?

1. Понять непонимаемое
2. Формализовать неформализуемое
3. Описать неопишуемое
4. Высказать невербализуемое
5. Впихнуть невпихуемое
6. Соединить несоединимое
7. Разделить неделимое
8. Найти альтернативу безальтернативному
9. Опровергнуть бесспорное
10. Реанимировать опровергнутое
11. Решить неразрешимое
12. Обрезать необрезаемое
13. Отбросить неотъемлемое
14. Принять неприемлемое
15. Понять, какая доля шутки в этой шутке
16. Делать вид, что ты идёшь строем и в ногу со всеми, когда ты на самом деле в своей пещере отшельника

17. Представить так, как будто всё новое уже широко используемое
18. Представить нечто несущественное как “инновацию”, чтобы замаскировать действительно новое
19. И в результате реализовать нереализуемое.

Чего он не должен делать?

1. Измерять неизмеримое.

Где он может работать? Везде, где есть трудные задачи на обработку сложной информации.

Что это по существу? Безжалостная альтернатива т. н. «менеджерам», которых учат делать собственную карьеру, а выпускают управлять другими.

## Олимпиада «Реализация нереализуемого»

Стартует 1 февраля 2012 года.

Олимпиада предназначена для выявления, развития и поддержки студентов, имеющих способности стать аналитиками и руководителями в области информационных технологий. Поэтому она ортогональна существующим структурам олимпиад ACM и конкурсов TopCoder, Imagine Cup. Аналитик и руководитель среднего звена должен уметь прежде всего разбираться в неточно и неправильно поставленных задачах в любой предметной области и переводить их на язык точных постановок. Навыки кодирования для него необходимы, но не являются определяющими. Олимпиада проводится ежегодно с сентября по май (в 2012 году с февраля по май). Она состоит из трёх туров: двух заочных и одного очного.

1. Первый заочный тур является отборочным. На нём участникам предлагается серия 5-7 задач из одной и той же области с постепенно усложняющимися условиями. Следующую задачу участник имеет право решать после решения всех предыдущих. Прошедшие все этапы лестницы отбора имеют право участия во втором туре.
2. Второй заочный тур: участники решают три нестандартные творческие задачи:
  1. задачу с неточным условием, требующим содержательного дополнения и уточнения;

2. задачу соревновательного типа, в которой программа участника будет в том или ином смысле соревноваться с программами других участников и тестовыми программами жюри;
  3. задачу состязательного типа, в котором участниками предлагается точно поставленная задача, для которой неизвестно точное решение, и оценивается качество приближённого решения.
3. На заочных турах участники имеют право пользоваться любым общедоступным инструментом программирования. Жюри имеет право проверять код решений задач как в целях его оценки, если та или иная характеристика кода оговорена в условии задачи, так и в целях выявления наиболее интересных решений.
4. От 20 до 30 победителей заочных туров приглашаются на очный тур в Ижевск, УдГУ. Продолжительность очного тура — 3 дня.

Г. В. Курячий, А. В. Бояршинов, В. А. Старых, В. В. Кондауров  
Москва, ALT Linux, ФГУ ГНИИ ИТТ «Информика»

## **Информика и Альт Линукс: новый комплект для образования**

### **Аннотация**

Компания Альт Линукс и сообщество ALT Linux Team продолжает поддержку и совершенствование школьной линейки дистрибутивов. После выхода партикулярных дистрибутивов на базе Шестой платформы [2, 3] готовится к выпуску Школьный комплект, продолжающий линейку дистрибутивов для учебных заведений [4, 1]. Расширение возможностей дистрибутива позволило использовать его в рамках проекта ФГУ ГНИИ ИТТ «Информика» по организации мобильного учебного класса.

### **Дистрибутив Альт Линукс Школьный**

Новый школьный комплект, по традиции, является существенным расширением пары «ALT Linux Server» + «ALT Linux Desktop», как в области программного наполнения клиентской и серверной части, так и предлагаемыми вариантами использования.

- Серверная часть включает в себя:
  - средства организации учебного класса с единой системой авторизации,
  - средства централизованного развёртывания учебного класса,
  - средства быстрой организации терминального класса (без-дисковые клиенты),
  - систему управления обучением МООДУС (LMS Moodle) [6],
  - систему совместной разработки контента MediaWiki [7],
  - электронный классный журнал РУЖЭЛЬ [5].
- Клиентская часть представлена тремя вариантами рабочего окружения:
  - GNOME3, в том числе в исполнении для планшетов,
  - XFCE4, в том числе в исполнении для нетбуков,
  - GNOME2 (входит в состав серверной части, но может использоваться для организации рабочего места на сервере или рабочей станции).
- Программное наполнение клиентской части адаптировано к требованиям государственного контракта М-3-9208нпр. Это предполагает наличие некоторых приложений в составе дистрибутива по умолчанию. Разумеется, к установке доступно всё программное обеспечение в рамках Шестой платформы [9].
- Предполагается использование клиентских компонентов дистрибутива на нетбуках и планшетах с учётом специфики их работы; ведётся соответствующая доработка.

## Проект «Мобильный класс»

Упомянутая выше адаптация линейки сделала возможным использование Школьного комплекта в комплексных инфраструктурных решениях, например в ходе работ по выполнению госконтракта от 19 августа 2011 г. № М-3-9208нпр (ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2007-2013 годы»). Решением этой задачи занимается ФГУ "Государственный научно-исследовательский институт информационных технологий и телекоммуникаций" (Информика).

---

Главная цель проекта — объединение различных по способу использования электронных устройств в единый комплекс, решающий задачи информационного обеспечения учебного процесса:

- Рабочее место преподавателя организуется на базе ПК или мобильного ПК (нетбука) с установкой полного спектра ПО плюс ПО для управления комплексом. Используется окружение на базе XFCE4.
- Стационарное или мобильное рабочее место ученика организуется на базе мобильного ПК с установкой полного спектра ПО и полной интеграцией в инфраструктуру класса. Используется окружение на базе XFCE4, подходящее также и к стационарным ПК.
- Удалённое рабочее место ученика организуется на базе планшетного ПК с установкой полного или урезанного (в зависимости от характеристик компьютера) спектра ПО. Удалённое рабочее место может быть включено в инфраструктуру класса, но основное его назначение — работа в отсутствие доступа к внутренним ресурсам. Используется адаптированное окружение на базе GNOME3.
- Устройство для чтения электронных книг организуется на базе соответствующего Android-устройства с установкой программ, обеспечивающих работу с базовым контентом (электронные книги в различных форматах, файлы текстовых процессоров, графика и т. п.).

Дополнительно в рамках работ по госконтракту весь комплекс оснащается подсистемой синхронизации образовательного контента, призванной актуализировать информацию, содержащуюся на компьютерах ученика и учителя.

## Литература

[1] <http://www.altlinux.ru/news/archive/2011/08/item/646/>

[2] <http://www.altlinux.ru/news/archive/2011/11/item/655/>

[3] <http://www.altlinux.ru/news/archive/2009/10/item/558/>

[4] <http://www.altlinux.ru/solutions/state/school-server>

[5] <http://www.rujel.net/>

- [6] <http://moodle.org/?lang=ru>
- [7] <http://ru.wikipedia.org/wiki/MediaWiki>
- [8] <http://contests-mon.informika.ru/lot/7771/>
- [9] <http://ftp.altlinux.org/pub/distributions/ALTLinux/p6/>

Дмитрий Ванькевич, Григорий Злобин

Львов, Львовский национальный университет имени Ивана Франко

## **Быстрое развёртывание учебного полигона для проведения лабораторных работ в курсе «Системное администрирование ОС Linux» в компьютерных лабораториях общего пользования**

### Аннотация

В докладе рассмотрено использование технологии виртуализации OpenVZ для проведения лабораторных работ спецкурса «Системное администрирование ОС Linux»

Первоначально лабораторные работы курса «Системное администрирование ОС Linux» планировалось выполнять в специализированной лаборатории на ПК с процессором Intel Celeron 400 МГц, 128 МБ оперативной памяти без предустановленной операционной системы. Однако, в конце августа оказалось, что подготовительное отделение, одолжившее у кафедры помещение до 1 июля 2011 г., освобождать его не желает. Поэтому возник вопрос о быстром развёртывании программных и аппаратных средств для проведения лабораторных работ курса «Системное администрирование ОС Linux» в лаборатории общего пользования. При этом следовало учесть, что некоторые лабораторные работы по системному администрированию являются потенциально опасными для работоспособности учебного компьютера, например — работа с редактором разделов, конфигурирование загрузчика и т.п. Ранее для быстрого восстановления системного и пользовательского ПО использовалась система VpBatch[1], которая доказала свою эффективность, но требовала значительных трудозатрат для начальной настройки, особенно в случае, если компьютеры учебного класса имели разнотипное аппаратное обеспечение. Поэтому

была поставлена задача — создать полигон для проведения лабораторных работ, который не требовал бы значительных трудозатрат для развёртывания.

Для проведения лабораторных работ использовалось следующее оборудование:

1. сервер с процессором AMD Phenom(tm) II X4 945, 8ГБ оперативной памяти и операционной системой GNU Debian и системой виртуализации OpenVZ[3];
2. 5 рабочих мест студента с процессором Intel Pentium 4, 512 МБ оперативной памяти и операционной системой Microsoft Windows XP;
3. На сервере было создано 8 виртуальных машин — 5 для студентов и 3 служебные (кеширующий прокси пакетов apt-cacher-ng, wiki с методическими материалами, локальный DNS-сервер).

Каждому студенту была выделена собственная операционная система Debian GNU/Linux в контейнере OpenVZ и доступ к управлению контейнером при помощи ovz-web-panel [4]. На клиентских компьютерах для управления виртуальными машинами использовались ssh-клиент (putty) и Веб-обозреватель.

В пределах спецкурса «Системное администрирование ОС Linux» студенты должны были установить и настроить систему управления вебхостингом ISPConfig [2] в ходе последовательного выполнения таких работ:

лабораторная работа №1 «Ознакомление с программами OpenVZ, PuTTY, vi»;

лабораторная работа №2 «Установка компонент почтового сервера»;

лабораторная работа №3 «Установка компонент Веб-сервера»;

лабораторная работа №4 «Работа с сервером электронной почты»;

лабораторная работа №5 «Перенаправление стандартных потоков ввода/вывода в ОС Linux».

Для предохранения от выхода из строя виртуальных машин в ходе выполнения лабораторных работ предусматривалось создание контрольных точек для восстановления работоспособной конфигурации.

Несмотря на экспромтную реализацию полигона в целом результаты проведения лабораторных работ курса «Системное администрирование ОС Linux» можно считать удовлетворительными.

## Литература

- [1] *Р. Петрів, Д. Ванькевич, Х. Засадна.* Система резервного відновлення програмного забезпечення комп'ютерних лабораторій, 2007, Збірник наукових праць, Комп'ютерні технології друкарства, №17, Українська академія друкарства, ISBN978-966-322-096-3, <http://goo.gl/18cYq>
- [2] <http://www.ispconfig.org/>
- [3] [http://wiki.openvz.org/Main\\_Page](http://wiki.openvz.org/Main_Page)
- [4] <http://code.google.com/p/ovz-web-panel/>

Дмитрий Костюк, Станислав Дереченник, Алексей Шитиков  
Брест, Брестский государственный технический университет

### Оценка эффективности управления окнами в современных графических оболочках

#### Аннотация

Выполнен практический анализ эффективности многооконной работы оператора в условиях нововведений современных графических оболочек, направленных на унификацию интерфейсов настольных и планшетных компьютеров. С помощью предложенной системы тестирования для графических оболочек Plasma Desktop, Gnome Shell и Unity проведено экспериментальное исследование скорости, устойчивости к ошибкам и напряженности работы оператора при частом переключении окон. Обосновывается вывод о преимуществах классической панели задач и об ограниченной целесообразности ее замены на полноэкранный переключатель миниатюр окон при использовании многооконных приложений.

Исторически первым расширением метафоры рабочего стола для переключения фокуса окон была док-панель. В 90-х ее почти везде заменяет более компактная и информативная панель задач. Однако сейчас из-за роста популярности планшетов разработчики заняты поиском универсальных решений, удобных как при управлении мышью, так и на сенсорном экране. В версиях графических оболочек 2011 г. наблюдаются изменения, направленные на отход от классической метафоры: альтернативные интерфейсы запуска приложений, стимулирование пользователя к полноэкранной работе и отказ от панели задач (отсутствующей вовсе, либо замененной на док, более удобный при сенсорной активации).

В ряде публикаций делаются замечания о меньшей пригодности таких графических оболочек для работы в многозадачной среде благодаря заимствованию решений из интерфейсов портативных устройств, частично или полностью построенных на принципах невытесняющей многозадачности. Среди рассмотренных изменений на эффективности многозадачной работы оператора должны сильнее всего сказаться перемены, связанные с панелью задач, как с наиболее активно используемым вспомогательным элементом.

Нами предпринята попытка оценить влияние данных изменений на работу в многооконной среде. Для исследования выбраны 3 свободные графические оболочки: одна классическая (KDE с панелью задач) и две экспериментальной природы (Gnome Shell с экраном обзора и Unity с вертикальной док-панелью). Разработанная программа тестирования создает для оператора несколько окон и выделяет фиксированный промежуток времени на то, чтобы выполнить с помощью манипулятора (например, мыши) максимальное количество однотипных действий (использовано копирование чисел через буфер обмена). Система фиксирует время и сделанные ошибки, а состояние оператора регистрируется датчиком частоты сердечных сокращений (ЧСС). Само тестирование выполнено преимущественно студентами — участниками лаборатории микро и мидиэргономики БрГТУ (22 человека).

Тест включает два этапа. На первом этапе пользователь имеет дело с двумя перекрывающимися окнами, многократно копируя числа из одного в другое и переключая фокус окон средствами графической оболочки (по условиям клавиатура не используется кроме специально оговоренных случаев). Окна создаются разными исполняемыми модулями — для оценки эффективности переключения окон различных приложений. На втором этапе оператор копирует числа из окна-источника в одно из трех окон-приемников, указанное ему случайным образом. Так выявляется эффективность управления окнами многооконных приложений.

Чтобы учесть разнообразие защитных реакций от перегрузки оператора, фиксировались ошибки четырех типов: нажатие на неправильную кнопку, пропуск и дублирование числа и неверный выбор окна. Длительность каждого этапа составляла 5 минут (за это время при легкой нагрузке ЧСС практически выходит на устойчивое состояние, но на эффективности оператора не успевают сказаться монотонность работы). Сравнение показателей, полученных для разных

оболочек, выполнялось отдельно для каждого оператора, с последующим усреднением.

Рассматривались три типа показателей: быстрота выполнения оператором заданных действий, допущенные ошибки и параметры сердечного ритма. Для оценки быстроты выбран темп прохождения теста  $\tau$  (число итераций, выполняемых оператором за минуту). Чтобы абстрагировать данные об ошибках от числа итераций для конкретных оператора и оболочки, рассматривались среднее число сбоев  $n$  на одну итерацию и средняя длительность сбоя  $t$  (число затронутых сбоем итераций). При оценке ЧСС сравнивался интегральный параметр: напряженность выполнения теста  $\nu$ , т. е. среднее за 5-минутный интервал число сердечных сокращений, приходящееся на одну итерацию в конкретном тесте. Это уменьшало вклад, вносимый в показатель разной утомляемостью, характерной для оператора при разном темпе операций. В диапазоне напряженности выделяется оптимальная зона, а также зоны недогрузки и перегрузки, где эффективность оператора снижается, причем в ряде случаев положительное воздействие на эффективность оказывает слабая помеха (например, действия по переключению окон, второстепенные к основному заданию).

Основные результаты представлены на рисунке 1.

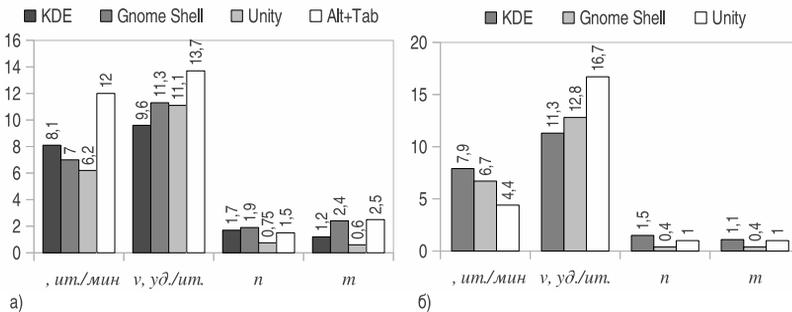


Рис. 1: Результаты тестирования на первом (а) и втором (б) этапах

В целом панель задач наиболее эффективна для переключения окон — по скорости, по нагрузке и по концентрации оператора на выполняемой задаче. При работе с однооконными приложениями наименее быстр режим обзора Gnome Shell, а вертикальная док-панель занимает промежуточное положение. Режим обзора также дает наи-

большую нагрузку и максимальную плотность ошибок, вопреки тому, что вынесение управления окнами в отдельный экран призвано увеличить концентрацию на работе. Низкий темп в Unity можно объяснить меньшей эффективностью перемещения мыши в горизонтальном направлении, что согласуется с кинематической моделью «кисть-предплечье-плечо».

При управлении многооконными приложениями наименее эффективна док-панель из-за группирования окон приложения. Панель задач дает максимальную скорость работы, но при этом возрастает вероятность ошибки. Минимальное же число ошибок достигается в Gnome Shell при визуальной различимости миниатюр окон.

Т. о., при выборе окружения рабочего стола (в учебной компьютерной лаборатории или для компьютеров предприятия) графические оболочки с рассмотренными нововведениями могут быть рекомендованы преимущественно при работе с единственным однооконным приложением — либо в ситуации, требующей монотонной работы не менее чем с тремя полноэкранными задачами, когда предъявляются более высокие требования к точности, чем к скорости работы.

**Ружников В.А.**

Самара, ФГОБУ ВПО Поволжский государственный университет телекоммуникаций и информатики, Самарский региональный тренинг центр

Проект: Курсы повышения квалификации [www.psuti.ru](http://www.psuti.ru)

## **Опыт обучения СПО сотрудников районных администраций Самарской области на базе ФГОБУ ВПО ПГУТИ**

### **Аннотация**

В период с сентября по декабрь 2011 года на базе Поволжского государственного университета телекоммуникаций и информатики проводились курсы по повышению навыков сотрудников органов исполнительной власти и местного самоуправления Самарской области по использованию информационных технологий. В результате обучение прошли более 200 человек. В докладе планируется осветить все «подводные камни», с которыми столкнулись преподаватели учебного центра.

В период с сентября по декабрь 2011 года на базе Самарского регионального тренингов центра при Поволжском государственном университете телекоммуникаций и информатики (ПГУТИ) проводились курсы по повышению навыков сотрудников органов исполнительной власти и местного самоуправления Самарской области по использованию информационных технологий.

Слушатели были разбиты на группы: «базовую» и «администраторы».

Далее поговорим об опыте общения с первой категорией. В большинстве своём это люди, которым нужен в повседневной работе «офис», Интернет, электронная почта. Возраст аудитории — свыше 40, за редким исключением.

Для данной категории слушателей был разработан курс «Основы использования ОС Windows и ALTLinux».

Первая часть включала основы работы в Windows, Word, Excel, Powerpoint, вторая — посвящена использованию СПО, в частности: основы использования ОС ALTLinux (набор повседневных программ — Thunderbird, Firefox, офисные игрушки, «аськи») и большой блок был отведён на использование пакета OpenOffice.org 3. Курс был недельным, 40 часов.

Общий итог по первой части курса — слушателям привычно работать в среде, которую они видят каждый день, если не у себя на компьютере, то на компьютере у своих детей, коллег.

Приступив ко второй части курса, слушатели узнали всю предысторию, идеологию СПО, познакомились с существующими лицензиями, особенностями файловой системы, дистрибутивами Linux и т.д. Согласно условиям контракта, обучение необходимо было проводить на базе дистрибутива *ALT Linux 6.0 Kdesktop*.

Слушатели, попав в непривычную среду *ALT Linux 6.0 Kdesktop*, реагировали по-разному, автор специально давал время слушателям самостоятельно познакомиться с дистрибутивом, моделирую ситуацию «быстрого внедрения». В большинстве своём многих пугал непривычный (по сравнению с Windows) интерфейс, однако некоторые легко умудрялись находить как офисные игры (надо сказать, что это первое, что искалось) так и офисные программы.

Практические занятия были посвящены работе с KDE (настройки интерфейса, работа с папками и т. д. ) и работе с повседневными программами.

Общий итог, который вынесен в результате обучения работы с KDE, — использовать нежелательно, интерфейс для «чайника» должен быть простым. Большое количество настроек, перегруженный интерфейс, программы-комбайны — всё это осложняло восприятие и оставляло негативное восприятие об ОС Linux.

Автор пошёл на некоторое отступление от пунктов договора и провёл эксперимент по обучению в среде *Xfce*. Удивительно, но факт — пользователи не испытывали дискомфорта, легко создавали папки, настраивали интерфейс, делали привычную работу. Итог — для первоначального знакомства с ОС Linux необходимо использовать простые графические среды особенно при внедрении в корпоративном секторе.

Работа с привычным набором программ (почта, «аська», браузеры, игрушки) не вызывало ни у кого трудностей.

Офисный пакет OpenOffice.org встретили без проблем. Надо сказать, что для обучения в первой части использовался пакет Microsoft Office 2010. Итог — интерфейс OpenOffice.org является для многих более удобным, особенно для тех, кто работал с более ранними версиями Microsoft Office.

Курс для «администраторов» по желанию заказчика назывался «Системное администрирование ALTLinux», основной упор необходимо было сделать на настройках интеграции ALTLinux в Windows сети, настройках сканеров, добавлению пользователей, программ. Однако, в процессе общения с аудиторией стало понятно, что необходимы ещё и дополнительные знания по установке ftp-серверов, web-серверов, ssh, программированию на shell, хотя это уже входит в курс «сетевое администрирование». Поэтому дополнительно был начитаны и данные разделы. В большинстве своём слушатели имели первоначальный опыт общения с Linux, но кому-то не хватало времени, а кому-то системного подхода к обучению. Слушатели остались довольны данным курсом, а для многих это стало следующим этапом развития в профессиональной деятельности.

Итогом работы преподавания основ СПО в ПГУТИ стал накопленный опыт, который дальше планируется учесть в преподавании для следующих слушателей. Автор в настоящее время ведёт переговоры по организации 2-х дневных семинаров для учителей Самарской области .

Виталий Брагилевский, Сергей Гуда, Герман Худолей  
Ростов-на-Дону, Южный федеральный университет

## СПО на мехмате Южного федерального университета

### Аннотация

В докладе представлен опыт настройки и использования локальной сети из 80 компьютеров на базе Ubuntu Linux под управлением FreeBSD-сервера для обучения студентов и школьников. В частности, описывается порядок подготовки дисплейных классов к учебному году, организация доступа к серверным каталогам со стороны пользователей (Samba + дополнительные инструменты), контроль доступа к интернету с локальных машин (Squid), использование сервера для поддержки учебного процесса (Moodle, Joomla). Также характеризуется набор СПО, используемого в рамках учебного процесса при обучении студентов (программирование, математические пакеты) и школьников (офисные приложения, графика).

Центром сетевой инфраструктуры лаборатории, обслуживающей дисплейные классы факультета математики, механики и компьютерных наук Южного федерального университета, является сервер, работающий под управлением операционной системы FreeBSD. Последние несколько лет в дисплейных классах используется дистрибутив Ubuntu Linux.

### Подготовка дисплейных классов к учебному году

В настоящее время в состав лаборатории входит шесть дисплейных классов. Аппаратное обеспечение по классам довольно сильно различается: имеется пять совершенно разных типов комплектации. Компьютеры используются ежедневно (включая воскресения — в эти дни идут занятия со школьниками) с 10:00 до 21:30 (дневное и вечернее отделения) с сентября по май, некоторое ослабление графика работы имеет место в январе и июне (сессии). В июле студенты отрабатывают предусмотренную учебными планами летнюю практику, которая также проходит на компьютерах. Понятно, что поддержание актуальных версий операционных систем на таком большом количестве машин и в таких условиях является чрезвычайно сложной задачей.

Задача обновления систем решается посредством программы `udrcast`, позволяющей организовать групповую рассылку (`multicast`) необходимых разделов жёсткого диска по локальной сети. Сначала система обновляется на одном компьютере, затем все машины (или машины одного дисплейного класса) загружаются по сети, принимая модифицированное ядро Linux, содержащее клиент `udrcast`, с `tftp`-сервера. После загрузки ядра на всех машинах автоматически стартует `udrcast`, на обновлённой машине запускается раздача и в течение получаса 100-гигабайтный раздел копируется сразу на все машины. Групповая рассылка гарантирует, что передаваемые данные будут получены одновременно на всех подписанных на раздачу машинах. При этом важно, что компьютеры смежных сегментов сети работают без всяких помех.

После копирования разделов жёсткого система Linux работает без всяких изменений. Другие системы обычно требуют некоторой дополнительной настройки (особенно в ситуациях, когда комплектации машины-источника и машины-приёмника существенно различаются).

С поступлением нового оборудования хранящееся на `tftp`-сервере ядро Linux, с которого загружаются обновляемые по сети компьютеры, разумеется, приходится менять. Сборка этого ядра происходит с учётом минимизации числа включаемых драйверов.

## Доступ к сетевым ресурсам

Каждый пользователь дисплейных классов (студенты, школьники, преподаватели) имеют собственный каталог на сервере, доступ к которому можно получить с любого компьютера любого класса. Общее число пользователей в течение года превышает 1000 человек. Кроме того на факультете регулярно проводятся олимпиады, в которых принимают участие школьники и студенты со всей области. Каждому из них необходим собственный сетевой ресурс.

Оперативное заведение всех этих пользователей на сервере довольно проблематично, поэтому для этой цели на языке Python разработано специализированное ПО. Использование языка Python позволяет на всех используемых системах применять одну и ту же кодовую базу. Код сценария подключения загружается с сервера, поэтому при добавлении новой группы пользователей нет необходимости что бы то ни было менять на стороне клиента. Результат работы сценария —

подключенный сетевой ресурс, назначенный авторизованному пользователю.

В дисплейных классах по умолчанию открыт доступ к сети Интернет. Весь доступ осуществляется через прокси-сервер Squid, кэширующий наиболее часто запрашиваемые ресурсы и позволяющий при необходимости оперативно отключать доступ к Интернету в конкретном дисплейном классе.

## Поддержка учебного процесса на сервере

На сервере установлена система Moodle, обеспечивающая методическую поддержку для более чем двух десятков курсов. Наполнение курсов различается от элементарной публикации методических материалов до полной интерактивности, включающей тестирование, проверку работ и информирование студентов о результатах.

В учебном процессе также используется викимедиа-система (создание и поддержание учебных материалов студентами), форум (общение студентов и преподавателей), и несколько сайтов на базе CMS Joomla (сайты факультета и направления ИТ).

## Используемое в процессе обучения СПО

В обучении школьников принята идея использования в рамках одного курса различных операционных систем. В части Linux в основном применяются программы пакета LibreOffice и графические программы Gimp, Dia, Blender.

Для обучения студентов обычно используется среда Eclipse (Java и C/C++),  $\text{T}_{\text{E}}\text{X}$ Live и  $\text{T}_{\text{E}}\text{X}$ Maker, GHC (программирование на Haskell), различные реализации языка Пролог, Sage (комплекс открытого математического ПО).

Г.В.Разумовский

Санкт-Петербург, Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина) (СПбГЭТУ)

## **Методика преподавания технологии построения параллельных и распределенных приложений на базе свободного программного обеспечения**

### Аннотация

Рассматривается программа подготовки специалистов в области построения распределенных информационных систем с использованием ОС Linux и языка Java.

Одной из основных тенденций развития современных средств вычислительной техники и обработки информации является создание и совершенствование распределенных вычислительных систем. На кафедре Вычислительной техники СПбГЭТУ при разработке учебного плана бакалавров и магистров по направлению «Информатика и вычислительная техника» учитывалось не только актуальность данной тематики, но и активно развивающийся процесс перехода на свободное программное обеспечение (СПО), которое дает возможность студентам легитимно использовать программные продукты на своих компьютерах при выполнении самостоятельной работы. Кроме того, знания продуктов СПО обеспечит в будущем его трудоустройство и легкую адаптацию в процесс разработки информационных систем. Поэтому методика преподавания технологии построения параллельных и распределенных приложений была построена таким образом, чтобы учесть все указанные обстоятельства.

Вопросы построения распределенных приложений рассматриваются в двух дисциплинах: «Организация процессов и программирование в среде Linux» и «Программирование распределенных систем». Целью первой дисциплины является изучение внутренней организации ядра ОС Linux и программного интерфейса, обеспечивающего управление, взаимодействие и синхронизацию процессов на базе одного компьютера, а целью второй дисциплины — это изучение способов построения сетевых распределенных приложений с использованием языка Java.

Основные задачи дисциплины «Организация процессов и программирование в среде Linux»:

1. Ознакомить с принципами построения и реализацией операционной системы Linux.
2. Ознакомить с механизмами реализации параллельных процессов в среде Linux.
3. Научить пользоваться системными вызовами для организации параллельных процессов.

В результате изучения этой дисциплины студенты должны:

*Знать* — внутреннюю организацию операционной системы Linux, модели работы ее отдельных подсистем, способы управления и взаимодействия процессов.

*Уметь* — проектировать, разрабатывать и отлаживать программы на языке C++, в которых используются системные вызовы, обеспечивающие порождение, взаимодействие и синхронизацию процессов.

*Владеть* — практическими навыками самостоятельного объектно-ориентированного программирования в среде Linux.

В рамках дисциплины «Организация процессов и программирование в среде Linux» читаются лекции (34 часа) и проводятся лабораторные работы (34 часа). На лекциях рассказывается архитектура ОС Linux, подсистема начальной загрузки, процессы и их представление в ОС Linux, алгоритмы порождения и диспетчеризации процессов, организация вызова системных функций, программный интерфейс управления, взаимодействия и синхронизации процессов, управление оперативной памятью. На лабораторных работах студенты пишут программы, где решаются такие задачи как: порождение и идентификация процессов, синхронизация работы предков и потомков, управление процессом с помощью сигналов, организация периодических процессов, обмен данными через канал, взаимодействие процессов на основе сообщений и разделяемой памяти, синхронизация процессов с помощью семафоров.

Основные задачи дисциплины «Программирование распределенных систем»:

1. Познакомить с технологиями взаимодействия распределенных приложений, построенных на языке Java.
2. Научить разрабатывать клиент-серверные приложения на языке Java с различными протоколами взаимодействия.

В результате изучения этой дисциплины студенты должны:

*Знать* — принципы и методы создания прикладных программных комплексов, работающих в распределенных компьютерных системах.

*Уметь* — проектировать клиент-серверное приложение и выбрать наиболее эффективную технологию взаимодействия в распределенной системе.

*Владеть* — навыками построения распределенных приложений на языке Java, используя протоколы UDP, TCP, RMI и CORBA.

Для дисциплины «Программирование распределенных систем» распределение часов по видам занятий следующее: лекции — 18 часов, лабораторные работы — 36 часов, курсовой проект — 18 часов. На лекциях рассказывается архитектуры сетевых приложений, взаимодействие приложений через датаграммы, потоковые сокеты, удаленный вызов процедур и технология CORBA. На лабораторных работах студенты разрабатывают классы на языке Java для серверного и клиентского приложений, используя различные протоколы взаимодействия. Курсовое проектирование является завершающим этапом в изучении дисциплины, в ходе которого осуществляется обучение применению полученных знаний и умений при решении комплексных задач, связанных с проектированием и программированием распределенных приложений на языке Java.

В заключении можно отметить, что данная программа отвечает требованиям ФГОС ВПО и нацелена на подготовку специалистов, обладающих фундаментальными знаниями и практическими навыками в области построения распределенных информационных систем с использованием ОС Linux и языка Java.

**И.А. Хахаев**

Санкт-Петербург, Институт международного бизнеса и права Национального исследовательского университета информационных технологий, механики и оптики

## Свободные программы в ИМБИП НИУ ИТМО

### Аннотация

Рассматривается внедрение СПО-решений в учебном процессе и организации деятельности в ИМБИП НИУ ИТМО, описываются организационные и методические проблемы.

Институт Международного Бизнеса и Права Санкт-Петербургского государственного университета информационных технологий, механики и оптики (ИМБиП) в значительной степени работает по программам дополнительного образования (второе высшее и повышение квалификации) по направлениям организации внешнеэкономических связей и таможенного дела, в также менеджмента (по договорам).

На конференции 2011 года<sup>1</sup> рассматривались варианты СЭД для внедрения. За прошедший год удалось внедрить следующие СПО-решения.

1. Для обеспечения дистанционного обучение по направлениям дополнительного образования (второе высшее, повышение квалификации) — Moodle<sup>2</sup>
2. В качестве СЭД — eGroupWare<sup>3</sup>
3. Для обеспечения учебного процесса (первое высшее, очная форма) — терминальный класс на основе ALT Linux 5.0 «Школьный терминал»

В декабре 2011 года подписано соглашение о сотрудничестве НИУ ИТМО и ОАО «НИИ программных средств» концерна «Сириус» (на базе которого создан Региональный центр компетенции СПО) в области инновационного развития и реализации ТП НПП, предусматривающее совместную подготовку магистров и аспирантов.

Ресурсы для развёртывания Moodle и eGroupWare предоставлены НИУ ИТМО. LMS Moodle установлена в варианте «хостинга», а СЭД eGroupWare установлена на виртуальной машине в ЦОД (виртуальная машина VMWare, гостевая ОС — ALT Linux 5.0 «Школьный сервер»), т. е. по определению работает как «облачный» ресурс.

По поводу решения «Школьный терминал» нужно отметить, то это действительно замечательное решение и данная сборка ALT Linux является (пока) субъективно лучшей.

Относительно eGroupWare следует отметить, что эта СЭД практически полностью соответствует тем задачам, которые планировалось решать с её помощью. Поддержка WebDAV (с дополнительным модулем mod\_dav для Apache) обеспечивает работу с хранилищем файлов без использования web-интерфейса.

<sup>1</sup>См. <http://freeschool.altlinux.ru/wp-content/uploads/2011/02/pereslavl-winter-2011.pdf>

<sup>2</sup>См. <http://ins.imbip.com/lms/>

<sup>3</sup>См. <https://tds.imbip.com/egroupware>

В качестве сервера БД используется MySQL, авторизация — SQL, поскольку требуется поиск контактов в адресных книгах по организациям (месту работы).

Пока используется версия eGroupWare 1.6, не используются модули myDMS (неподдерживаемый), Workflow, pERP и SiteManager (пока непонятны принципы работы). Ввод в действие и развитие модуля Workflow является темой магистерской работы.

Переход на eGroupWare 1.8 сдерживается наличием незакрытых багов<sup>4</sup>, которые сильно портят впечатление от системы.

Среди проблем перехода на СЭД следует отметить трудность преодоления привычки редактирования файлов с «сетового диска» напрямую без создания локальных (или каких-то других) копий документов. В некоторой степени проблема смягчается подключением ресурса WebDAV в качестве «сетового диска».

О результатах соглашения о сотрудничестве ОАО «НИИ ПС» и Центра компетенции СПО с НИУ ИТМО планируется доложить в 2013 году.

Ирина Рогожкина, Анатолий Кушниренко  
Москва, НИИСИ РАН, МГППУ

Проект: Пиктомир <http://www.piktomir.ru/>

## **Пиктомир: опыт обучения азам программирования старших дошкольников**

### **Аннотация**

Несколько лет назад в НИИСИ РАН была разработана среда ПиктоМир<sup>1</sup>, позволяющая составлять программы детям дошкольного возраста, не умеющим или не слишком любящим читать и писать. Вместо текстовых команд в ПиктоМире используются пиктограммы, из которых дети собирают на экране несложную программу, управляющую виртуальным Роботом. В ПиктоМир встроены шесть циклов задач, которые, теоретически, дают педагогу возможность познакомить детей с такими понятиями последовательного программирования как программное управление, подпрограммы, конструкции ветвления и повторения. Практически, однако, возникает вопрос о том, насколько эти

<sup>4</sup>См. <https://bugzilla.altlinux.org/buglist.cgi?quicksearch=eGroupWare>

<sup>1</sup><http://www.piktomir.ru/>

базовые понятия доступны и интересны детям дошкольного возраста. В работе изложены результаты эксперимента по обучению старших дошкольников азам программирования в среде ПиктоМир.

В 2010-2011 учебном году с детьми подготовительных и старших групп детского сада №1511 г. Москвы был проведен курс занятий, разработанный авторами доклада. В эксперименте участвовали 48 детей: из них 7 — моложе 6 лет (старшая группа), остальные 41 — старше 6 лет (подготовительная группа).

Курс состоял из 8 занятий по 25 минут. Они проводились раз в неделю в подгруппах из 6 человек. На первых трех занятиях дошкольникам предлагалось составить простые линейные программы. Последующие занятия были посвящены введению циклов (повторителей) и подпрограмм.

После проведения первых же занятий стало понятно, что в базовой версии Пиктомира не хватает заданий на тренировку навыка создания линейных программ и были добавлены 10 новых циклов заданий.

Непосредственная работа детей на компьютере занимала 10 минут из 25 минут каждого занятия, что было обусловлено как санитарными нормами, так и возможностями детей, многие из которых уставали к концу 10-минутной игры на компьютере. Для занятий без использования компьютера нами были разработаны несколько методических приемов и игр:

- *Игры в Робота и Капитана*: один ребенок изображает Капитана, отдающего команды, а другой — выполняющего их Робота. текстовые выделения курсивом: *nnnn*;
- *Использование математических корабликов*. Центральной проблемой для дошкольников оказалось освоение понятия программного управления. Этап составления программы был понятен всем детям. А вот процесс пошагового выполнения программы вызывал у детей затруднения. Дело в том, что процесс выполнения каждой команды линейной программы имеет две стадии, две грани: 1) работа с исполнителем — выдача Роботу очередной команды и 2) работа с программой — мысленный перевод этой команды из разряда еще не исполненных в уже исполненные. В то время как первая стадия наглядна и при компьютерном представлении и в игровом режиме (когда роль Робота выполняет один из детей), вторая стадия — работа с программой — визуализируется недостаточно четко. Для визуализации

процесса исполнения программы мы решили использовать «математические кораблики», с помощью которых детей учат считать в начальной школе. На фишки были наклеены стрелочки с командами «вперед», «направо» и «налево». Программа, которую нужно было выполнить, выкладывалась в правых корабликах. Выполненные команды перемещались в левую сторону. С помощью этого нехитрого приспособления работа с программой получила наглядное материальное воплощение.

- *Раскрашивание клетчатых полей:* в соответствии с заданной программой ребенок должен правильно раскрасить клетчатое поле. В этих заданиях помимо привычных в Пиктом мире команд «ВПЕРЁД», «НАЛЕВО», «НАПРАВО», использовались команды «ЗАКРАСИТЬ КРАСНЫМ», «ЗАКРАСИТЬ СИНИМ», «ЗАКРАСИТЬ ЖЕЛТЫМ». Эти задания были направлены на тренировку навыка пошагового выполнения программы.
- *Разрезание листа с программой:* детям предлагалось разрезать длинный лист бумаги с нарисованными на нем пиктограммами команд на одинаковые кусочки и заменить исходную линейную программу программой с циклом-повторителем. Дошколята складывали одинаковые куски программы в стопку, пересчитывали их и выбирали правильный повторитель. Таким образом, идея короткой записи программы с помощью цикла получила наглядное воплощение.

После 8 занятий, все дети научились создавать линейные программы, а все 6-летки научились использовать циклы и подпрограммы. Из 5-летних детей освоить циклы и подпрограммы, сумели только двое.

Результаты эксперимента доказывают возможность обучения дошкольников приемам настоящего «серьезного» программирования, то есть отсутствие (в терминологии Пиаже) каких-либо физиологически обусловленных барьеров при освоении основных алгоритмических конструкций в возрасте 6-6.5 лет. Что касается 5-леток, мы пришли к очевидному выводу, что работу с ними лучше начинать с короткого курса «Введение в программирование», посвященного линейным программам.

В настоящее время мы проводим второй, более продолжительный цикл занятий с детьми 6-летнего возраста, надеясь достичь стопроцентного освоения детьми конструкций ветвления и повторения.

Хачко Д. В., Яковлев В. В, Субоч Н. М., Джелядин Т. Р.,  
Ройтберг М. А., Кушниренко А. Г., Леонов А. Г.

Москва, Пущино, НИИСИ РАН

Проект: Кумир <http://niisi.ru/kumir>

## Кумир 1.9. Практикумы и исполнители. Средства интенсификации обучения

### Аннотация

Сообщение представляет систему программирования Кумир 1.9. Особое внимание будет уделено системе практикумов, состоящей из двух компонент ученической (системе выполнения практикумов), интегрированной в КуМир, и учительской (системе подготовки практикумов), которая реализована в виде отдельного модуля. Практикум позволяет учителю зафиксировать свою методику в виде серии заданий, с которой каждый ученик может работать самостоятельно, в частности, выполняя самопроверку. В заданиях практикумов могут использоваться различные исполнители, которые поддерживаются системы Кумир.

Кумир 1.9 — последняя к настоящему моменту версия системы семейства Кумир 1.x. Главным отличием этой версии от версии Кумир 1.7 является наличие системы поддержки практикумов. Практикум включает в себя:

1. систему заданий, которую должен выполнить ученик;
2. методику, которая описывает рекомендованный порядок выполнения заданий.

Задание включает поясняющий текст, а также шаблон программы. Обычно шаблон включает в себя заголовок программы, который требуется написать ученику, пред- и пост-условия, комментарии учителя. В более сложных случаях, по желанию разработчика, в шаблон могут включаться и элементы реализации требуемого алгоритма. Отдельные фрагменты шаблона могут быть защищены от редактирования или вообще скрыты от ученика.

К каждому заданию прилагается созданная разработчиком практикума тестирующая программа, которая позволяет проводить проверку того, как ученик выполнил задание, в автоматическом режиме. Эта программа, как правило, оформляется как скрытая от ученика

часть шаблона. Кроме того, разработчик готовит набор тестовых примеров. Выполнение задания считается успешным, если тестирование прошло успешно на всех тестовых обстановках для данного задания.

Описание практикума представляет собой xml-файл. При выполнении практикума учеником создаётся его личный файл («рабочая тетрадь»). Рабочий файл ссылается на файл курса, в нём хранятся все сведения о прохождении курса учеником.

В настоящее время завершается разработка специализированного редактора практикумов TaskEdit. Этот редактор работает отдельно от системы КуМир. Рекомендуется при создании практикума иметь одновременно открытые окно редактора практикумов, в котором создаётся практикум, и Кумира, в котором просматриваются внесённые в практикум изменения и редактируются шаблоны заданий и обстановки используемых исполнителей.

Главное окно редактора практикумов состоит из двух частей. В левой части окна производятся операции с «деревом заданий»: добавление новых заданий, удаление ненужных и изменение порядка заданий. Для удобства, поскольку задания часто бывают однотипными, добавление производится копированием существующего задания. В правой части окна производится редактирование текущего задания — изменение описания, выбор заготовки программы, исполнителей и обстановок к ним.

В. В. Яковлев, Д. В. Хачко, Т. Р. Джелядин, М. А. Ройтберг  
Москва, Пущино, НИИСИ РАН

Проект: КуМир <http://niisi.ru/kumir>

## КуМир 2.0: новые цели — новые решения

### Аннотация

Система обучения программированию КуМир (Комплект Учебных МИРов) была разработана в конце 80-х годов, она была реализована для существующего на тот момент парка компьютеров. К 2005–2011 г. была разработана кроссплатформенная система программирования КуМир 1.x, в которой на новом уровне были воспроизведены все возможности исходного КуМир'а, а также ряд новых возможностей, в частности, практикумы и новые исполнители. В сообщении представлен проект новой системы КуМир 2.0 и первые результаты реализации

этого проекта. В системе Кумир 2.0 будут сохранены все возможности Кумир 1.x. Основные отличия КуМир 2.0 от Кумир 1.x с точки зрения пользователя — увеличение скорости выполнения на порядок, с точки зрения разработчика — модульная архитектура.

## Постановка задачи

В ходе эксплуатации систем семейства Кумир 1.x стала понятна необходимость расширения возможностей системы. Основными из них являются следующие.

1. Возможность выполнения программ со скоростью, сопоставимой с промышленными языками программирования. Это требование обусловлено необходимостью решения ресурсоемких задач, как правило возникающих в сфере олимпиадного программирования. От Кумира требуется скорость работы если не уровня C/C++, то как минимум не хуже чем у Python, Perl или Ruby.
2. Возможность интеграции в системы автоматизированной проверки заданий на удалённом сервере, например, возможность работы в паре с системой ejudge[1].
3. Наличие системы поддержки практикумов с возможностью интеграции в существующие системы управления курсами, например Moodle[2].
4. Поддержка различных учебных языков программирования с использованием пользовательского интерфейса Кумир а также набора Кумир-исполнителей.
5. Возможность использования исполнителей системы[3] как самостоятельных учебных продуктов, не входящих в поставку Кумир.

## Ускорение выполнения программ

Существующая ранее реализация Кумир 1.x во время «компиляции» выполняет анализ корректности программ, написанных на алгоритмическом языке[4], и строит упрощённое представление программы для исполнителя. Такой подход является крайне неэффективным с точки зрения скорости выполнения и сохраняется как рудимент с тех времён, когда перед нами не стояла задача реализации эффективной системы выполнения. Для ускорения выполнения необходимо как

минимум упростить выполнение за счёт переноса части его функциональности в компилятор, а для достижения большей скорости выполнения — получать выполняемый машинный код.

### **Возможность работы без пользовательского интерфейса**

В 2010 году для апробации проведения компьютерного ЕГЭ с использованием языка «Кумир» [5, 6], нами была создана консольная версия интерпретатора. Данный инструмент был создан под конкретную задачу как ответвление от кодовой базы Кумира путём выбрасывания кода, зависящего от библиотек пользовательского интерфейса. В новой версии необходимо изначально предусмотреть возможность работы как с использованием графического пользовательского интерфейса, так и возможность независимого использования отдельных компонент системы.

### **Интеграция системы поддержки курсов в существующую инфраструктуру**

Во-первых, система поддержки практикумов должна иметь возможность работы с удалённым сервером Moodle [2], не используя сторонних программ, в частности – не требуя запущенного веб-браузера. Во-вторых, пользовательский интерфейс должен быть спроектирован таким образом, чтобы точкой входа пользователя в интерфейс системы был не текстовый редактор с пустой программой, а некоторая осмысленная сущность, например веб-страница со списком невыполненных заданий.

### **Интеграция сторонних компиляторов и интерпретаторов в систему Кумир**

Опыт работы с системой показал эффективность нашего подхода к редактированию программ: проверка программы при переходе на новую строку и отображение сообщений на полях редактора. Кроме того, на данный момент для многих языков программирования, в частности Паскаль, не существует кроссплатформенной современной среды разработки, пригодной для учебных целей. Мы ставим перед собой задачу восполнить данный пробел, создав тем самым полный единообразный комплект программного обеспечения для обучения

программированию в рамках школьной программы. Также представляется интересным возможность использования графических Кумир-исполнителей при написании программ на других учебных языках программирования.

## Исполнители

Система Кумир, помимо реализации своего языка программирования, содержит набор исполнителей, которые могут быть использованы без программирования. Для этого у большинства исполнителей имеется «пульт управления», предназначенный для решения задач [3]. Естественным образом возникает потребность в исполнителях с пультами, как отдельных программных продуктов, предназначенных для младшей школы. Соответственно, новая система должна быть спроектирована таким образом, чтобы данные исполнители можно было использовать как отдельные программы, устанавливаемые на компьютер, а ещё лучше – как веб-приложения.

## Реализация системы

Система Кумир 2.x состоит из набора *компонент*, между которыми могут быть зависимости, а также набора небольших программ – *стартёров*, которые загружают по некоторой предопределённой конфигурации «свой» набор компонент и передают управление стартовой компоненте. Данная архитектура реализована на основе системы PlugIn'ов библиотеки Qt4 [7]. Компонентами системы являются анализатор языка программирования Кумир, генераторы выполнимого байт-кода и С-кода, различные части пользовательского интерфейса а также исполнители. Стартёрами являются консольный компилятор или консольный исполнитель байт-кода, а также различные варианты пользовательского интерфейса для разных областей применения.

## Анализаторы программ

Анализатор выполняет проверку программ на корректность, предоставляет информацию и подсветке кода и расстановке отступов редактору, и, в случае языка «Кумир» – дерево разбора программы [8]. Помимо анализатора языка «Кумир», возможна поддержка других языков программирования. Так например, у нас имеется эксперимен-

тальная реализация анализатора для языка Паскаль, основанная на слегка модифицированном компиляторе FreePascal[9]. Модификация компилятора была обусловлена во-первых необходимостью работать в режиме «запрос-ответ» без перезапуска компилятора, а во-вторых – необходимостью получения дополнительных сведений (подсветка синтаксиса) об анализируемой программе.

## Генераторы кода

На данный момент у нас имеется два генератора кода, которые по ранее построенному дереву разбора Кумир-программы строят: а) выполнимый компактный байт-код (интерпретатор байт-кода является отдельной компонентой системы); б) C-программу для дальнейшей трансляции в машинный код средствами `gcc` или `clang`.

Использование байт-кода позволяет полностью контролировать процесс выполнения программы, в то время как выполнение программы, полученной через C – позволяет достигнуть максимальной скорости выполнения. В то же время, компиляция C-программы и запуск отдельного процесса требуют большего времени, поэтому использование данного способа генерации кода и выполнения оправдано только для ресурсоемких задач.

В настоящее время проводятся эксперименты по использованию библиотеки LLVM[10] для генерации кода. В случае, если получится создать надёжную кросс-платформенную реализацию контролируемого выполнения программы, то генератор LLVM-кода станет основным, а вместо использования C-кода для генерации машинного, будет использована технология JIT-компиляции[11].

## Пользовательский интерфейс

Пользовательский интерфейс будет существовать как минимум в двух различных вариантах: *классический* – для средней школы, и *профессиональный* – для старшей школы и олимпиадного применения. Классический вариант интерфейса максимально идентичен таковому в существующей системе Кумир 1.x, в то время профессиональный вариант будет предоставлять пользователю доступ ко всем новым возможностям системы. Также возможно создание интерфейсов, ориентированных под конкретные задачи, например подготовку

к экзаменам, используя систему курсов[12], или сдачу экзамена, используя для проверки удалённый сервер.

## Исполнители

На данный момент только отрабатывается технология для взаимодействия с исполнителями. Предполагается некоторую их часть реализовать в виде Web-приложений, отображаемых средствами WebKit. Кроме того, в планах обеспечить возможность работы с исполнителями используя другие языки программирования. Для этого планируется реализовать инструмент, генерирующий по формальному описанию исполнителя соответствующие оболочки для различных поддерживаемых языков программирования.

## Текущее состояние и дальнейшие планы

Релиз системы 2.0 предполагается, когда будут полностью реализованы и протестированы, то есть успешно пройдут все тесты из системы тестирования Кумир 1.x, следующие компоненты:

1. анализатор языка Кумир — уже реализован и успешно проходит тесты;
2. генератор и исполнитель байткода — уже реализованы и успешно проходят тесты;
3. стандартная библиотека функций языка Кумир — реализовано и протестировано всё, кроме файлового и терминального ввода-вывода.

На данный момент также реализован пользовательский интерфейс (в различных вариантах), однако он требует доработки (в том числе концептуальной), поэтому появится только в следующем релизе 2.1.

Также в дальнейших релизах предполагается реализация:

1. Выполнение программ средствами LLVM.
2. Создание варианта интерфейса, ориентированного на работу с системой курсов.
3. Реализация различных исполнителей.
4. Поддержка языка программирования Паскаль, и возможно, Python.

5. Поддержка новых целевых платформ для языка Кумир: *Lego NXT* [3, 13] и *Arduino*[14].

Исходные тексты системы успешно собираются на любом современном дистрибутиве Linux и свободно распространяются под свободной лицензией GPL 2.

SVN-репозиторий исходных текстов находится по адресу [15].

## Литература

- [1] А. В. Чернов *Система тестирования ejudge*. Вторая конференция «Свободное программное обеспечение в высшей школе». М.: AltLinux, 2007.
- [2] <http://moodle.org/>
- [3] А. Г. Кушниренко, А. Г. Леонов, М. А. Ройтберг, В. И. Хачко, Д. В. Хачко, В. В. Тарасова, В. В. Яковлев *Новые Миры в системе КуМир*. Пятая конференция «Свободное программное обеспечение в высшей школе». М.: AltLinux, 2010. с. 56–58.
- [4] *Информатика: 7–9 кл.: Учеб. для общеобразоват. учр.* А. Г. Кушниренко, Г. В. Лебедев, Я. Н. Зайдельман. М.: Дрофа, 2003. 335 с.
- [5] А. Г. Кушниренко, А. Г. Леонов, В. Р. Лещинер, Д. А. Путищев, П. С. Шестаков *Структура программного обеспечения для проведения ЕГЭ по информатике и ИКТ в компьютеризированной форме*. Шестая конференция «Свободное программное обеспечение в высшей школе». М.: AltLinux, 2011. с. 21–23.
- [6] А. Г. Кушниренко, А. Г. Леонов, М. А. Ройтберг *Принципы выбора языков и сред программирования при проведении ЕГЭ по информатике и ИКТ в компьютеризированной форме*. Шестая конференция «Свободное программное обеспечение в высшей школе». М.: AltLinux, 2011. с. 24–26.
- [7] <http://qt.nokia.com/>
- [8] *Компиляторы. Принципы, технологии, инструментарий*. А. Ахо, Р. Сети, Дж. Ульман. М.: Вильямс, 2008. 768 с.
- [9] <http://freepascal.org/>
- [10] <http://llvm.org/>
- [11] Hans Wennborg *Emulator Speed-up Using JIT and LLVM*. Master's Thesis, Lund University, Lund, Sweden, Jan. 2010.

- [12] Д. В. Хачко, Д. П. Кириенко, А. Г. Кушниренко, А. Г. Леонов, М. А. Ройтберг *Поддержка курсов в системе КуМир*. Шестая конференция «Свободное программное обеспечение в высшей школе». М.: AltLinux, 2011. с. 19–21.
- [13] <http://mindstorms.lego.com/en-us/Default.aspx>
- [14] <http://www.arduino.cc/>
- [15] <http://lpm.org.ru/svn/kumir2/>

М. А. Ройтберг  
Москва, НИИСИ РАН

## Новые тенденции преподавания информатики в школе

### Аннотация

На конференции «Свободное программное обеспечение в высшей школе — 2011» был представлен доклад об апробации компьютерного ЕГЭ, проведённой в конце 2010 г. За истёкший год в школьной информатике произошёл ряд интересных событий, в частности, была разработана программа по информатике для основной школы, соответствующая новым образовательным стандартам, произошли изменения в корпусе заданий ЕГЭ по информатике. В сообщении будет представлен обзор новых тенденций в развитии школьной информатики и о связи этих тенденций с высшей школой.

В докладе будет представлен анализ модели ЕГЭ по информатике 2012 г. Эта модель является преемственной по отношению к модели 2011 г., но имеет ряд важных отличий. Если говорить о тенденциях, то они состоят в следующем:

1. увеличение количества заданий, проверяющих умение ученика понимать представленный ему алгоритм, анализировать результаты выполнения алгоритма без выполнения полной трассировки;
2. снижение «арифметической нагрузки» на ученика, снижение риска случайных ошибок;
3. введение новых типов задач, в частности, связанных с анализом графов.

Среди задач, стоящих перед разработчиками ЕГЭ, можно отметить необходимость обновления корпуса задач, относящихся к темам, связанным с использованием информационно-коммуникационных технологий.

В докладе будет обсуждена связь тенденций в развитии ЕГЭ по информатике и разрабатываемых в настоящее время новых федеральных государственных стандартов для средней школы, а также с потребностями высшей школы.

**Владимир Николаевич Лукин, Лев Николаевич Чернышов**  
Москва, Московский авиационный институт, Государственный университет  
Министерства финансов РФ  
<http://www.gumf.ru>

## **Опыт использования приложения с открытым кодом StarUML в системе дистанционного обучения MOODLE с виртуальным практикумом**

### **Аннотация**

Для преподавания дисциплин по проектированию программных систем в высшей школе одними из лучших считаются СПО-продукты StarUML и ArgoUML. В рамках курсов повышения квалификации по направлению «Мастер информационных технологий», организованных НИИ информационных технологий и телекоммуникаций («Информика»), разработаны учебные программы и проведён курс обучения дисциплине «Технологические основы проектирования программных систем». В качестве основного инструмента выбрана система StarUML. Благодаря хорошей документированности системы и открытым кодам, удалось успешно адаптировать её к среде дистанционного обучения MOODLE.

В циклах дисциплин по программированию большинства направлений подготовки студентов обязательно встречаются темы, в которых рассматриваются технологии проектирования программных систем и, в частности, инструменты проектирования, базирующиеся на языке моделирования UML. Немаловажно, какой конкретно инструмент выбирается. Это должен быть простой, эффективный и желательно бесплатный инструмент, который студенты (особенно в условиях дистанционного обучения) могли бы установить на свои домашние

компьютеры. Иногда такие инструменты встроены в среды программирования, например, *Delphi*, *Visual Studio* и т.п. Но желательно иметь отдельный инструмент. Одно из требований — соблюдение стандартов. Среди таких инструментов наиболее подходящие СПО-продукты — *StarUML* и *ArgoUML*.

В рамках курсов повышения квалификации по направлению «Мастер информационных технологий», организованных НИИ информационных технологий и телекоммуникаций («Информика»<sup>1</sup>), разработаны учебные программы и проведён курс обучения дисциплине «Технологические основы проектирования программных систем».

Особенностью широко известной обучающей среды MOODLE, используемой в этих курсах обучения, является наличие виртуального практикума: специально разработанного сотрудниками «Информики» ПО. В нём слушатели курсов могут использовать программные продукты в режиме удалённого доступа к серверу, установленному на площадке «Информики». Одним из таких продуктов и является *StarUML*. Таким образом, слушатели работают с программным средством через браузер. Это позволяет использовать и проприетарные продукты без нарушения лицензий. Но возможность для слушателя установки продукта на свой компьютер весьма желательна.

*StarUML*, как и многие другие систем с открытым кодом, характеризуется простотой интерфейса. Следует отметить и стабильность системы: в течение нескольких последних лет в ней не производится никаких изменений. Конечно, можно говорить о потере интереса разработчиков к продукту, но с точки зрения обучения это положительное качество. Появления бесконечной цепочки версий проприетарных продуктов часто диктуется чисто коммерческими соображениями. Для обучения в любом продукте достаточно иметь базовые возможности, а излишества только затрудняют освоение технологий.

Для *StarUML* были произведены дополнительные настройки, упрощающие интерфейс и ограничивающие число типов моделей и стереотипов элементов. Все настройки производились путём корректировки XML-файлов конфигурации. Благодаря хорошей документированности системы и открытым кодам, удалось успешно адаптировать её к среде дистанционного обучения MOODLE. Проведённая апробация курса на небольшой группе слушателей подтвердила правильный выбор инструмента.

---

<sup>1</sup>[www.informika.ru](http://www.informika.ru)

Н. С. Лагутина, А. М. Васильев, И. В. Парамонов  
Ярославль, Ярославский государственный университет им. П. Г. Демидова

## **Обучение студентов объектно-ориентированному программированию на основе архитектурного шаблона Модель—Вид—Контроллер**

### **Аннотация**

В докладе обсуждается структура курса по изучению объектно-ориентированного программирования. Основой обучения является архитектура Модель—Вид—Контроллер. Свободная библиотека Qt используется как удобный инструмент реализации этой архитектуры.

Весьма поверхностное освоение некоторых технических приёмов объектно-ориентированного программирования позволяет создавать несложные программы в визуальных средах. Чаще всего качество таких программ не удовлетворяет требованиям надёжности и сопровождаемости. Для создания сложного приложения необходимо глубокое понимание принципов объектно-ориентированного программирования, устройства объектно-ориентированной программы, технологии создания программных систем.

При обучении технологиям программирования преподаватель сталкивается с рядом проблем. Структура изучаемых курсов в вузе зачастую такова, что дисциплины связанные с проектированием приложений и программной инженерией изучаются на старших курсах, поэтому при разработке программы студент начинает непосредственно с написания кода, все остальные части жизненного цикла ПО рассматриваются им после, а часто никогда. Не редко курс по изучению языка программирования или конкретной библиотеки для разработки определённого вида программных продуктов сводится к описанию технических особенностей изучаемого предмета, оставляя на заднем плане общие принципы построения программ: архитектуру, модифицируемость, юзабилити. С другой стороны, многие теоретические курсы по информатике построены так, что почти не позволяют студентам увидеть реальное применение изучаемых технологий. Ещё одной проблемой является отсутствие взаимосвязи между отдельными курсами читаемыми разными преподавателями.

В данной статье рассматривается подход к изложению материала в рамках предмета «Языки и методы программирования», изучаемого на втором курсе направления «Прикладная математика и информатика» на факультете информатики и вычислительной техники ЯрГУ.

В первой половине курса излагаются общие принципы объектно-ориентированного программирования и сразу же демонстрируется их применение в конкретном языке и библиотеке. В нашем случае используется язык C++, что обусловлено в основном историческими причинами и связностью с предметами, преподаваемыми на старших курсах. Перед изучением данной дисциплины студенты уже ознакомились с парадигмой структурного программирования и языком Си. Данная часть курса содержит описание инвариантных к языкам программирования понятий, приёмов, методов объектно-ориентированного программирования. Методическая направленность первой части заключается в овладении технологией объектно-ориентированного программирования. Основное внимание уделяется понятию класса: созданию пользовательских классов и использованию их и уже разработанных классов на примере стандартной библиотеки STL.

Вторая часть курса посвящена созданию приложений с графическим пользовательским интерфейсом. Основная идея второй половины курса: рассмотреть создание приложений на основе архитектуры Модель—Вид—Контроллер. Различные средства библиотеки представляются как способ реализации этого шаблона программирования.

Для второй части курса выбрана библиотека Qt, так как на её основе весьма удобно демонстрировать реализацию в системе программирования наследования, инкапсуляции, полиморфизма.

Цикл лекций, посвящённый проектированию и созданию приложений с графическим пользовательским интерфейсом, объединён идеей создания приложения, хранящего и обрабатывающего данные с постепенным его усложнением и добавлением новых возможностей, относящихся как к общей архитектуре подобных программ, так и к особенностям самой библиотеки Qt.

Один из самых важных моментов обучения связан с тем, что приложение можно написать «вручную», без использования дополнительного инструмента визуального программирования. При этом намного понятнее становятся взаимосвязи между классами и другими элементами программы. Достоинством Qt является и то, что практически вся функциональность приложения достигается за счёт создания

объектов соответствующих классов и вызова методов для настройки необходимых параметров, а логика этих действий достаточно простая.

Механизм сигналов и слотов, имеющийся в библиотеке Qt, позволяет быстро и просто связывать между собой отдельные объекты. Это даёт возможность на начальном этапе создания приложения сосредоточиться именно на его архитектуре, не углубляясь в механизм обработки событий, связанный с переопределением методов базовых классов.

Первоначальный вариант приложения явным образом реализует архитектуру Модель—Вид—Контроллер. Постепенное усложнение программы в рамках единой архитектуры позволяет быстро перейти к рассмотрению сложных классов самой библиотеки Qt, использующих эту архитектуру.

Часть лекций, описывающих детали разработки приложений, опубликована на сайте <http://qt.e-werest.org> в цикле статей «Архитектура Модель—Вид—Контроллер в Qt» [1].

Направленность описанного курса на овладение студентами навыков использования основ объектно-ориентированного программирования и построения графических пользовательских приложений в рамках архитектурных шаблонов позволяет в дисциплинах, изучаемых на старших курсах сосредоточиться на изучении более глубоких аспектов разработки ПО.

## Литература

- [1] Раздел учебные материалы на E-WeREST // <http://qt.e-werest.org/blog/tutorial/>

Станислав Фомин  
Москва, ИСПРАН, МФТИ, CUSTIS  
<http://discopal.ispras.ru>

## Efficient [E]-Learning for Advanced Topics. Эффективное дистанционное преподавание нетривиальных предметов

### Аннотация

Как эффективно преподавать сложные предметы старшекурсникам? Ежедневно сгонять работающих студентов и внештатных преподавателей из науки/индустрии в аудиторию — проблема. Да, решение вроде бы очевидно: дистанционное образование. Но как? Полно сложных и полностью бесплодных систем, многое ещё и дорого... Наш успешный опыт по преподаванию курса по теории сложности и алгоритмам для 6 курса МФТИ, основан только на open-source решениях и бесплатных интернет-сервисах.

К старшим курсам ВУЗов загрузка студентов падает, практически все из них начинают совмещать работу с учёбой, а нетривиальные предметы часто преподаются внештатными специалистами из науки или производства. И предлагаемый классический формат «все собираются еженедельно на часовые лекции в необорудованной аудитории» фантастически неудобен и студентам и преподавателям. Тратится куча времени на поездку туда-обратно и разворачивание оборудования, плюс неудобные аудитории и ограниченный слот времени: всё это роняет КПД процесса ниже плинтуса. В результате большинство разумных (и работающих) студентов игнорируют лекции, обучение превращается в судорожное чтение методичек за несколько дней до экзамена, после чего всё мгновенно забывается. Преподаватели же страдают от малой эффективности лекций для остатка группы и бессилы предложить интересные форматы обучения остальным.

Теоретически, это идеальная ситуация для дистанционного обучения:

- Большинство студентов работают в офисах и даже в рабочее время могут участвовать в онлайн занятии не отходя компьютера.
- Тем, кто всё же не сможет участвовать, можно предложить записи.

- Для всех — интересную образовательную коллаборацию.
- Преподаватель же получает свободу, экономию времени, и возможность усилить и углубить курс.

Вопрос как? Рынок и интернет забит платными и бесплатными системами для e-learninga, и чего только нет — и «виртуальные 3D-миры», и SCORM-сертифицированные системы от солидных вендоров, годные максимум для обучения «технике безопасности» или «регламентам продаж», системы «образовательной бюрократии», . . . в общем, всё это «не о том». А нужно максимально эффективно сделать:

- Эффективные и интерактивные онлайн-занятия, когда можно показать всё — от формул на слайдах, до лайвкодинга.
- Получить качественные записи, посмотрев которые действительно можно выучить предмет, и которые даже выигрывают у иллюстрированной книги (запись бытовой видеокамерой таковой не является).
- Впрочем, также получить и хорошую электронную книжку.
- Тренажёр для самоконтроля и средство «отсечки» для ленивых студентов, склонных к «прорыву на халяву».
- Максимально простая и в то же мощная среда для совместной коллаборации:
  - Решение задач.
  - Обсуждение тем.
  - Создание контента (текст-формулы-иллюстрации и т.п.).
- Справедливые и мотивирующие метрики активности.

И да, чтобы всё это было:

- Бесплатно (не хватало ещё платить, когда «информация хочет быть бесплатной»©. . .)
- Мультиплатформенно — среди студентов всё больше пользователей Linux и Mac.
- И чтобы опыт всех этих инструментов был полезен студентам даже отдельно от преподаваемого курса.

Конкретно в нашем случае, это курс по алгоритмам и теории сложности для студентов 6 курса МФТИ, и мы решили все эти задачи используя:

- Open-source софт (включая свои наработки). Конкретно: MediaWiki с десятком расширений проекта MediaWiki4IntraNet<sup>1</sup>, система записи и монтажа SeminarAssembler.
- Бесплатно доступные интернет-сервисы, которые хоть и не предназначены напрямую для e-learninga, но их комбинация отлично обеспечила весь процесс (Google Docs, Skype, join.me, yatv.ru...).
- Ставший общедоступным безлимитный интернет.

Получился отличный результат, и кроме довольных, обученных в индивидуально-комфортном ритме студентов, получились ещё и дополнительные материалы видеолекций, которые освободят преподавателя от ежегодного самоповторения в пользу новых и интересных тем.

Разрушены некоторые мифы:

- о необходимости eye-to-eye контакта, о важности личного присутствия, и голосового общения. На самом деле:
  - Все гораздо комфортней чувствуют себя не видя друг друга. Это позволяет не беспокоиться о качестве обстановки в комнате и даже наличии одежды.
  - Наиболее эффективный канал общения — преподаватель общается голосом, студенты пишут вопросы и замечания текстом — нет проблем «перебивания».
  - Самое важное, для чего нужно выделить максимум площади «образовательного окна» — это видео экрана, с слайдами и ad-hoc рисунками лектора.
- О нужности моделирования метафор стандартного процесса («аудитория», «доска», ...). На самом деле:
  - Манипуляция метафорами (сейчас надо использовать инструмент «презентация», теперь надо использовать инструмент «доска») — только мешает.
  - Визуальная часть может быть любой — может потребовать показать не только слайды, но и живой код, документы в браузере, видеoaнимацию, работающие программы и т. п.
  - И поэтому, нужен только один инструмент — эффективная и качественная трансляция экрана преподавателя. И всё!

---

<sup>1</sup><http://wiki.4intra.net>

- О том, что «классическая доска» лучше заготовленных слайдов.
  - На самом деле, можно совместить лучшее обоих подходов — интерактивность визуализации от «доски» и качество формул и иллюстраций в заготовленных слайдах: основную визуализацию несут заготовленные слайды, а преподаватель, легко, простыми штрихами, акцентирует внимание на важном, или дорисовывает неочевидные аудитории моменты.
- А также многие другие. И для всего найдены простые и бесплатные решения.

Да, всё, о чём мы предлагаем рассказать, можно совершенно свободно использовать в своём учебном процессе.

Евгений Алексеев, Оксана Чеснокова, Иван Филь, Екатерина Горбенко, Ксения Остапенко, Михаил Савченко  
Донецк, Украина, Донецкий Национальный Технический Университет  
<http://openmp.ucoz.ru>

## **Свободное программное обеспечение. Опыт внедрения на кафедре вычислительной математики и программирования Донецкого национального технического университета**

### **Аннотация**

Рассмотрен опыт использования свободного программного обеспечения для организации учебного процесса и исследовательской деятельности на кафедре вычислительной математики и программирования Донецкого национального технического университета.

Организация учебного процесса по информационным дисциплинам в техническом университете имеет ряд особенностей. Важную роль здесь играет выбор программного обеспечения (ПО). Особую роль при обучении в техническом университете играют компиляторы и математические программы. Серьёзную альтернативу проприетарным программам составляют свободные приложения, которые особенно бурно развиваются в последние годы.

В Донецком национальном техническом университете (ДонНТУ) на кафедре «Вычислительная математика и программирование»

(ВМиП) с 2005 года авторы используют свободные программы, как в учебном процессе, так и в научных исследованиях.

В 2005 году авторы познакомились с пакетом Scilab (версия 3.x) и дистрибутивом ALT Linux 3. На базе пакета Scilab был разработан курс информатики для студентов технических специальностей. В этом курсе студенты изучали программирование, решение инженерных и математических задач с помощью пакета Scilab. На базе этого курса была написана книга: «Е. Р. Алексеев, О.В. Чеснокова, Е. А. Рудченко Scilab: Решение инженерных и математических задач»<sup>1</sup>, которая была издана в 2008 году при технической и финансовой поддержке компании ALT Linux. Авторам известно, что ею пользовались и продолжают пользоваться во многих университетах России, Украины и Белоруссии.

В 2007-2008 годах на кафедре ВМиП на базе довольно старых компьютеров (Celeron 600 Мгц, ОЗУ — 512 Мб, жёсткие диски 10-40 Гб) был развёрнут первый и единственный в университете компьютерный класс свободного ПО. В качестве операционных систем используются ОС Debian, ОС ALT Linux Школьный. Были установлены следующие прикладные программы OpenOffice.org, TeX, geany, g++, gfortran, Free Pascal, Lazarus, Scilab, qtOctave+Octave, Maxima, Mozilla FireFox, Mozilla Thunderbird, pidgin и dia. Этот класс функционирует до сих пор. В 2008-11 гг. в нём проходили занятия по повышению квалификации преподавателей ДонНТУ, с 2007 года в нём проходят лабораторные работы студентов электротехнического факультета по курсу «Вычислительная техника и алгоритмические языки», а также один из авторов преподаёт там все информационные дисциплины студентам инженерно-экономического колледжа.

Начиная с 2007 года началось постепенное внедрение свободно-программного обеспечения при изучении программирования студентами электротехнического факультета, для чего использовались текстовый редактор geany, компиляторы g++, fpc, среда визуального программирования Lazarus и редактор диаграмм dia. Результатом этой работы стало учебное пособие по алгоритмизации и программированию на C++<sup>2</sup>, а также учебник по Free Pascal и Lazarus<sup>3</sup>, который также был издан при технической и финансовой поддержке

---

<sup>1</sup><http://www.altlinux.org/Books:Scilab>

<sup>2</sup>[http://teacher.ucoz.net/index/lekci\\_po\\_s/0-4](http://teacher.ucoz.net/index/lekci_po_s/0-4)

<sup>3</sup><http://www.altlinux.org/Books:FreePascal>

компании ALT Linux. На основе этой книги было написано учебное пособие, которое в 2010 году получило гриф МОН Украины.

Начиная с 2009 года авторы начали активно использовать пакет GNU Octave в учебном процессе и в своих научных исследованиях. Результатом этого стал учебник по GNU Octave <sup>4</sup>, который, как надеются авторы, будет также издан при содействии компании ALT Linux.

В 2010/2011 на практических занятиях по курсу «Высшая и прикладная математика» на кафедре ВМиП применялась свободная программа Maxima. В качестве учебного пособия использовалась прекрасная, но до сих пор к сожалению не изданная книга Е.А. Чичкарёва «Компьютерная математика с Maxima. Руководство для школьников и студентов» <sup>5</sup>. Параллельно с классическими способами принятыми у математиков, студенты решали математические задачи на персональном компьютере с использованием Maxima. И преподавателю, и студентам этот опыт показался очень интересным. Грамотное применение Maxima в учебном процессе обеспечило бы повышение фундаментальности математического и технического образования, могло бы содействовать подлинной интеграции процесса образования. Но, к сожалению, этот опыт не нашёл поддержки в нашем университете.

Ещё одним направлением исследований на кафедре ВМиП становится параллельное программирование с использованием технологии OpenMP и компилятора g++. Планируется включить в курс «Вычислительная техника и алгоритмические языки» для студентов электротехнического факультета лекции по принципам создания параллельных программ на C++ с использованием технологии OpenMP.

На данный момент авторы исследовали распараллеливание алгоритмов умножения матриц и решения систем линейных алгебраических уравнений (СЛАУ) с помощью технологии OpenMP. Использовался следующий ПК: материнская плата MSI H67MA-E45; ОЗУ — 4Гб; процессор Intel Core I5 (4 ядра) с тактовой частотой 3.3 Ггц; 32-х разрядная ОС Linux Mint с рае ядром, компиляторы g++ 4.5 и gfortran 4.5. Тестирование умножения матриц проводилось на матрицах размером от 200x200 до матриц размерности 4000x4000. При использовании 4-х ядер время счёта уменьшилось в 3.8 раза (по сравнению с использованием одного ядра). При решении СЛАУ (от 500

---

<sup>4</sup><http://gnu-octave.narod2.ru>,<http://ea.donntu.edu.ua:8080/jspui/handle/123456789/2934>

<sup>5</sup><http://www.altlinux.org/Books:Maxima>

до 11000 уравнений) различными методами с использованием технологии OpenMP время счёта уменьшалось в 3 — 3.5 раза.

В 2011 году на международной студенческой конференции «СОВРЕМЕННЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ И НАУЧНЫХ ИССЛЕДОВАНИЯХ (СИТОНИ-2011)», проводимой кафедрой ВМиП, первый раз работала секция «ИСПОЛЬЗОВАНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ОБРАЗОВАНИИ И НАУКЕ». Мы надеемся продолжить этот опыт и приглашаем студентов, аспирантов и молодых учёных вместе с их научными руководителями 4-5 октября 2012 в г. Донецк на 3-ю международную студенческую конференцию СИТОНИ-2012, на которой будет работать секция, посвящённая свободному программному обеспечению. К началу конференции будет издан сборник трудов. Заявки на участие присылайте до 15 мая 2012 на e-mail sitoni2011@gmail.com, срок приёма докладов до 1 июля 2012.

**Иван Филь, Евгений Алексеев**

Донецк, Украина, Донецкий национальный технический университет

## **Об использовании свободного программного обеспечения для решения задач электротехники**

### **Аннотация**

В докладе рассмотрены возможности применения свободного программного обеспечения для решения задач электротехники.

На сегодняшний день на электротехническом факультете Донецкого национального технического университета при решении задач электротехники применяется проприетарное программное обеспечение. В докладе рассмотрена альтернатива применения свободного программного обеспечения для этих целей.

При решении электротехнических задач компьютеры широко применяются в курсах «Теоретические основы электротехники», «Математические задачи энергетики» для решения дифференциальных уравнений, систем линейных и нелинейных уравнений с комплексно сопряжёнными коэффициентами, в курсах «Электрические машины», «Электротехнические материалы», «Основы релейной защиты и автоматики» для построения и обработки характеристик, а также

для построения чертежей и 3D-моделирования курсах «Электрические машины», «Инженерная графика».

Для различных вычислений и построения графиков на электротехническом факультете нашёл применение Mathcad, для построения чертежей и 3D-моделирования — Autocad и КОМПАС и для построения электрических схем (схем электрических систем, релейной защиты и автоматики, преобразовательной техники) — Autocad и Visio.

Mathcad можно полностью заменить пакетами Scilab и GNU Octave при решении дифференциальных уравнений, систем линейных и нелинейных уравнений, при построении графиков и аппроксимации характеристик, а при моделировании переходных и установившихся режимов в электрических цепях и расчётах токов коротких замыканий в электрических системах можно использовать Xcos.

На рис. 1, 2 представлены схема RLC-цепи и графики переходного процесса (напряжения и тока), полученные в Xcos.

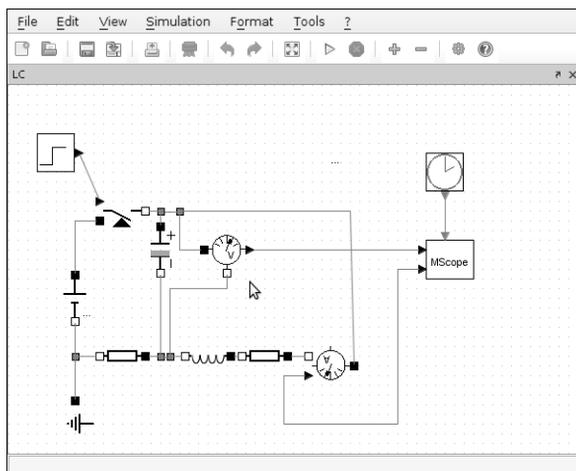


Рис. 1: Схема RLC-цепи

При решении сложных математических задач также следует обратить внимание на Sage или его On-line версию — <http://sagenb.org>.

При построении чертежей и векторных диаграмм рекомендуется двухмерная САПР Qcad Community Edition с открытым исходным

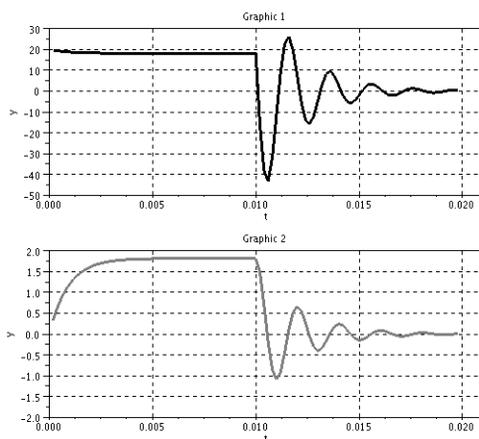


Рис. 2: Зависимости напряжения на ёмкости и тока на индуктивности

кодом. Qcad имеет инструменты для черчения точек, прямых, дуг, окружностей, эллипсов, ломаных, кривых, текста, размеров, штриховки и сплошной заливки. Он также может работать с растровыми изображениями и имеет множество других инструментов. Данный пакет работает с файлами формата DXF (Autocad's Drawing Interchange format), которые может открыть практически каждая САПР. Также Qcad имеет возможность экспорта чертежей в формат растровых изображений. В отличие от Autocad и КОМПАС имеет менее удобный интерфейс и не включает в себя 3D-моделирование. В связи с чем Qcad может частично заменить КОМПАС и Autocad, однако стоит отметить, что Qcad является одной из немногих полноценных 2D САПР с открытым исходным кодом. На рис. 3 представлен чертёж паза статора синхронного генератора, выполненный в Qcad.

Для построения электрических схем возможна полная замена Visio и Microsoft Office Word на редактор диаграмм «Dia». Данный пакет позволяет экспортировать диаграммы в форматы: EPS (Encapsulated PostScript), SVG (Scalable Vector Graphics), DXF (Autocad's Drawing Interchange format), CGM (Computer Graphics Metafile defined by ISO standards), WMF (Windows Metafile), PNG (Portable Network

Graphics), JPEG (Joint Photographic Experts Group), VDX (Microsoft's XML for Visio Drawing).

Основным недостатком Dia является отсутствие возможности вставки объектов других приложений.

Этого недостатка лишена программа Libre (Open) Office Draw, которую авторы рекомендуют использовать при рисовании блок-схем и других не очень сложных объектов.

У любого инженера и научного работника возникает необходимость в оформлении результатов решения задач. В качестве текстового процессора рекомендуется использование OpenOffice.org Writer вместо Microsoft Office Word, в качестве редактора презентаций — OpenOffice.org Impress вместо Microsoft Office Power Point, в качестве редактора формул — OpenOffice.org Math, для построения таблиц и обработки табличных данных — OpenOffice.org Calc вместо Microsoft Office Excel и Mathcad.

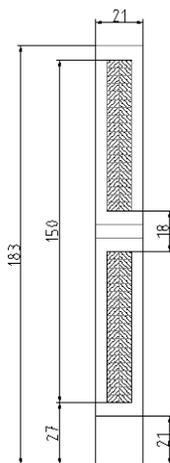


Рис. 3: Чертеж паза статора синхронного генератора

Также OpenOffice.org Calc, как и Octave, и Scilab, может служить инструментом для построения графиков, диаграмм и характеристик, обработки экспериментальных данных. OpenOffice.org может работать с документами в формате Microsoft Office.

Ещё одной альтернативой для подготовки документов является использование TeX, который уже давно де-факто стал стандартом у математиков как за рубежом, так и на территории бывшего СССР. Однако его освоение требует значительного времени, что компенсируется высоким качеством получаемых документов. Как альтернативу TeX можно рассмотреть приложение LyX.

Таким образом, по мнению авторов, при решении задач электротехники можно полностью отказаться от применения таких проприетарных пакетов Microsoft Office, Mathcad и Visio. Существуют выше перечисленные полноценные свободно распространяемые аналоги. А для построения чертежей при отсутствии необходимости использования 3D-моделирования рекомендуется использование Qcad Community Edition. Применение свободного программного обеспечения даст возможность студентам как будущим инженерам и учёным воспользоваться его достоинствами, а также повысить уровень компьютерной грамотности, расширить кругозор и выбирать самое подходящее программное обеспечение при решении той или иной задачи.

Михаил Быков

Москва, diglossa.ru

## SEO-дружелюбное web-приложение одной страницы — diglossa.ru

### Аннотация

Создание полностью SEO-дружелюбного приложения одной страницы, то есть полностью на ajax, причем не имеющего ни сервера приложения, ни веб-сервера. Этот результат получен благодаря соблюдению принципа «постепенного улучшения».

Версия 0.4 библиотеки diglossa.ru — приложение одной страницы. Все тексты подкачиваются по частям по мере необходимости по ajax-запросу. Обычно это вызывает сложности при индексировании сайта поисковыми машинами. Потому что поисковые роботы не умеют обрабатывать javascript и ajax-запросы. Однако предлагаемый подход позволяет создать полностью SEO-дружелюбное приложение.

При этом диглосса не имеет ни сервера приложения, ни веб-сервера.

Звучит как-то сомнительно, для тех кто не в курсе. Дело в том, что v.0.4 библиотеки построена на технологии CouchDB [1]. А доступ к этой документо-ориентированной базе данных происходит по протоколу `http`, в REST-стиле. То есть строка ввода ресурса в браузере является консолью доступа к каучу. Другими словами, кауч сам является веб-сервером.

Я переписал диглоссу со связки Unicorn / Rails / Mysql на CouchDB всвязи с развитием другого проекта — создания морфологического анализатора Morpheus. В морфологии и грамматике все построено на деревьях, и вдобавок всякое правило имеет исключения, можно сказать, исключений больше чем правил. В этих условиях работа с реляционными базами превращается в каторгу. А документо-ориентированные базы подходят к этой задаче идеально. Но оказалось, что и сама диглосса ложится в эту структуру данных идеально, поскольку структура книги тоже дерево, и все они очень разные, т.е. с реляционной базой возникают те же проблемы.

Чтобы добиться от поисковиков адекватной работы, я применил следующий прием. Обычно ни одну страницу сайта перегружать не нужно. Однако, у всех содержательных страниц сайта можно посмотреть `html`, с помощью `ctrl-U`. При этом кауч отдаст все необходимые тексты, которые будут проиндексированы поисковиком. Используется метод `list` кауча. После этого включается скрипт, преобразующий страницу к тому же в точности виду, который должен быть обычно. Этот эффект видно при перезагрузке страницы по `ctrl-R`.

Этот подход является применением на практике принципа «постепенного улучшения», который называют даже «самым важным принципом веб-разработки» [2].

## Литература

[1] <http://wiki.apache.org/couchdb>

[2] [http://en.wikipedia.org/wiki/Progressive\\_enhancement](http://en.wikipedia.org/wiki/Progressive_enhancement)

Евгений Синельников  
Саратов, ООО «Этерсофт»

Проект: Кафедральный сервер исходных кодов <http://git.toiit.sgu.ru>

## Использование системы контроля версий Git в преподавании курсов, связанных с разработкой ПО.

### Аннотация

Обучающие курсы по разработке программного обеспечения очень часто не затрагивают особенности коллективной работы над кодом. В данном докладе представлен вариант применения технологий, используемых командой разработчиков ALT Linux для сборки пакетов, в рамках учебного сервера исходных кодов. На примерах показаны подходы и их эволюция по применению системы контроля версий Git для обучения в курсах Технологии программирования и Операционные системы.

Совместная работа над исходным кодом крайне важна при обучении основ программной инженерии. Умение читать уже написанный кем-то программный код формирует адекватное представление о программировании, как о коллективной работе. Возможность работы группами, при постановке реализуемой задачи, является дополнительным фактором мотивации.

В данной работе, в качестве основного примера, обосновывающего тезисы о совместной работе над кодом при обучении, рассматривается кафедральный сервер исходных кодов, использующий систему контроля версий Git. Сервер применяется на кафедре дискретной математики и информационных технологий факультета компьютерных наук и информационных технологий саратовского государственного университета для распространения примеров преподавателей и публикации выполненных заданий студентов.

### Организация сервера исходных кодов

Главной проблемой, с которой сталкиваются преподаватели при организации совместной работы над исходным кодом в рамках централизованных систем контроля версий (например, Subversion) состоит в необходимости доступа к серверу во время самостоятельной

работы студентов, а также индивидуального управления каждым отдельным проектом студента или творческой группы.

Соответственно, одним из важнейших элементов современного подхода к использованию систем контроля версий в образовательном процессе, является переход к распределённым системам контроля версий. Этот переход также позволяет избавиться от необходимости доступа к сети при самостоятельной работе над исходным кодом во время выполнения заданий.

В основной список свободных распределённых систем контроля версий входят следующие проекты:

- **Git** (сложен в освоении, популярен в современных крупных проектах);
- **Mercurial** (лёгко в освоении, подходит для небольших проектов и учебных задач);
- **Monotone** (относительно прост в освоении, непопулярен).

Выбор системы контроля версий определяется предпочтениями и навыками в использовании. Для сообщества пользователей и разработчиков GNU/Linux, выбор **Git** является естественным, поскольку многие проекты в GNU/Linux уже ведутся с применением **Git**, а многие переходят к использованию системы контроля версий, используемой для разработки ядра Linux.

После выбора системы контроля версий, вне зависимости от выбора реализации, возникает вопрос публикации результатов работы. Хотя большая часть проблем, связанных с публикацией результатов, в последние годы, решается, путём использования бесплатных сетевых служб, так называемых сервисов хостинга проектов свободного программного обеспечения:

- [code.google.com](http://code.google.com)
- [github.com](http://github.com)
- [gitorious.org](http://gitorious.org)
- и другие

Независимость уровня и качества работы от внутренней инфраструктуры образовательного учреждения, а также возможность выполнять всю работу по управлению проектами студентами самостоятельно, являются положительными чертами подхода по использованию бесплатных сервисов. Тем не менее, использование таких служб в учебных проектах имеет и определённые недостатки:

- сложность поиска заданий конкретного студента или группы;
- необходимость применения специальной схемы именования проектов в рамках глобального пространства имён службы хостинга проектов;
- отчуждённость независимых проектов от места обучения — проект не ассоциируется курсом, сохранность истории наработок не гарантирована.

Разрешение этих недостатков требует организации своего сервера. Для этого могут использоваться различные проекты, большая часть которых организована в виде web-сервисов, выполняющих следующие функции:

- Создание, удаление, учётных записей пользователей;
- Создание, удаление проектов для публикации своих результатов;
- Доступ к репозиториям проектов на запись по протоколу ssh.

Одним из вариантов такого сервиса, управляемого по протоколу ssh, является проект girar [2], на котором основан сервер `git.altlinux.org` [1], а также `git.etersoft.ru` [5]. Сервер `git.toiit.sgu.ru` [3] размещён в лаборатории теоретических основ информатики и её приложений и имеет следующие особенности по отношению к `git.altlinux.org`:

- кроме репозитория участников, сервер управляет общим для всех пользователей набором проектов, на котором хранятся примеры исходных кодов;
- создание, удаление и переименование репозитория проектов настроено, по умолчанию, на каталог `public` вместо каталога `packages`, предназначенного для сборки пакетов;
- регистрация пользователей и замена ключей выполняется с помощью модуля в системе управления проектами `project.toiit.sgu.ru` [4] на основе `Chiliproject` — ответвление проекта `Redmine` (данное решение находится в стадии отладки и реализуется силами одного из активных студентов, в виде одного из учебных проектов).

Рабочая сборка girar-сервера размещена в репозиториях компании Etersoft и доступна для тиражирования в решениях на базе ALT Linux Sisyphus. Исходный код модифицированного girar-сервера также доступна на сервере `git.etersoft.ru`

## Организация учебной работы с сервером исходных кодов

Порядок работы с сервером, напрямую определяет возможности его использования в учебном процессе. В первую очередь, порядок работы, в рассматриваемом примере, включает в себя клонирование предварительно подготовленных примеров кода. Репозитории для примеров, подготавливаемых преподавателем, доступны в общем списке репозитория.

Первый этап знакомства с системой контроля версий `Git`, выполняется в виде публикации изменений на сервере исходных кодов, что требует самостоятельной генерации `ssh`-ключей и регистрации учётной личной записи на сервере. До последнего момента задача регистрации осуществлялась вручную администратором. С недавних пор началась отладка автоматической системы регистрации пользователей через сервер управления проектами — `project.toiit.sgu.ru`

В последующем обмен исходным кодом между студентами и преподавателем осуществляется через сервер исходных кодов, а доля самостоятельного участия в изменении примеров исходных кодов по заданиям возрастает. В первую очередь, за счёт сложности заданий и работы с ветками в систему контроля версий `Git`.

Поскольку доступ к репозиториям студентов открыт, обмен исходным кодом между собой осуществляется автоматически. Таким образом переход от единоличной работы над исходным кодом к совместной осуществляется в тот момент, когда это требуется для обмена наработками, а также в соответствии новыми заданиями.

Кроме результатов выполнения заданий сервер исходных кодов используется для публикации исходных кодов курсовых и дипломных работ.

На текущий момент завершается поэтапный ввод применения сервера исходных кодов в задачах курсов *Операционные системы и Технологии программирования*, которые читаются на 3 и 2 курсах очного и заочного отделений. Этапность предполагала переход от задач под `Linux` к задачам под `Windows`, замещающих использование `Visual Studio` средой разработки `QtCreator`, имеющей интеграцию с системой контроля версий `Git`.

## Литература

- [1] *ALT Linux Team*, *git.alt* — сервер совместной разработки ALT Linux Team, <http://www.altlinux.org/Git.alt>
- [2] *ALT Linux Team*, *girar* — система, обеспечивающая функционирование git-хостинга *git.alt*, <http://www.altlinux.org/Girar>
- [3] *Кафедра ДМИТ СГУ*, *git.toi* — учебный сервер исходных кодов, <http://git.toiit.sgu.ru/>
- [4] *Кафедра ДМИТ СГУ*, *project* — учебный сервер управления проектами, <http://project.toiit.sgu.ru/>
- [5] *Etersoft*, *git.eter* — сервер совместной разработки Etersoft, <http://www.etersoft.ru/>

Андрей Михеев

Москва, Консалтинговая группа РУНА / МЭСИ

Проект: RunaWFE <http://wf.runa.ru/rus>

## Использование концепции бинарных отношений в системе RunaWFE как пример интеграции математической теории в свободное ПО

### Аннотация

В докладе показываются преимущества СПО как возможного средства кооперации авторов идей и производителей ПО на примере создания инициализатора ролей в свободной системе управления бизнес-процессами и административными регламентами RunaWFE, основанного на математическом понятии «бинарное отношение».

В докладе предлагается использовать свободное ПО как средство кооперации представителей ВУЗовской науки и разработчиков промышленного ПО. В этом заинтересованы обе стороны:

Профессора, аспиранты и студенты, передав свои идеи и теории в проекты разработки свободного ПО, получают в конечном счёте инструмент, реализующий их идеи. При этом:

- Этот инструмент они получают бесплатно

- При помощи данного инструмента они смогут передавать (продавать) реализацию своих идей другим людям и этим людям не придется платить за ПО, а также тратить усилия на получение и установку ключей, лицензионных файлов, заключение договора с вендором и т. п. Например, преподаватели легко смогут передавать курсы практических занятий в другие ВУЗы.
- При использовании реализации идей и теорий ВУЗовских профессоров на реальных предприятиях в данном случае предприятия смогут избежать расходов на приобретение ПО
- Разработанное ПО можно будет свободно модифицировать при дальнейшем развитии идей и теорий

Разработчики промышленного ПО в свою очередь бесплатно получают идеи и теории, которые позволят разрабатываемому ПО получить качественные преимущества.

В докладе рассмотрен пример такого возможного взаимодействия: применение абстрактной математической концепции (изучаемой в большинстве физ.-мат или технических ВУЗ'ов на младших курсах) в свободной промышленной системе для решения практической задачи. Это привело к появлению конкурентного преимущества у программного продукта.

В качестве математической концепции рассматриваются бинарные отношения [1]. Программный продукт, в котором была применена концепция — свободная система управления бизнес-процессами и административными регламентами RunaWFE [2]. На восьмой конференции разработчиков свободных программ были рассмотрены технические аспекты реализации бинарных отношений в RunaWFE [3]. В настоящем докладе этот вопрос рассматривается с другой точки зрения — как возможный пример кооперации представителей науки и производителей свободного ПО.

## Описание абстрактной математической концепции

Рассмотрим математическое понятие «Бинарное отношение».

**Определение.** Бинарным отношением между множествами  $A$  и  $B$  называется любое подмножество  $P$  декартова произведения множества  $A$  на множество  $B$ . Часто, чтобы обозначить принадлежность упорядоченной пары  $(a, b)$  к бинарному отношению  $P$ , вместо запи-

си используют обозначения  $(a, b) \in P$  используют обозначения  $P(a, b)$  или  $aPb$ . При этом говорят, что  $a$  находится в отношении  $P$  к  $b$ .

**Замечание.** Для множеств  $A$  и  $B$ , состоящих из конечного числа элементов, любое отношение можно задать, определив набор упорядоченных пар  $a, b$  для этого отношения.

Некоторые бинарные отношения можно поставить в соответствие функциям. Функцию можно определить как такое бинарное отношение, в котором каждому значению  $b$  отношения  $aPb$  соответствует лишь одно единственное значение  $a$ .

Таким образом, бинарные отношения можно рассматривать как расширение понятия функция.

## Применение математической концепции в программном продукте для решения практической задачи

### Проблемы, связанные с разработкой инициализаторов ролей

Системы управления бизнес-процессами и административными регламентами раздают задания исполнителям и контролируют их исполнение.

Исполнителями заданий могут быть как сотрудники предприятия, так и информационные системы. Связывание узлов бизнес-процесса с исполнителями заданий производится при помощи ролей. При разработке бизнес-процесса создаётся роль и ставится в соответствие определённым узлам схемы бизнес-процесса. Инициализация роли — это назначение на роль конкретного исполнителя.

Реализация компонентов-инициализаторов ролей бизнес-процессов является одной из самых неудобных и трудоёмких проблем при внедрении систем управления бизнес-процессами и административными регламентами на предприятиях.

Традиционных подходов к реализации инициализатора роли два:

1. Внутри системы управления бизнес-процессами и административными регламентами задаётся организационная структура предприятия и роли инициализируются при помощи указания параметров этой структуры
2. Процедура инициализации роли выносится в какую-то другую информационную систему предприятия, — в бизнес-процесс помещается ссылка на удалённую функцию этой системы и настраивается механизм вызова удалённой функции, инициализирую-

щей роль конкретным исполнителем или группой возможных исполнителей

У обоих этих подходов есть существенные неудобства. Настроить удалённый вызов процедур из другой информационной системы обычно технически сложно. Как правило, эта настройка требует согласования с сотрудниками предприятия, отвечающими за безопасность данных, что требует много усилий. Задание иерархической организационной структуры предприятия внутри системы управления бизнес-процессами и административными регламентами часто тоже очень неудобно. Для работы с организационной структурой предприятия разрабатываются самостоятельные системы. Эти системы весьма сложны, их сложность может быть сравнима со сложностью самой системы управления бизнес-процессами. Крайне сложно сделать так, чтобы один компонент системы управления бизнес-процессами мог заменить такую систему. Кроме того, при помощи такой структуры легко инициализировать роли, соответствующие иерархии административного управления — «руководитель сотрудника», «руководитель отдела», «директор». Однако, в этом случае оказывается сложно инициализировать роли, непосредственно не относящиеся к административному управлению, такие как «сотрудник кадровой службы, ответственный за адаптацию недавно принятого на работу сотрудника», или «секретарь, отвечающий за корреспонденцию данного сотрудника».

В этих условиях оказалось, что использование понятия чистой математики — «бинарного отношения» позволяет разработать очень простое, но весьма эффективное решение конкретной программистской задачи построения инициализатора роли.

Такое решение сложно придумать программисту, не использующему абстрактных математических концепций в своей текущей деятельности. В данном случае полезной была бы кооперация программиста с математиком.

### **Описание решения.**

Предлагается кроме традиционных способов инициализации ролей добавить возможность инициализации ролей при помощи бинарных отношений.

Во-первых, это даст возможность инициализировать роль сразу множеством возможных исполнителей заданий. Часто в бизнес-

процессе задание направляется не одному исполнителю, а множеству возможных исполнителей задания. Выполняет это задание тот пользователь, который первым возьмёт его на исполнение.

Во-вторых, при использовании отношений процедура задания возможных исполнителей задания становится очень простой и её легко реализовать прямо в графическом интерфейсе.

Отношение над исполнителями заданий предлагается строить при помощи задания набора пар (Исполнитель1, Исполнитель2). При этом не требуется проверять каких-либо ограничений (как, например, для функции — что она возвращает только одно значение для одного исполнителя).

Задавать отношения перечислением всех определяющих его пар пользователей неудобно, так как таких пар может быть очень много. Для уменьшения количества вводимых данных имеет смысл воспользоваться группами пользователей.

То есть в паре (Исполнитель1, Исполнитель2), под Исполнитель является пользователем или группой пользователей.

Инициализация производится следующим образом:

1. Из указанной в инициализаторе роли переменной бизнес-процесса берётся её значение-Исполнитель. Это значение будет соответствовать правой части отношения.
2. Строится множество значений всех левых частей отношения, соответствующих данному элементу правой части. Для Исполнителя — значения правой части отношения выбираются все группы, в которые он входит. Находятся все пары определённые для данного отношения, у которых в правой части стоит Исполнитель или одна из выбранных групп.
3. Далее роль инициализируется множеством всех левых частей этих пар.

## **Применение.**

Решение было реализована в графическом интерфейсе системы следующим образом:

В главном меню системы появился ещё один пункт меню — «Отношения».

В этом пункте можно посмотреть/добавить/удалить отношение, открыть отношение и отредактировать множество составляющих его пар.

В редакторе в бизнес-процессе при редактировании инициализатора роли была реализована закладка «задать роль с помощью отношения». В этом случае можно задать настройки соединения с сервером и импортировать существующие в системе бинарные отношения в редактор.

Далее бинарное отношение можно поставить в соответствие роли. В форме выбирается имя отношения и переменная или константа, соответствующая правой части отношения, задающая пользователя или группу пользователей.

## Литература

- [1] А. Н. Колмогоров, С. В. Фомин, Элементы теории функций и функционального анализа. 4 изд. М. Наука, 1976
- [2] Ссылка на сайт проекта RunaWFE: <http://wf.runa.ru/>
- [3] Михеев А. Г. Реализация бинарных отношений в свободной системе управления бизнес-процессами и административными регламентами RunaWFE для упрощения инициализации ролей. — в кн.: Восьмая конференция разработчиков свободных программ. Тезисы докладов / Обнинск, 25 — 26 июля 2011 г. М.: Альт Линукс, 2011. (стр. 14 — 20)

Денис Пынькин, Иван Глецевич

Минск, Беларусь, Белорусский государственный университет информатики и радиоэлектроники  
<http://bsuir.by>

## Открытый подход к обучению студентов технической специальности ВУЗа

Аннотация  
Рассмотрены некоторые практические подходы организации обучения студентов технической специальности с использованием открытых методов и технологий. Описывается гетерогенная инфраструктура кафедры ЭВМ БГУИР на базе бездисковых рабочих станций, используемая для проведения практических занятий и научно-исследовательских экспериментов.

## Введение

Исторически сложилось, что при классической схеме обучения поддержка курса выглядит следующим образом:

- преподаватель выдает список рекомендованной литературы, а вся остальная информация по курсу «строго засекречена» и дается исключительно под запись на лекциях;
- практические занятия проводятся в строго контролируемой среде, с ограниченными правами для студентов;
- курсовые и самостоятельные работы, как и положено, выполняются каждым студентом самостоятельно, готовый результат предоставляется в конце семестра.

Таким образом, на каждом этапе появляются серьезные ограничения:

- студенты практически не могут изучить всю литературу по теме, а конспекты часто ведутся беспорядочно, причем сильно зависят не только от способностей лектора, но и от индивидуального восприятия;
- ограничение доступа к оборудованию делает невозможным проведение некоторых типов работ по системному и сетевому программированию;
- процесс выполнения самостоятельных работ тяжело отслеживается и приводит к закономерным результатам при защите — эпидемии «сгоревших» носителей «со всеми исходниками».

Открытый подход к обучению включает в себя отсутствие ограничений, как на информацию по изучаемому предмету, так и на техническую часть — начиная от ПК для лабораторных и практических работ и заканчивая сопутствующей инфраструктурой.

## Лекционный материал

В последние годы наметилась тенденция открывать курсы лекций и предоставлять доступ к материалам курса. К сожалению, в белорусских ВУЗах такое поведение не является нормой. Более того, лектор обязан готовить методические пособия, доступ к которым ВУЗ предоставляет в закрытом режиме — только для студентов ВУЗа (фактически платно).

В результате, если есть желание выкладывать материалы в открытый доступ, то приходится вести две версии методических материалов — открытую и «служебную».

Открытая же версия служит сразу нескольким целям:

- студенты получают доступ к актуальным версиям лекций;
- отсутствует «испорченный телефон» между преподавателем и студентами, которые по тем или иным причинам пропустили лекцию;
- лектор получает отзывы (и баг-фикс запросы) на свою программу, причем не только от студентов и коллег, но и от людей, работающих с соответствующими технологиями;
- у преподавателя есть стимул к постоянной модификации курса лекций.

## Открытая инфраструктура

Начиная с середины 2000-х на кафедре ЭВМ БГУИР проводится постепенный перевод инфраструктуры на максимальное использование бездисковых рабочих станций. На данный момент все учебные ПК и многие ПК сотрудников используют бездисковую загрузку нужной ОС[1].

Возможна загрузка как Linux[2], так и различных версий Windows. Так как ОС загружается в режиме «stateless», то у студентов есть полная свобода действий (хоть вирусы тестировать – жить им до щелчка рубильником). При использовании систем виртуализации существует возможность и одновременной загрузки различных ОС.

Для поддержки сетевой загрузки 60–70 станций используется 2 сервера — под управлением ОС Linux и Windows (для загрузки соответствующих ОС). Как показала практика, ключевыми элементами являются коммутаторы, которые должны обеспечить достаточно высокую пропускную способность (де-факто нужен Gigabit Ethernet).

Одним из самых больших плюсов такого подхода является уменьшение времени на обслуживание рабочих станций, ведь необходимо всего 2-3 квалифицированных инженера, которые могут произвести первоначальную настройку всей системы и затем реагировать на внештатные ситуации.

В качестве побочного эффекта имеется возможность использовать ресурсы учебных машин для научно-исследовательских эксперимен-

тов, что при классической организации затруднительно с технической точки зрения и практически неразрешимо с административной. Так, на базе лабораторий был создан учебный и исследовательский вычислительный кластер, на котором проводились эксперименты[3][4] для программ «Скиф» и «Скиф-ГРИД».

## Самостоятельная работа студентов

Курсовые работы — одна из самых важных и в то же время одна из самых «болезненных» частей обучения. Наверное, одна из самых больших проблем — это время, отведенное на консультации и контроль своевременности выполнения задания.

Использование стандартного инструментария современного разработчика одним махом решает кучу проблем — если соблюдать простые правила:

- использовать любую публичную систему контроля версий исходного кода;
- при отсутствии активности в репозитории на момент контрольной точки — минус балл от оценки.

Как результат:

- студенты изучают один из важнейших инструментов для совместной работы (печально, что изучение этих систем отсутствует в программе подготовки специалистов);
- появляется возможность выдать коллективное задание и, хотя бы приблизительно, оценить степень вклада каждого из участников;
- каждый студент на практике учится как создать свой открытый проект либо присоединиться к существующему;
- всегда существует минимум 3 копии исходников: у студента, на сервере и у преподавателя;
- ведется журнал активности студента;
- «автоматическое» прохождение контрольных точек;
- затрудняется возможность плагиата.

## Контроль знаний

Тесты — одна из наиболее спорных методик оценки результата подготовки студента. Тем не менее, она вполне приемлема, так как гарантирует минимальный уровень знаний. А в сочетании с классическими методами контроля знаний позволяет преподавателю больше времени уделить людям, которые стремятся к знаниям.

Важный этап — подготовка вопросов. И тут принципы и технологии, применяемые при разработке свободного ПО показывают себя во всей красе. Достаточно использовать открытые текстовые форматы данных и возможности, предоставляемые системами контроля версий. После этого составление вопросов можно доверить самим студентам, поощряя их за каждый принятый вопрос или найденную ошибку в основной базе.

## Вывод

Нетрудно заметить, что открытые принципы и технологии положительно сказываются на обучении студентов, стимулируют получение новых знаний и изучение технологий. Можно отметить, что при совместной работе преподавателя и студентов процесс обучения выходит на качественно новый уровень. Преподаватель уже не является непререкаемым авторитетом в силу своей должности, а раз за разом вынужден подтверждать свою квалификацию, фактически становясь при этом лидером открытого проекта.

## Литература

- [1] *Пынькин Д.А., Глецевич И.И.* Проектирование IT-инфраструктуры учебных заведений на базе ОС Linux <http://freeschool.altlinux.ru/wp-content/uploads/2010/02/Pereslav1-2010.pdf>
- [2] *Пынькин Д.А.* Бездисковые рабочие станции на базе технологий ALT Linux [http://lvee.org/media/presentations/lvee\\_2008\\_03-1.pdf](http://lvee.org/media/presentations/lvee_2008_03-1.pdf)
- [3] *Пынькин Д.А.* Изоляция GRID-задач с помощью lxc <http://lvee.org/en/articles/207>
- [4] *Пынькин Д.А.* Создание специализированного дистрибутива Linux для подключения к национальной GRID-сети <http://lvee.org/en/articles/242>

Владимир Симонов, Марина Храпченко

Москва, Московский гуманитарный педагогический институт, Институт системного программирования РАН

## **Применение свободного программного обеспечения для управления делопроизводством кафедры с учётом компетентностного подхода в соответствии с требованиями ФГОС 3 поколения**

### **Аннотация**

Рассматривается вопрос создания информационной системы кафедры с использованием свободного программного обеспечения.

С введением компетентностного подхода при реализации основных образовательных программ (в соответствии с ФГОС 3 поколения) актуален вопрос о синхронизации данных о количестве зачётных единиц/часов, отводимых на изучение какой-либо темы рабочей программы дисциплины, с уровнем сложности и важности данной темы, а также месте данной рабочей программы в учебном плане направления подготовки. При этом также учитывается разграничение на аудиторную и самостоятельную работу, форму итогового контроля и пр. Указанное является базой для формирования индивидуальной образовательной траектории. Хотя традиционная форма ведения записей в журналах, ведения ведомостей на бумажных носителях и т.п. неизбежно остаётся в силе, тем не менее, расчёты нагрузки профессорско-преподавательского состава (ППС), ведение электронных картотек ППС, сотрудников и студентов может с успехом осуществляться с применением современных систем управления базами данных (СУБД), локальных вычислительных сетей и т.п. Необходимо отметить, что большинство разработчиков CRM (1С, Галактика и пр.) не имеют готовых модулей для соответствующего учёта, поэтому в случае, когда организация не имеет опыта использования таких систем, представляется целесообразным на первом этапе отказаться от приобретения дорогостоящих решений и, если есть возможность, применить собственные разработки.

Требования к применяемым средствам разработки могут основываться на разных критериях, однако в данном случае выделим главные: надёжность функционирования, защищённость, понятность и простота использования, эффективность и простота установки. Кроме

того для ВУЗов очень важен вопрос стоимости программного обеспечения (ПО).

Первоначально встаёт вопрос о выборе операционной системы (ОС). В настоящее время всё большее число ВУЗов используют ОС Linux. Почти все Linux-системы являются свободным программным обеспечением (СПО), то есть дистрибутив можно скачать или просто скопировать его с другого диска. Вторым существенным преимуществом является низкие требования к ресурсам. И, наконец, существует большое количество дистрибутивов, из которых можно выбрать наиболее соответствующий поставленным целям. В качестве ОС использовался дистрибутив Fedora (<http://fedoraproject.org>) из группы дистрибутивов Red Hat. Fedora хорошо подходит как для установки на сервер, так и на рабочие станции. Стоит упомянуть и Webmin — программу для конфигурирования системы через Web-интерфейс. Установка Fedora очень проста и большей частью сводится к нажатию мышью кнопки «Далее». Если предполагается, что информационная система должна быть с возможностью удалённого доступа, то для ОС Linux в качестве web-сервера чаще всего используется Apache, который является свободным web-сервером.

Поскольку ядром информационной системы является база данных, то в качестве СУБД наиболее удачным решением является использование MySQL. Программное обеспечение MySQL — это ПО с открытым кодом. На веб-сайте MySQL <http://dev.mysql.com> представлена самая последняя информация о программном обеспечении MySQL. Код написан на C и C++. СУБД может быть установлена на все операционные системы с работающими потоками Posix и компилятором C++. Система основана на привилегиях и паролях, за счёт чего обеспечивается гибкость и безопасность с возможностью верификации с удалённого компьютера. Пароли защищены, поскольку они при передаче по сети при соединении с сервером шифруются. Для соединения клиенты могут использовать сокет TCP/IP, сокет Unix или именованные каналы (named pipes, под NT).

Наиболее популярным средством написания интерфейса для систем с возможностью удалённого доступа является язык PHP. PHP — это широко используемый язык программирования общего назначения с открытым исходным кодом. PHP относится к интерпретируемым языкам и работает на стороне сервера. С точки зрения грамматики и синтаксиса PHP является C-подобным языком. Более того, он аккумулирует наиболее востребованные средства, первоначально по-

явившиеся в других языках, например, поддержка регулярных выражений (Perl), объектно-ориентированные методы и поддержка работы с базами данных (C++), большое число стандартных функций, поддержка XML (Extensible Markup Language) и ODBC (Open Database Connectivity).

В настоящее время завершается разработка АСУ кафедры, содержащая практически всю информацию по учебному процессу (за исключением расписания занятий) с использованием СПО. Также с целью реализации компетентного подхода разработаны алгоритмы учёта количества зачётных единиц/часов и иных данных, учитываемых при реализации основной образовательной программы и рабочих программ по ФГОС 3 поколения.

## Литература

- [1] Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки 230400 «Информационные системы и технологии» (квалификация (степень) «бакалавр») (2010 г.).
- [2] Богословский В.А., Караваева Е.В., Ковтун Е.Н. и др. Методические рекомендации по проектированию оценочных средств для реализации многоуровневых образовательных программ ВПО при компетентном подходе. М.: Изд-во МГУ, 2007.
- [3] Соловьёв В.П., Крупин Ю.А., Золотарёва Н.М. Моделирование образовательного процесса и его результатов на основе компетентного подхода: Сб. статей Всероссийской научно-методич. конф. «Государственно-общественные объединения в системе профессионального образования». М.: МГТУ им. Н.Э.Баумана, 2008.
- [4] Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки 230400 «Информационные системы и технологии» (квалификация (степень) «бакалавр») (2010 г.).

А.Н.Пустыгин, Б.А.Тарелкин, А.А.Ковалевский,  
Е.А.Огуречникова, А.В. Десинов, Н.А. Ошнуров,  
Е.В.Старцев, М.В. Зубов  
Челябинск, Челябинский государственный университет

## Программы автоматизированного построения технических комментариев на основе промежуточного представления открытых исходных текстов программ

### Аннотация

В докладе излагаются примеры построения технических комментариев программ с открытым исходным текстом

В результате разработок набора строителей промежуточных представлений программ на языках высокого уровня [1] с открытым исходным текстом появилась экспериментальная база для разработки утилит извлечения знаний из исходного текста. Будем называть полученные данные техническим комментарием исходного текста.

1. Строитель технического комментария для исходных текстов Xblite. Xblite — клон языка высокого уровня Basic

```
FUNCTION Entry () hStdOut = XioGetStdOut ()
' standard console is 80 x 25
' turn off console curser
XioSetCursorType (hStdOut, $$NOCURSОР)
IF a & b THEN PRINT "OK"
END IF
IF (X=-ABS(-S)+ -2*(1-3)/7) THEN
PRINT "OK"
END IF
a = -ABS(-S5)+ -2*(1-3)
IFT ok= true & (x-5 <0) THEN
PRINT "OK"
END IF
IFT ok & (x <0)THEN PRINT "OK"
ELSE PRINT "ERROR"
END IF
```

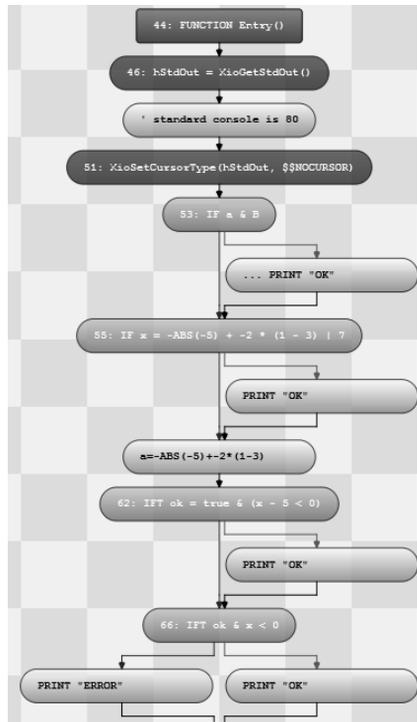


Рис. 1: Фрагмент технического комментария исходного текста на XVLite: БСА—подобное графическое представление алгоритма

Для получения промежуточного представления в виде текстового XML-документа использован препарат компилятора. Препарирование компилятора заключалось в доделке парсера путем дописывания операторов вывода, так что синтаксическое дерево разбора выводилось в текстовый файл.

2. Построитель технического комментария для исходных текстов Java

```

private void search(File topDirectory,
                    List res, int objectType){
    File[] list = topDirectory.listFiles();
  
```

```

for(int i = 0; i < list.length; i++){
    if(list[i].isDirectory()){
        if(objectType != FILES && accept(list[i].getName())){
            res.add(list[i]);
        }
    }
    search(list[i], res, objectType);
}
}
}
}

```

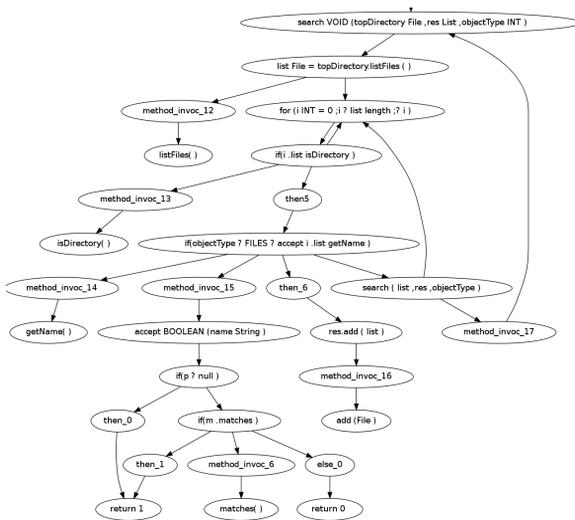


Рис. 2: Фрагмент технического комментария исходного текста на Java: БСА–подобное графическое представление алгоритма

### Литература

[1] А.Н.Пустыгин, Б.А.Тарелкин, А.А.Ковалевский, Е.А.Огуречникова, А.В. Десинов, Н.А. Ошнуров, Е.В.Старцев, М.В. Зубов Прототипы построителей промежуточных представлений исходных текстов программ, основанные на компиляторах с открытым исходным кодом — см. настоящий сборник стр. 82-86

А.Н.Пустыгин, Б.А.Тарелкин, А.А.Ковалевский,  
Е.А.Огуречникова, А.В. Десинов, Н.А. Ошнуров,  
Е.В.Старцев, М.В. Зубов  
Челябинск, Челябинский государственный университет

## Прототипы построителей промежуточных представлений исходных текстов программ, основанные на компиляторах с открытым исходным кодом

### Аннотация

В докладе описываются существующие способы промежуточного представления открытых исходных текстов программ и прототипы построителей, применяемые для их генерации.

Статический анализ исходного кода с использованием методов обработки текста оказывается недостаточно эффективным. На практике инструменты, выполняющие статический анализ, формируют промежуточные представления исходного кода. Самое распространённое решение — использование абстрактного дерева разбора AST [1]. Деревья можно получать различными способами, в том числе инструментами генерации парсеров, такими как ANTLR [2](Checkstyle[3] для Java), либо специфичными инструментами, такими как инфраструктура компилятора ROSE(Compass/ROSE)[4] для анализа языков C, C++ и Fortran. В проекте Pylint (для языка Python) для генерации AST используется отдельный модуль logilab-astng[5], который может быть использован независимо от Pylint.

Принципиально другой вариант — реляционное представление исходного кода. В результате предварительной обработки кода строятся реляционные отношения и заполняются значениями. В результате процесс анализа сводится к выполнению различных запросов к программному коду. Эта идея заимствована из логического программирования (Prolog, Datalog), в котором использовались запросы к базе знаний в коде. В современных инструментах используются SQL-подобные языки QUEL [6] (в Linton's OMEGA System [7] для Ada, C и Pascal) и .QL [8] (SemmlCode[9] для Java).

Промежуточные представления могут строиться как одноязыковые, так и мультязычные. Универсальные решения позволяют производить типовые процедуры анализа для различных языков, однако не все детали конкретного языка могут быть представлены.

Как правило, AST позволяет точно восстановить исходный код, однако для сложных языков (например, объектно-ориентированного) проще оказываются два уровня промежуточного представления. Так, в академическом проекте Bauhaus project [8] (языки Ada, C, C++, C#, и Java) имеются 2 представления — низкоуровневое InterMediate Language (IML) описывает конструкции языка на синтаксическом и семантическом уровнях, а представление более высокого уровня Resource flow graphs (RFG) описывает архитектурные особенности программной системы (файлы, компоненты и др.).

Зачастую представление продиктовано специальными требованиями. Так в свободном проекте BLAST [9] — утилите проверки моделей (верификации) для C анализируемая программа представляется в виде автомата потока управления (CFA), описывающего набор состояний-процедур и переходы между ними. Далее в соответствии с алгоритмом «абстракция и уточнение по контрпримерам» (counterexample-driven automatic abstraction refinement) создаются абстрактные деревья достижимости (ART), которые описывают возможные пути распространения управления программы с той долей абстракции, которая используется на очередной итерации алгоритма.

Как уже указывалось, большинство инструментов современных инструментов статического анализа основываются на представлении исходного кода в виде AST. Это можно отнести к недостаткам, т. к. по сути это формат внутренних данных компилятора и жёстко привязан к синтаксису языка программирования. Более эффективным представляется использование совокупности представлений, каждое из которых, возможно, и не несёт такой полной информации, с точки зрения соответствия исходному коду как AST, но упрощает типовые задачи статического анализа путём сокращения потребности машинного времени и памяти.

## Прототип построителя промежуточного представления C/C++

При разработке прототипа утилиты анализа открытых исходных текстов C/C++ рассматривались следующие варианты.

1. Открытая библиотека VivaCore [13] для работы с C и C++ кодом, основанная на проекте OpenC++, которая позволяет сохранять дерево разбора для дальнейшей обработки в произвольном формате пользователя, на любом этапе процесса обработки исходного кода внутри библиотеки VivaCore.
2. GCCXML [14] — свободный, с открытым исходным кодом парсер языка программирования C++, который является front-end'ом для компилятора GCC. Он выполняет XML-описание исходного текста языка C++ на основе внутреннего представления самого компилятора GCC. Достоинством данного парсера является реализация в полном объёме стандарта языка C++ в пределах используемой версии компилятора.
3. GCC Abstract Syntax Tree Analyzer (GASTA) [15] — свободный с открытым исходным кодом анализатор синтаксического дерева разбора компилятора GCC. GASTA получает на вход синтаксическое дерево разбора от компилятора GCC, вызванного с определёнными ключами компиляции.
4. Использование компилятора как внутреннего модуля. Необходимо построить оболочку для компилятора, которая будет вызывать внутренние функции компилятора и поддерживать необходимые интерфейсы. По завершении лексического и синтаксического анализа компилятор сформирует конечное синтаксическое дерево разбора, корень которого может быть доступен через глобальную переменную `global_namespace`.
5. Внешний модуль (plug-in) для компилятора GCC [16] (начиная версии GCC 4.5) путём подключения динамических библиотек к компилятору во время выполнения. Реализованный интерфейс позволяет подписаться на одно или несколько из определённых событий в процессе компиляции, которые определены в заголовочном файле `GCC-plugin.h`. Каждый плагин должен экспортировать функцию инициализации и определить глобальную переменную `plugin_is_GPL_compatible`, которая показывает, что

плагин был лицензирован под GPL-совместимой лицензией. Основной особенностью реализации plug-in является то, что компилятор GCC не позволяет одновременно получить полное синтаксическое дерево разбора, а имеется возможность только получить представление лишь на этапе декларирования и определения операндов языка. Для того чтобы разрешить поставленную задачу, необходимо совместить два представленных дерева синтаксического разбора на разных этапах формирования. Это можно осуществить за счёт одновременного подключения к двум событиям, определённым компилятором:

- a) `PLUGIN_PRE_GENERICIZE` — позволяет получить дерево синтаксического разбора на этапе определения операндов языка.
- b) `PLUGIN_OVERRIDE_GATE` — событие этапа объявления операндов языка.

## Прототип построителя промежуточного представления C#

Для реализации этого прототипа выбрана утилита `dmcs` проекта MCS (Mono C# Compiler) — компилятор языка C# по лицензии Creative Commons, GNU Public License, Lesser GNU Public License, Mozilla Public License, Microsoft Permissive License, MIT X11. Mono [17] — проект по созданию полноценного воплощения системы .NET Framework на базе свободного ПО.

При разработке прототипа утилиты рассматривались следующие варианты.

1. Использование интерфейсов CodeDom API . CodeDom — Code Document Object Model, предназначенный для разработки автоматических генераторов исходного кода.
2. Использование пакета C# 3.0 Parser Kit — набор для разбора файлов исходного кода языка C# от компании Temporal Wave.
3. Использование библиотеки ICSHarpCode.NRefactory , которая реализует функционал для лексического анализа языков C# и VB.NET и является проектом с открытым исходным кодом.
4. Использование функционала MCS путём помещения процедур лексического и синтаксического анализаторов исходного текста в «обёртку»

5. Использование технологии Отражения (Reflection), при котором не происходит изменения исходного кода компилятора, а все манипуляции с деревом разбора выполняются посредством программного интерфейса.

## Литература

- [1] *Ахо А., Лам М., Сети Р., Ульман Д.* Компиляторы: принципы, технологии и инструментарий, 2-е изд., ООО «И.Д. Вильямс», 2008, 1184 с.
- [2] ANTLR — <http://www.antlr.org>
- [3] Checkstyle — <http://checkstyle.sourceforge.net>
- [4] ROSE — <http://rosecompiler.org>
- [5] Pylint — <http://www.logilab.org/857>
- [6] logilab-astng — <http://www.logilab.org/856>
- [7] QUEL — Ingres 10.0 QUEL Reference guide, Ingres corp. 2010, <http://code.ingres.com/ingres/main/src/tools/techpub/pdf/QUELRef.pdf>
- [8] Linton's OMEGA System <http://www.eecs.berkeley.edu/Pubs/TechRpts/1983/5296.html>
- [9] QL — Oege de Moor, Elnar Hajiyev, Mathieu Verbaere. Object-oriented queries over software systems, ACM Press, 2007.
- [10] SemmleCode — <http://www.semmle.com//solutions/enabling-tools-for-your-projects/index.html>
- [11] Bauhaus project — <http://www.bauhaus-stuttgart.de/bauhaus/index-english.html>
- [12] BLAST — <http://mtc.epfl.ch/software-tools/blast/index-epfl.php>
- [13] Библиотека VivaCore для анализа языка C++ OpenC++: <http://www.viva64.com/ru/a/0013/>
- [14] Парсер языка C++ GCCXML: <http://gccxml.org/HTML/Index.html>
- [15] Анализатор синтаксического дерева разбора GASTA: <http://gasta.sourceforge.net/gastaDocs/gastaDocs.html>
- [16] Компилятор GCC: <http://gcc.gnu.org/onlinedocs/>
- [17] *Miguel de Icaza.* The Internals of the Mono C# Compiler. <https://github.com/mono/mono/blob/master/mcs/docs/compiler.txt>

Александр Малибашев

Новочеркасск, ГОУ ВПО Южно-Российский государственный технический университет (Новочеркасский политехнический институт)

## **Использование «Комплексной системы компьютерного обучения» в образовательном процессе технического университета**

### Аннотация

Рассматриваются основные компоненты комплексной системы компьютерного обучения (КСКО), представляющей собой дидактически обоснованную концепцию взаимодействия преподавателя со студентами в образовательном процессе технического университета, учитывающую требования Федеральных государственных образовательных стандартов. Система КСКО реализована на базе открытого программного обеспечения.

Современный период общественного развития характеризуется интенсивным становлением новых образовательных подходов в образовании. В соответствии с требованиями Федеральных государственных образовательных стандартов все изучаемые в ВУЗе дисциплины должны быть организованы в форме учебно-методических комплектов (УМКД).

Для создания УМКД и его эффективного использования при обучении студентов с различным уровнем начальной подготовки была разработана комплексная система компьютерного обучения (КСКО).

КСКО представляет собой автоматизированную, основанную на современных компьютерных телекоммуникационных и интернет-технологиях, систему управления обучением (СУО), которая является составной частью автоматизированной системой управления ВУЗа (АСУ ВУЗ). КСКО ориентирована на организацию высшего образования в технических университетах по государственным образовательным стандартам 2 и 3 поколения, а также учитывает возможность подключения (школьного) обучения как системы довузовской подготовки школьников как будущих абитуриентов ВУЗа, так и послевузовского (корпоративного) обучения.

КСКО является сложной интегрированной системой работающей в режиме клиент-сервер. СУО работает непосредственно на серверах

университета, Клиентская часть — автоматизированное рабочее место обучаемого (студента) может работать в трёх базовых режимах — сеть Интернет, корпоративная сеть ВУЗа и локальная версия. Наиболее функциональным является корпоративный режим, так как он имеет средства взаимодействия с АСУ ВУЗ и предоставляет обучаемым расширенный набор инструментов, делая процесс усвоения учебного курса более эффективным. Локальная версия применяется при самостоятельной работе студентов за пределами университета.

Поскольку основой образовательного процесса при очной форме обучения в ВУЗе остаются лекции, то в системе КСКО реализованы электронные слайд-лекции, позволяющие совместить текстовое и графическое сопровождение (фотоснимки, диаграммы, рисунки) с мультимедийными технологиями, анимацией и компьютерным моделированием изучаемых процессов, с возможностью показа документального видеоряда натуральных экспериментов.

Включённые в процесс чтения лекции фронтальные тесты превращают её в средство контроля образовательного процесса в аудитории с достаточно большим числом обучающихся и получения своевременной обратной связи об уровне усвоения учебного материала, что позволяет по реакции обучаемых регулировать темп изложения материала в рамках образовательного стандарта: упростить или усложнить изложение, сократить один раздел, усилить другой или сделать новый акцент в изложении материала.

Существует также возможность быстро изменять порядок представления учебного материала, а также постоянно модернизировать курс, вставляя новые разделы и главы или подключая внешние модули, такие как проверочные, исследовательские или моделирующие программные продукты.

Лабораторные работы организованы в виде компьютерных моделей, построенных с учётом реальных лабораторных установок. Виртуальные лабораторные работы могут также использоваться в качестве демонстраций.

Практические занятия проводятся в виде индивидуальных заданий на базе номеров зачётных книжек студентов. Модуль практических занятий выполняет обучающее-контролирующие функции.

Огромную помощь при выполнении самостоятельной работы студентам оказывает встроенная в систему электронная библиотека, включающая в себя полнотекстовую информацию, выдержки и главы из книг. Информационный поиск, осуществляемый студентами в

электронной библиотеке, учит студентов самостоятельно делать конспекты и выводы, анализировать информацию.

Дополнительной помощью при выполнении самостоятельной работы студентов является применяемая по каждой теме курса подсистема «Вопрос-ответ». Из базы вопросов и ответов может быть сформирован локальный вариант хранилища, что позволяет использовать эту подсистему при выполнении внеаудиторной самостоятельной работы студентов.

Для проверки уровня усвоения материала лектором используется созданная им система тестирования и тест-тренинга. Разработаны тесты и тест-тренинги по всем темам курса.

При реализации КСКО использовалось только открытое программное обеспечение входящее, как правило, в любые широко распространённые дистрибутивы Linux. Основу системы составляют — WEB сервер Apache, система управления базами данных MySQL, языки программирования Perl и PHP. При создании лабораторных работ использовались Java и Lazarus.

В качестве внутреннего обмена информацией между обучаемыми и преподавателем используются форматы представления данных и документов XML и PDF, что позволяет создавать и просматривать документы в различных операционных системах, делая систему частично кроссплатформенной.

Посредством КСКО сегодня преподаются около 50 курсов разного блока на 4 факультетах Южно-Российского государственного технического университета. Получен первый опыт использования системы при дистанционном обучении.

**Василий Олоничев, Борис Староверов**

Кострома, Костромской государственной технологической университет

## **Linux в подготовке студентов IT-специальностей**

### **Аннотация**

При подготовке будущих специалистов в области IT, основным является формирование специфичного компьютерного стиля мышления. Это достигается тем, что вместо Windows на всех видах занятий предлагается использовать один из клонов Unix, а именно, Linux. Самым подходящим дистрибутивом для российских условий является Alt Linux.

В настоящее время в большинстве российских университетов учебный процесс как для будущих специалистов в области информатики, так и для студентов всех остальных специальностей базируется на операционных системах семейства Windows с изучением специфических приёмов работы в данных ОС и с использованием прикладных программ, написанных в стиле и духе Windows. Обычно это объясняется тем, что ОС Windows доминирует в сегменте настольных систем, и поэтому в будущей практической деятельности молодые специалисты будут иметь дело именно с ней. Если для не ИТ-специальностей подобные утверждения могут являться предметом дискуссий, то для ИТ-специальностей, с нашей точки зрения, дело обстоит с точностью до наоборот. Дело в том, что ОС Windows изначально спроектирована для использования непрофессионалами и состоит из двух сильно различающихся частей: «интуитивно понятный» графический интерфейс, с одной стороны, и чрезмерно усложнённые и запутанные системные вызовы и конфигурационные файлы, с другой стороны. В результате как школьники, так и студенты, решившие получить специальность в области информационных технологий довольно быстро осваивают «хождение по визардам» и приходят к заключению, что именно в этом и заключается «искусство управления компьютером», тем более что на углублённое изучение системы времени как правило не хватает. В дальнейшем, обычно уже в условиях профессиональной деятельности, неизбежно происходит «встреча с суровой реальностью». Эту проблему можно в значительной мере смягчить, если подобную «ломку» организовать на первом курсе университета, т.е. дать понять студентам, что есть бытовой развлекательный прибор, который остался дома, и есть профессиональная вычислительная система, которая будет использоваться на всех занятиях в университете.

Для этих целей идеально подходят операционные системы семейства Unix. Эта ОС в своё время была создана для профессионалов, и работа с ней, в первую очередь в командной строке, прививает вкус к эффективной работе, формирует специфический «компьютерный» стиль мышления и тем самым превращает так называемый компьютер из «вещи в себе» в «вещь для нас». Тем более, что Unix имеет прекрасную систему документации. Общеизвестно, что тот, кто изучил Unix, легко переходит на Windows, в то время как «продвинутый пользователь Windows» с большим трудом осваивает не только Unix, но и системное программирование и администрирование в Windows.

На первый взгляд может показаться, что предлагаемое изучение Unix — это такой извращённый способ изучения Windows, с которой всё равно придётся иметь дело на практике. Однако на самом деле мнение о всеобщем доминировании Windows является мифом, основанном на том факте, что для большинства «пользователей» компьютер и настольная система — это синонимы. В то время как суперкомпьютеры, серверы и встраиваемые системы в большинстве случаев работают под управлением Unix и выполняют не в пример больше полезной работы по сравнению с настольными системами.

Для установки в учебные классы из имеющихся Unix-подобных операционных систем наиболее подходящей на сегодняшний день является Linux. Это объясняется тем, что Linux является свободной системой, что избавляет системных администраторов от хлопот по регистрации и перерегистрации ОС после переустановки, которая очень часто требуется в условиях учебного заведения, да и позволяет экономить немалые средства. Кстати, именно Windows приходится регулярно переустанавливать, а в Linux без прав администратора чрезмерно любопытный студент может навредить только самому себе. Linux быстро развивается и имеет большую команду разработчиков, поэтому он поддерживает очень большой спектр оборудования, что очень актуально в условиях учебного заведения, где как правило имеется большой парк самых разномастных компьютеров. И, наконец, Linux, будучи установленным и настроенным на одном компьютере без проблем переносится на другие компьютеры, и при этом не возникает никаких проблем с логическими дисками, чипсетами и т.д.

Из имеющегося разнообразия Linux для российских условий желательно выбрать дистрибутив, русифицированный «из коробки» и имеющий достаточно большой репозиторий. Этим требованиям в наибольшей мере удовлетворяет Alt Linux.

Сидун Наталья Николаевна, Чичкарев Евгений Анатольевич  
Мариуполь, Донецкая область, Украина, Приазовский государственный  
технический университет

## Использование вычислительного кластера под управлением ОС Linux в высшем учебном заведении

### Аннотация

Работа посвящена использованию открытого программного обеспечения в учебном процессе технического университета для организации курса параллельного программирования. Рассмотрены особенности построения вычислительного кластера на базе учебной лаборатории, укомплектованной персональными компьютерами.

Необходимость изучения в высшей школе теории, методов и технологий параллельного программирования становится всё более очевидной. Многие задачи прикладной и фундаментальной науки требуют использования этих технологий для решения.

Круг задач, решаемых с помощью параллельных программ, достаточно специфичен. Обязательным условием их рационального использования является их дискретность, распараллеливание на независимые подзадачи. К таким задачам относятся автоматизация проектирования, управление технологическими процессами, но наиболее широкий круг задач — вычислительные задачи, решаемые приближённо численными методами: математическое и имитационное моделирование процессов и систем (метеорологическое моделирование и прогнозирование, гидро- и газодинамика, биохимия, биофизика, теплофизика, экология), анализ экономических систем (валютные и фондовые рынки, макроэкономические показатели).

Изучение параллельных вычислительных алгоритмов в целом невозможно без использования многопроцессорных комплексов. Поскольку затраты на покупку и поддержку таких систем велики, сократить их может проектирование и построение вычислительного кластера на основе нескольких персональных компьютеров. Рациональным вариантом решения этой проблемы является построение вычислительного кластера на базе компьютеров в учебной лаборатории.

Необходимым и достаточным условием построения кластера является наличие группы относительно однородных по аппаратной кон-

фигурации компьютеров, соединённых высокоскоростной сетью. Компьютеры становятся узлами кластера, при чём каждый из них работает под управлением своей операционной системы, один из них следует выделить под управляющий узел.

В первую очередь при проектировании вычислительного кластера, следует учесть требования к аппаратному обеспечению каждого узла. При прочих равных следует отдать предпочтение компьютерам на базе процессоров Intel (это связано с наличием высокоэффективных компиляторов параллельных программ, например Intel® Fortran Compiler) с наибольшей доступной тактовой частотой, а также обратить внимание на стабильность устройства (компьютера) в целом. Менее значимыми параметрами является объём памяти и хранения данных, однако, в целом для системы этот объём должен быть достаточно большим. Управляющий узел кластера должен быть более производительным, чем рядовые, так как на него ляжет основная нагрузка по распределению заданий и сбору и выводу результатов работы.

Идеальным вариантом ОС для кластера являются UNIX-образные ОС (Linux, FreeBSD), процесс администрирования таких систем прозрачен для программиста и гибок. [1][2]

Важным условием является возможность установки ОС параллельно Windows на компьютерах в лаборатории, поскольку учебный процесс в целом ориентирован на её использование.

Основой кластера является не столько операционная система, сколько коммуникационная среда, обеспечивающая возможность частям параллельной программы, выполняющимся на разных компьютерах, эффективно взаимодействовать между собой. При выборе сетевого соединения для кластера необходимо учитывать его скорость и устойчивость, идеальным вариантом может быть Myrinet или 10Gbit Ethernet, однако, это дорогие решения, так что для создания учебного вычислительного кластера можно ограничиться FastEthernet или GigabitEthernet. При этом оптимальное число вычислительных узлов будет зависеть от скорости сетевого соединения, для каждого типа существует порог, при превышении которого сеть становится узким местом системы и дальнейшее наращивание числа узлов не даёт прироста производительности. [2]

Кластер в ПГТУ собран из 7 системных блоков идентичной конфигурации (DualCore Intel Pentium, 1800 MHz cache 1 Mb; RAM: 1Gb DDR2-667; 1 HDD 250Gb SATA 2) и одного управляющего блока (2

CPU Intel Core 2 Duo P8400, 2266 MHz cache 3 Mb; RAM: 4Gb DDR2-667; 1 HDD 320Gb SATA 2), связанных GigabitEthernet, установленная ОС — Ubuntu 11.04 i386. Теоретическая вычислительная мощность кластера — 851 GFlops, а лучший результат при использовании тестов Linpack — 634 GFlops [3].

В ходе изучения курса параллельного программирования студенты обучаются постановке задач, а также созданию программ с использованием технологий OpenMP и MPI и языков программирования C++ и Фортран, получая практические навыки разработки, оптимизации и тестирования параллельных программ, а также решения прикладных математических задач с использованием математических пакетов, например, Octave, поддерживающих организацию параллельных вычислений.

Таким образом, построение кластера на базе учебной лаборатории помогает организовать изучение технологий параллельного программирования и избежать существенных затрат на приобретение и обслуживание многопроцессорных систем.

## Литература

- [1] *А.В. Старченко* Высокопроизводительные вычисления на кластерах, Издательство Томского университета, 2008
- [2] *Сбитнев Ю.И.* Кластеры, Екатеринбург, 2009
- [3] *Чичкарев Е.А.* Оценка эффективности решения вычислительных задач на кластерных системах по, Приазовский государственный технический университет, 2011

Филипп Занько

Казань, Исследовательский центр проблем энергетики КазНЦ РАН

## Свободные инструменты на языке Python в исследованиях турбулентности

### Аннотация

Свободные научные программы — достойная современная альтернатива коммерческим продуктам. На примере реальной научной проблемы, относящейся к исследованию турбулентности, демонстрируются возможности некоторых известных научных пакетов, написанных на языке Python, анализируются их сильные и слабые стороны.

Чтобы работать с данными, полученными в результате экспериментального или численного моделирования турбулентных потоков, требуется соответствующее программное обеспечение. Как минимум, необходимо иметь возможность строить научные графики и проводить математическую обработку больших массивов чисел. Как правило, для такого рода задач используют профессиональные коммерческие пакеты, например Origin, MATLAB или Tecplot. Не подвергая сомнению высокий уровень подобных программ, их удобство и большие возможности, укажем лишь, что их законное использование — очень дорогое удовольствие. Только одной этой причины достаточно, чтобы начать поиск альтернативы из мира свободного программного обеспечения.

Таких достойных альтернатив, на самом деле, не мало. Их обзор, а также анализ причин, по которым выбраны именно свободные научные продукты на языке Python, не входит в цели этой статьи. Вместо этого сформулируем требования, которые предъявляет одна из задач по исследованию турбулентности к языку программирования и вспомогательным научным пакетам, а также проанализируем полученный опыт их применения.

*Задача:* написать программу, рассчитывающую и строящую графики основных интегральных статистических характеристик для любого несжимаемого пограничного слоя на плоской пластине.

Требования к программе (см. Рис. 1):

- входные данные отличаются большим разнообразием, поэтому бессмысленно делать программу с графическим интерфейсом

пользователя; вместо этого лучше всё время работать непосредственно с *кодом*, написанном в *стиле языка MATLAB*, который всегда можно быстро модифицировать, если нужно обработать данные нового вида;

- перед обработкой числовые *данные* требуется *записать в файл формата hdf5* универсальной структуры;
- программа обработки должна *считывать данные*, приведённые к единому виду, *из файла формата hdf5*;
- для математической обработки числовых данных нужны *библиотечные функции* линейной интерполяции, численного интегрирования и пр.;
- для представления результатов необходим *пакет научной графики* публикационного качества с поддержкой ввода формул, русского языка и др.

Для решения задачи были задействованы следующие свободные научные пакеты на языке Python:

- NumPy — пакет, поддерживающий большие многомерные массивы и матрицы, а также соответствующие математические операции над ними
- Matplotlib — библиотека научной графики с поддержкой формул Latex
- PyTables — пакет для работы с очень большими объёмами числовых данных, поддерживает ввод/вывод в файлы формата hdf5.

От работы с указанными инструментами сложились следующие впечатления:

- Все пакеты *устанавливаются без осложнений*, также их можно *быстро начать использовать*.
- Следует отметить *сравнительную простоту, выразительность и гибкость языка программирования*. Код легко пишется и легко читается.
- Показательно, что по научным инструментам на Python уже появилась серьёзная образовательная литература, по крайней мере на английском языке [1–4].
- Все инструменты достаточно хорошо отлажены; с ошибками, с непредсказуемым поведением сталкиваться не приходилось.

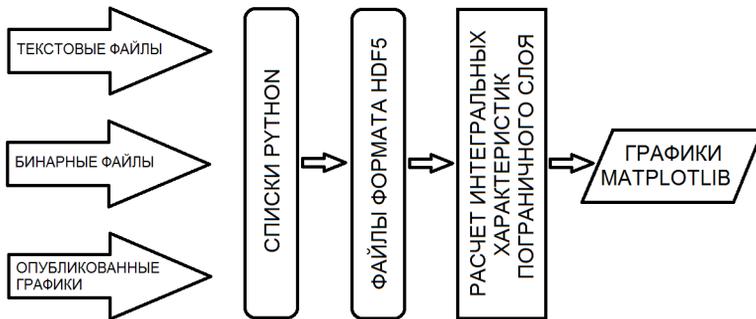


Рис. 1: Схема преобразования данных

- Качество графики не уступает коммерческим пакетам, поддерживаются сложные формулы и русский язык (см. Рис. 2).
- Все инструменты — кроссплатформенные, как и сам Python. Проект делался на Linux OpenSuse 11.3.
- При обработке большого количества численных данных, скажем 500 массивов размерностью  $2 \times 193$  из чисел формата float64, суммарное время ожидания результата выросло до 3 минут (процессор AMD Phenom II X4 945, оперативная память 4Гб).

*Выводы.* Свободный инструментарий на языке Python способен оказать большую помощь в решении небольших проблем, часто возникающих в рутинной научной работе. Заменить калькулятор, написать небольшое GUI-приложение с помощью Tkinter, считать данные нестандартного формата, построить график для научной статьи, выполнить не очень громоздкие расчёты и многое другое можно сделать быстро и с удобством. Особенно важно, что Python и его расширения не предъявляют слишком серьёзных требований к квалификации учёных как программистов. Крупные же проекты, связанные с большим объёмом вычислений, вероятно, лучше выполнять с применением других языков программирования.

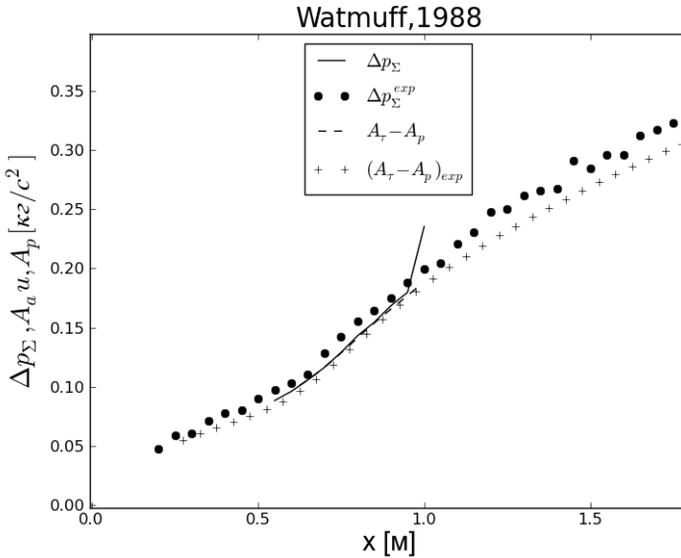


Рис. 2: Пример научного графика, полученного с помощью пакета Matplotlib

## Литература

- [1] *Kiusalaas J.* Numerical Methods in Engineering with Python. — New York: Cambridge University Press, 2005. — 424 p.
- [2] *Langtangen H.P.* Python Scripting for Computational Science. — Springer, 2008. — 750 p.
- [3] *Langtangen H.P.* A Primer on Scientific Programming with Python. — Springer, 2009. — 693 p.
- [4] *Tosi S.* Matplotlib for Python Developers. — Birmingham-Mumbai: Packt Publishing, 2009. — 293 p.

Александр Гороховский

Киев, Украина, Национальный технический университет Украины «КПИ»

Проект: Chemical cocktail <http://78.27.139.57/chemistry/>

## Идентификация, разбор и конструирование уравнений химических реакций с помощью Chemistry::Harmonia

### Аннотация

Рассмотрены концептуальные алгоритмы и интерфейсы подпрограмм Perl-модуля Chemistry::Harmonia — `class_cir_brutto`, `parse_chem_mix`, `prepare_mix`. С их помощью можно, соответственно: рассчитать *уникальный общий идентификатор реакции*, состоящий из буквенного CLASS и цифрового CIR (Chemical Integer Reaction identifier) компонента, с брутто-формулами веществ, химически правильно интерпретировать и разобрать химическую смесь (обычно уравнение реакции) на два множества — реагентов (исходных веществ) и продуктов, а также обратно сконструировать уравнение реакции.

Их применение разнообразно: CLI и Internet приложения, создание идентификационных ключей для уравнений реакций в химических базах данных и контроль целостности этих записей.

Разработка выполнена благодаря дистрибутивам ALT Linux последних лет, языку с раздвоением личности Perl и его модулям из Единой Архивной Сети — CPAN.

Общеизвестно [1], что химическое уравнение представляет из себя условную запись химической реакции формулами веществ с числовыми (стехиометрическими) коэффициентами, которая опирается на закон сохранения массы (стехиометрии). Благодаря такой записи, в достаточно лаконичной форме, передаются качественные и количественные представления о реакции, природе реагентов и продуктов.

В практике любых количественных химических расчётов (кинетических, термодинамических) или информационного поиска физико-химических, токсикологических свойств обычно возникает задача разделения записи химической смеси или уравнения реакции на два множества: реагентов и продуктов. С этой задачей в полной мере справляется рассматриваемая ниже подпрограмма `parse_chem_mix`, входящая в Perl-модуль Chemistry::Harmonia.

Интерфейс `parse_chem_mix` в качестве аргумента ожидает строку, в которой, по устоявшейся традиции, разделителями реагентов и

продуктов могут быть последовательности символов: *равно* (=) и/или *минус* (-), идущие совместно с одним или несколькими символами (или без него) *больше* (>). Например:

=, ==, =>, ==>, ==>>, -, --, ->, -->, ->>> и т. д.

Пробельные символы вокруг разделителя не имеют никакого значения. В случае, когда разделитель отсутствует (химическая смесь), формула последнего вещества рассматривается в качестве продукта.

Безусловно, кроме разделителя реагентов и продуктов, `parse_chem_mix` ожидает в анализируемой строке найти стехиометрические коэффициенты и химические формулы веществ, которые будут разделены одним из символов: *плюс* (+), *запятая* (,), *точка с запятой* (;) и/или *пробельными* символами. Для стехиометрических коэффициентов дополнительно формируется хеш их значений.

Интерфейс `parse_chem_mix`

```

1 use Chemistry::Harmonia qw( parse_chem_mix );
2 use Data::Dumper;
3 my %coef;
4 my $chem_eq = 'BaS + 2 H2O = Ba(OH)2 + 1 Ba(SH)2';
5 my $out_ce = parse_chem_mix( $chem_eq, \%coef );
6 print Dumper( $out_ce, \%coef );

```

OUTPUT

```

$VAR1 = [ [ 'BaS', 'H2O' ], [ 'Ba(OH)2', 'Ba(SH)2' ] ];
$VAR2 = {
    'Ba(SH)2' => '1',
    'H2O' => '2'
};

```

Возможности `parse_chem_mix` не ограничиваются только простым разделением на множества реагентов, продуктов и стехиометрических коэффициентов. Она выполняет функции распознавания и исправления различных ситуаций, с явно некорректно заданными структурами в строке аргумента. Так, нули (0) «похожие» на кислород (O) заменяются на последний. Существует возможность и более строгого (принудительного) управления такими ситуациями.

Следующая подпрограмма, `prepare_mix`, является антиподом предыдущей — `parse_chem_mix`. С её помощью можно наоборот, сконструировать уравнение реакции из двух множеств (массивов): реаген-

тов и продуктов, а также, используя хеш, дополнить их соответствующими стехиометрическими коэффициентами. Ещё одной её особенностью является контроль реагентов и продуктов в уравнении, с помощью корректирующего массива. Поясняющий интерфейс `prepare_mix` представлен ниже.

Интерфейс `prepare_mix`

```

1 use Chemistry::Harmonia qw( prepare_mix );
2 my $ce = [ [ 'O2', 'K' ], [ 'K2O', 'Na2O2', 'K2O2', 'KO2' ] ];
3 my $coef = { 'K' => 2, 'K2O2' => 1, 'KO2' => 0 };
4 my $real = [ 'K', 'O2', 'K2O2', 'KO2' ];
5 print prepare_mix( $ce, { 'coefficients' => $coef,
6                          'substances' => $real } );

```

## OUTPUT

```

O2 + 2 K == 1 K2O2 + 0 KO2

```

Одной из проблем хемоинформатики, ведения химических баз данных (БД) является выбор такого уникального идентификатора химических реакции, формирование которого происходило бы автоматически и при этом ему были бы присущи следующие свойства:

1. *не чувствительность* к форме записи химических формул участников превращений, порядку их следования в уравнении и указания направления реакции;
2. функция простого *классификатора* реакций по участникам превращений, через состав химических элементов (атомов) их образующих;
3. *уникальность* в пределах класса реакций и БД в целом, относительная *компактность* и *простота* реализации;
4. тривиальная функция *контроля целостности* и *корректности* хранящейся и поступающей информации.

Перечисленные выше минимальные требования к идентификатору не случайны.

Так, свойство *не чувствительности* продиктовано тем, что одна и та же химическая реакция может быть записана по разному внешне выглядящими уравнениями, поскольку одно и тоже вещество может быть представлено несколькими формулами с различающимся порядком следования атомов и/или их групп. Кроме этого, на практике, не накладывается вообще никаких ограничений и на порядок следования

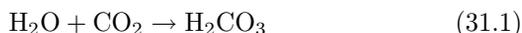
участников превращений в уравнениях, а направление реакции<sup>1</sup> зависит от свойств веществ, внешних условий, но в данном случае (закона стехиометрии) тоже не должно влиять на идентификатор. Конечно, свойство *не чувствительности* поможет в поиске идентичных реакций, а также в контроле их дубликатов.

Функция *классификатора* реакций должна способствовать оптимизации поиска, а также обеспечивать возможность находить уравнения реакций и в условиях не полностью заданной информации, например, при отсутствии некоторых реагентов или продуктов.

Следствием перечисленных выше и естественных требований *уникальности, простоты, компактности, контроля целостности* является идея составной и внутренне-пересекающейся формы идентификатора, а именно, *буквенно-цифровой*. Эта идея была реализована в виде подпрограммы `class_cir_brutto` Perl-модуля `Chemistry::Harmonia`.

Концептуально алгоритм расчёта *уникального общего идентификатора реакции* включает следующие действия:

1. Вначале химическая смесь (уравнение реакции) разбирается на два множества формул участников превращений — реагентов и продуктов<sup>2</sup>. Кроме этого формируется хеш их стехиометрических коэффициентов, а также, благодаря подпрограмме `parse_formula` модуля `CPAN Chemistry::File::Formula`<sup>3</sup>, готовится массив химических элементов. Например, для реакции:



будут сформированы множества участников превращений:

$$\{\text{H}_2\text{O}, \text{CO}_2\}, \quad \{\text{H}_2\text{CO}_3\} \quad (31.2)$$

хеш стехиометрических коэффициентов:

$$(\text{H}_2\text{O} \Rightarrow 1, \text{CO}_2 \Rightarrow 1, \text{H}_2\text{CO}_3 \Rightarrow 1) \quad (31.3)$$

и массив химических элементов:

$$[\text{H}, \text{O}, \text{C}] \quad (31.4)$$

<sup>1</sup> В том смысле, что считать реагентами, а что продуктами.

<sup>2</sup> Эту работу выполняет выше рассмотренная `parse_chem_mix`.

<sup>3</sup> Автор модуля *Ivan Tubert-Brohman*.

2. Далее из массива химических элементов (31.4), через объединение атомов в алфавитном порядке, формируется первая компонента идентификатора — класс реакции CLASS. Для примера выше (31.1) CLASS = CHO
3. Множества химических формул участников превращений (31.2) преобразуются в брутто-формулы, подобно системе Хилла [2], однако статус всех атомов, включая углерод (C) и водород (H) обычный, т. е. все атомы вещества упорядочиваются в алфавитном порядке с указанием их количеств.

Для примера (31.2) будут получены множества:

$$\{H2O1, C1O2\}, \quad \{H2C1O3\} \quad (31.5)$$

а в случае сложной формулы  $Va(HCO_3)_2 \rightarrow Va1C2H2O6$ .

4. Все брутто-формулы участников превращений (31.5) с соответствующими стехиометрическими коэффициентами (31.3) объединяются в одно *нормированное* множество и сортируются по возрастанию так, чтобы вначале находились формулы с наименьшими: стехиометрическими коэффициентами (перед каждой брутто-формулой), обозначениями атомов (в алфавитном порядке) и количеством атомов каждого химического элемента. Например, для (31.5) из двух множеств будет сформировано одно *нормированное* множество:

$$\{1C1H2O3, 1C1O2, 1H2O1\} \quad (31.6)$$

5. Вторая компонента общего идентификатора — химический целочисленный идентификатор реакции (CIR) — формируется через вычисление 32 битной циклической контрольной суммы (CRC-32)<sup>4</sup> [3, 4] нормированного множества (31.6), при этом его элементы объединяются в брутто-строку знаком плюс (+).

Завершая пример (31.6), получаем CIR<sup>5</sup>:

$$1C1H2O3 + 1C1O2 + 1H2O1 \quad \rightarrow \quad CIR = 1334303561$$

Для расчёта CIR можно воспользоваться и GNU-версией программы `cksum`, например, выполнив из shell<sup>6</sup>:

<sup>4</sup> Значения CRC-32 лежат в диапазоне  $0 \dots 2^{32} - 1$ , т. е.  $0 \dots 4294967295$ .

<sup>5</sup> CRC-32 брутто-строки находят с помощью модуля `CPAN String::CRC::Cksum`, автор *Andrew Clarke*.

<sup>6</sup> Здесь только первое число является CIR. Второе — длина строки.

```
$ echo -n 1C1H2O3+1C1O2+1H2O1 | cksum
1334303561 19
```

Таким образом, *уникальным общим идентификатором реакции* (31.1) будет: СНО-1334303561.

Рассмотренный идентификатор уравнений реакций ограничен более чем 4 млрд. реакций в каждом буквенном CLASS. Много это или мало?

Чтобы ответить на этот вопрос, обратимся к самому крупному держателю химической информации — Химической Реферативной Службе (Chemical Abstracts Service, CAS)<sup>7</sup>. На сегодня, CAS зарегистрировано более 38,1 млн. неорганических и органических реакций.

Для сравнения, ещё один крупнейший источник химической информации — фактографические базы данных ВИНТИ<sup>8</sup>, которые содержат более чем 3 млн. химических реакций, с ежегодным пополнением  $\approx 100000$ .

К сожалению, доступ к БД CAS и ВИНТИ закрыт и возможен только на платной основе. Поэтому возможность `class_cir_brutto` формировать идентификатор неорганических реакций была проверена автором более чем на 23500 уникальных уравнениях, собранных из отечественных и доступных зарубежных печатных изданий по химии<sup>9</sup>. Интерфейс этой подпрограммы представлен ниже.

Интерфейс `class_cir_brutto`

```
1 use Chemistry::Harmonia qw( class_cir_brutto );
2 use Data::Dumper;
3 my %coef;
4 my $reaction = '1 H2O + 1 CO2 --> 1 H2CO3';
5 my $ce = parse_chem_mix( $reaction, \%coef );
6 print Dumper class_cir_brutto( $ce, \%coef );
                                End of code
```

OUTPUT

```
$VAR1 = [ 'СНО',
          1334303561,
```

<sup>7</sup> Подразделение Американского Химического Общества (American Chemical Society) [www.cas.org/cgi-bin/cas/regreport.pl](http://www.cas.org/cgi-bin/cas/regreport.pl).

<sup>8</sup> Всероссийский Институт Научной и Технической Информации [www2.viniti.ru/](http://www2.viniti.ru/)

<sup>9</sup> Автор будет признателен всем желающим поделиться имеющейся информацией по уравнениям неорганических реакций ([angel@domashka.kiev.ua](mailto:angel@domashka.kiev.ua)).

```
{ 'H2CO3' => 'C1H2O3',  
  'CO2' => 'C1O2',  
  'H2O' => 'H2O1' }  
];
```

## Литература

- [1] *Левицкий, М.* Язык химиков / М. Левицкий // *Химия и жизнь*. — 2000. — Т. 1. — С. 50–52.
- [2] *Hill, E. A.* On a system of indexing chemical literature; Adopted by the Classification Division of the U.S. Patent Office / E. A. Hill // *J. Amer. Chem. Soc.* — 1900. — Vol. 22, no. 8. — Pp. 478–494.
- [3] *Уоррен, Г. С.* Алгоритмические трюки для программистов, испр. изд. : Пер. с англ. / Г. С. Уоррен. — М.: Вильямс, 2004. — 288 с.
- [4] *Williams, R. N.* Элементарное руководство по CRC алгоритмам обнаружения ошибок / R. N. Williams. — 1993. — С. 36. <http://rsdn.ru/article/files/classes/SelfCheck/crcguide.pdf>.

Михаил Пожидаев, Денис Жарков, Валентина Кирюшкина  
Томск, Томский государственный университет  
<http://inf.tsu.ru>

## Исследование методов извлечения данных при помощи СПО и материалов энциклопедии «Википедия»

### Аннотация

Доклад содержит описание исследования методов получения информации из материалов энциклопедии «Википедия» при помощи свободных программных пакетов с реализацией алгоритмов Data Mining и с утилитами, необходимыми для обработки естественного языка.

В крупнейших репозиториях свободного программного обеспечения (СПО) наблюдается недостаток развитых инструментов, традиционно называемых «интеллектуальными системами» (ИС). К ним относится программное обеспечение (ПО), способное к анализу и обработке информации в форме, сравнимой с восприятием информации

человеком. Научные сотрудники, желающие вести исследования в области ИС, сталкиваются с рядом трудностей, связанных с большим количеством сопутствующих прикладных задач и недостатком материалов, пригодных для использования в качестве входных данных для исследуемых алгоритмов.

С конца 90-х годов прошлого века получили развитие такие научные направления, как Data Mining и Text Mining. Они занимаются исследованием различных подходов к выделению значимой для человека информации из больших массивов слабоструктурированных данных. Обычно алгоритмы Data Mining носят эвристический характер и основаны на вычислении различных статистических параметров, но дополнительно предпринимаются попытки разработки методов, основанных на строгих и формализованных правилах логического вывода, с исследованием моделей хранения знаний, как, например, семантические сети.

Ряд зарубежных университетов ведёт разработку инструментов, способных значительно упростить исследования в области DATA Mining, и распространяемых на условиях свободных лицензий. Отметим среди них проект CoreNLP Стэнфордского университета<sup>1</sup> и проект WEKA университета Уайкато (Новая Зеландия)<sup>2</sup>.

Пакет CoreNLP предоставляет реализацию инструментов для решения некоторых задач по обработке естественного языка (Natural Language Processing), среди которых получение нормальной формы слов, обработка членов предложения и пр. Решение подобных задач часто необходимо при исследовании алгоритмов Text Mining.

Проект WEKA предоставляет реализацию распространённых типовых подходов к Data Mining, включая извлечение правил ассоциаций и выборку атрибутов на основе заранее подготовленной базы данных (БД), которая должна представлять из себя множество объектов с заданными значениями атрибутов. Применение пакета WEKA требует наличия достаточно обширной БД для получения значимых полезных результатов. Оба указанных проекта реализованы на языке Java.

В Томском государственном университете (ТГУ) ведутся исследования возможности подготовки БД для запуска алгоритмов WEKA на основе обработки статей свободной энциклопедии «Википедия».

---

<sup>1</sup>См. <http://nlp.stanford.edu/software/corenlp.shtml>

<sup>2</sup>См. <http://www.cs.waikato.ac.nz/ml/weka/>

Изучаются подходы к выделению множества объектов и их атрибутов на основе текста статей. При таком рассмотрении алгоритмы Text Mining выступают как предварительный этап для алгоритмов Data Mining.

В ходе исследований планируется постановка множества вычислительных экспериментов. В них главные члены предложений выступают как обозначения объектов БД в двух случаях: если они являются ссылками на некоторые статьи или если они не являются ссылками, но их нормальная форма совпадает с названием статьи, где они упоминаются. Набор второстепенных членов предложений рассматривается как потенциальные значения атрибутов объектов. При этом ряд сопутствующих задач по обработке естественного языка решается при помощи инструментов CoreNLP. Множество статей Википедии доступны для загрузки в виде единого файла в формате XML. Большой объём исходных материалов значительно усложняет задачу, и в случае серьёзных трудностей рассматривается возможность привлечения мощностей вычислительного кластера ТГУ «СКИФ Siberia».

Получение значимых результатов позволит подготовить новые свободные инструменты для решения сложных задач. К ним относится крайне актуальная задача фильтрации нежелательной почты. Сообщения рекламного характера содержат множество имён собственных, для анализа которых требуется подготовленная БД. При помощи материалов Википедии возможно назначение различным именам собственным оценки их близости к наиболее распространённым темам, встречающимся среди нежелательных сообщений. Другим примером потенциального применения БД на основе статей Википедии может быть система, воспринимающая для обработки утверждения на ограниченном естественном языке и оценивающая степень их достоверности. Задача фильтрации нежелательной почты во многом сводится к хорошо изученному методу классификации, и поэтому является более простой для исследований, чем система оценки достоверности утверждений на ограниченном естественном языке. Помимо этого, в последнем случае требуется дополнительное исследование набора хранимых атрибутов объектов и их назначения, поскольку они должны охватывать предельно большое количество характеристик объектов.

Сергей Мартишин, Марина Храпченко  
Москва, Институт системного программирования РАН

## Использование LAMP и MySQL Workbench в процессе обучения студентов

### Аннотация

Предлагается использование LAMP как базового пакета для выполнения студенческих проектов. Рассматриваются возможности MySQL Workbench для обучения студентов проектированию и реализации баз данных.

В настоящее время свободное программное обеспечение (СПО) завоевывает всё большую популярность. Особенно актуальным его применение видится для обучения студентов по различным направлениям подготовки, связанным с информационными технологиями, поскольку помимо теоретических знаний студенты должны обладать практическими навыками, позволяющими им после обучения, выполнения курсовых и дипломных проектов плавно перейти к работе в реальных условиях.

Используемые для обучения студентов программные средства должны быть, во-первых, доступны, а, во-вторых, обладать необходимой функциональностью.

Если говорить о доступности, то данное преимущество СПО очевидно. Возможность скачивания через Интернет бесплатной версии для установки позволяет любому студенту использовать СПО.

Что касается функциональных возможностей, то поскольку уже достаточно давно корпоративные системы строятся на Интернет-технологиях, то для разработок как в сети Интернет, так и в сетях Интранет необходимы средства, поддерживающие программирование для web.

В процессе обучения по специальностям, связанным с информационными системами (ИС), студентам для выполнения собственного проекта необходимы средства проектирования баз данных (поскольку база данных обычно является ядром ИС) и средства разработки программного обеспечения.

Для реализации такого рода проектов, как показал опыт, наиболее удобным оказалось использование LAMP — набора серверного программного обеспечения, названного по первым буквам входящих в

---

его состав компонентов. LAMP включает в себя операционную систему Linux, веб-сервер Apache, СУБД MySQL и язык программирования, используемый для создания web-приложений PHP.

Из списка перечисленных выше программных средств становится ясно, что этот пакет полностью способен обеспечить потребности разработчика, поскольку содержит все необходимые компоненты: операционную систему, веб-сервер, СУБД и язык программирования. Преимуществом использования LAMP является простота установки и полная согласованность используемого программного обеспечения.

Обратим особое внимание на возможности СУБД MySQL, поскольку наиболее ответственным и трудоемким при создании информационной системы является проектирование базы данных. При разработке баз данных часто в связи с большим объемом информации, необходимой для хранения, возникают сложности даже с построением логической модели, не говоря уже о физической модели, в которой необходимо учесть возможности конкретной системы управления баз данных (СУБД). В связи с этим многие СУБД создают свои CASE (Computer-Aided Software Engineering) средства. Они являются инструментом, позволяющим стандартизировать процесс разработки, автоматизировать процессы моделирования, проектирования и реализации баз данных, получить наглядные графические модели разрабатываемого ПО, дают возможность разделить значительные по объему задачи между разработчиками.

В настоящий момент удобным CASE-средством для разработчика на СУБД MySQL является средство *Workbench*, которое может быть получено на сайте <http://dev.mysql.com>. Его использование в значительной мере упрощает и делает более наглядным процесс проектирования баз данных, что особенно полезно для студентов, не имеющих большого опыта разработок.

MySQL *Workbench* — инструмент для визуального проектирования баз данных, который позволяет проектировать, моделировать, создавать и эксплуатировать БД. С его помощью можно представить модель базы данных в графическом виде, устанавливать связи между таблицами, наглядно представить инфологическую структуру разрабатываемой базы данных и сгенерировать таблицы и связи между ними из ER модели, восстанавливать структуры уже существующей на сервере БД (обратный инжиниринг), использовать редактор SQL запросов, позволяющий сразу же отправлять их на сервер и получать

ответ в виде таблицы, редактировать данные в таблице в визуальном режиме.

MySQL Workbench ориентирован на построение ER (Entity - Relationship) моделей. Для построения ER моделей поддерживаются две наиболее распространенные нотации: IDEF1X (методология структурного анализа для проектирования сложных ИС) и Information Engineering (IE), которая используется преимущественно в промышленности.

Также преимуществом MySQL Workbench являются не слишком высокие требования к аппаратному обеспечению. Для его установки достаточно обычного офисного компьютера с процессором, начиная с Pentium IV или аналогичного от AMD и 1 Gb оперативной памяти.

Использование языка PHP позволяет реализовать интерфейс к базе данных, а также требуемые функции ИС. Сервер Apache необходим для отладки проекта, если предполагается возможность удаленного доступа. Таким образом, набор СПО полностью обеспечивает потребности разработчика ИС и может быть широко использован для обучения студентов, помочь приобрести им опыт в проектировании и реализации баз данных, необходимый для их специальности и выполнять проекты различной степени сложности.

Настройки LAMP можно посмотреть на сайте одного из авторов [smartishin.narod.ru](http://smartishin.narod.ru).

### Список литературы:

1. MySQL [Электронный ресурс] — <http://www.mysql.com>, режим доступа свободный. — Загл. с экрана. — Яз. англ, франц., нем., японский.
2. IDEF [Электронный ресурс] — <http://www.idef.com>, режим доступа свободный. — Загл. с экрана. — Яз. англ.
3. Linux Online [Электронный ресурс] — <http://www.linux.org>, режим доступа свободный. — Загл. с экрана. — Яз. англ.

<b>ИЗВЕЩЕНИЕ</b>	ООО "С 13" <span style="float: right;">Форма № ПД-4</span> ИНН 7708654814 / КПП 770801001 Р.сч. 40702810300080001868 К.сч. 30101810100000000787 ОАО «УРАЛСИБ» г. Москва БИК 044525787 Коды: по ОКПО 84027582, по ОКОПФ 65											
	Вид платежа: <u>Редакционная подписка на журнал</u> <u>"Системный администратор"</u>											
	01	02	03	04	05	06	07	08	09	10	11	12
	Дата _____ Сумма платежа: _____ руб. _____ коп. <b>Информация о плательщике:</b> _____ <small>(Ф. И. О. почтовый индекс, адрес и телефон)</small> _____ _____ _____ Подпись _____											
<b>КВИТАНЦИЯ</b>	ООО "С 13" <span style="float: right;">Форма № ПД-4</span> ИНН 7708654814 / КПП 770801001 Р.сч. 40702810300080001868 К.сч. 30101810100000000787 ОАО «УРАЛСИБ» г. Москва БИК 044525787 Коды: по ОКПО 84027582, по ОКОПФ 65											
	Вид платежа: <u>Редакционная подписка на журнал</u> <u>"Системный администратор"</u>											
	01	02	03	04	05	06	07	08	09	10	11	12
	Дата _____ Сумма платежа: _____ руб. _____ коп. <b>Информация о плательщике:</b> _____ <small>(Ф. И. О. почтовый индекс, адрес и телефон)</small> _____ _____ _____ Подпись _____											