

**Государственное образовательное учреждение
высшего профессионального образования
«Елецкий государственный университет
им. И.А. Бунина»**

Центр свободного программного обеспечения

**Материалы
Всероссийской научно-практической
конференции
с международным участием**

**Информационные технологии
на базе свободного программного
обеспечения**

21 мая 2010 г.

Елец

УДК [681/3:430/49](063)

ББК 32.81+32.97

И 74

Печатается по решению редакционно-издательского совета
Елецкого государственного университета им. И.А. Бунина
от 28.05.10, протокол № 2

Редакционная коллегия:

Добровольский Н.М. – д. ф.-м. н., профессор, Тульский государственный педагогический университет имени Л.Н. Толстого

Козлов О.А. – д.п.н., профессор, заместитель директора Учреждения Российской академии образования «Институт информатизации образования» по общим вопросам и инновациям, Заслуженный работник высшей школы РФ, лауреат премии Правительства РФ в области образования

Хорошилов А.В. – к.ф.-м.н., н.с., Институт системного программирования РАН

Шишов В. В. – к.х.н., ст.н.с., Институт прикладной математики им. М.В. Келдыша РАН

Черный В.Л. – начальник отдела образовательных проектов ALT Linux

Андропова Е.В. – к.п.н, доцент, руководитель Центра СПО ЕГУ им. И.А. Бунина

Губина Т.Н. – к.п.н., ст. преподаватель ЕГУ им. И.А. Бунина

Губин М.А. – сотрудник Центра СПО ЕГУ им. И.А. Бунина, сертифицированный преподаватель компании ALT Linux

И 74 Информационные технологии на базе свободного программного обеспечения: материалы Всероссийской научно-практической конференции с международным участием. – Елец: ЕГУ им. И.А. Бунина, 2010. – 215 с.

В сборник включены материалы Всероссийской научно-практической конференции «Информационные технологии на базе свободного программного обеспечения». В статьях рассматриваются как теоретические, так и практические аспекты внедрения свободного программного обеспечения в учебно-воспитательный процесс средних, высших учебных заведений, а также организации деятельности государственных и коммерческих организаций.

Издание предназначено для преподавателей, научных сотрудников, аспирантов, студентов, работников сферы образования, других государственных и коммерческих структур, интересующихся свободными программными продуктами.

УДК [681/3:430/49](063)

ББК 32.81+32.97

ISBN 978-5-94809-449-6

© ЕГУ им. И.А. Бунина, 2010

© Авторы, 2010

СОДЕРЖАНИЕ

Свободное программное обеспечение в деятельности организаций и коммерческих структур

<i>Курячий Г.В.</i> Быстрое развёртывание компьютерного класса на базе Linux	6
<i>Демурин В.Б., Канарёв Д.И.</i> Нарушение авторских и смежных прав и опыт внедрения свободного программного обеспечения на территории города-курорта Анапа	9
<i>Бессонов С.Г.</i> Кризис копирайта.....	11
<i>Губин М.А.</i> Фактор свободного программного обеспечения в усилении информационной легитимации власти РФ	24
<i>Спендер В.А., Табакман А.П.</i> Правовые аспекты государственной политики по обеспечению информационной безопасности при внедрении свободного программного обеспечения ...	30
<i>Саенко И.Б., Баранов А.В.</i> Оптимизация схем дискреционного доступа к файлам в автоматизированных системах государственного управления, основанных на свободном программном обеспечении	37
<i>Настащук Н.А.</i> Программное обеспечение интеллектуальных информационных технологий	43
<i>Горошкин А.Н., Фисков М.М., Дербень А.М.</i> Концепция и технологии построения сетевого университета	49
<i>Еркимбаев А.О., Зицерман В.Ю., Кобзев Г.А., Трахтенгерц М.С.</i> Базы данных по свойствам наноразмерных объектов: практика использования свободного программного обеспечения	56
<i>Дербень А.М., Фисков М.М., Вейсов Е.А.</i> Проблемы адаптации пользователей к технологиям, применяемым в проекте «сетевой университет»	62
<i>Шалеев А.О.</i> Построение систем облачных вычислений на основе открытых технологий	68
<i>Голубев Р.А.</i> Сервер малого предприятия на основе Linux	73

Опыт использования свободного программного обеспечения в сфере среднего образования

- Черный В.Л.* Пакет свободного программного обеспечения для школ. Разработка и перспективы внедрения 78
- Логинов А.В., Лиликович С.А.* Использование Jasenu CMS для разработки сайтов средними учебными заведениями..... 82
- Сафонова Л.А.* Применение графического редактора Tux Paint в организации учебного процесса 86
- Сафонов В.И.* Использование свободного редактора 3D-графики GMAX в учебном процессе 91
- Мироненко Е.В., Губин М.А.* «Мысли о Linux» школьника провинциальной гимназии 96
- Козлов А.О.* Проблемы администрирования компьютерного парка школьных образовательных учреждений на основе технологий «виртуальный компьютер», «тонкий клиент», «бездисковая рабочая станция» 100
- Барышева А.С.* Опыт внедрения свободного программного обеспечения в профессиональном училище 104
- Ильина С.А., Соловьев Р.Ю.* Построение единой образовательной среды техникума на основе свободного программного обеспечения 111
- Подаев М.В.* Динамическое Web-приложение как средство развития мыслительной деятельности подростков при обучении геометрии 117
- Губина Т.Н.* Методика организации автоматизации расчётов в электронной таблице OpenOffice.org Calc 120
- Андропова Е.В.* Динамика развития и внедрения свободного программного обеспечения на региональном уровне 135

Опыт использования свободного программного обеспечения в сфере высшего образования

- Панюкова А.А.* Дистрибутив Linux для ЕГУ им. И.А. Бунина 146
- Смирнова М.Н.* Опыт применения свободного программного обеспечения в филиале ГУ КУЗГТУ в г. Прокопьевске..... 149
- Ивличева Н.А., Ивличев П.С.* Альтернативное программное

обеспечение в вузах министерства обороны	153
<i>Аверьянов Г.П., Будкин В.А., Дмитриева В.В.</i> Структура и программно-техническая реализация учебно-научной среды «электрофизика»	160
<i>Панюкова Т.А.</i> О системе оценок творческих заданий, выполняемых с помощью растрового графического редактора GIMP..	172
<i>Кормилицына Т.В.</i> Опыт изучения специальных дисциплин в педагогическом вузе на базе свободного программного обеспечения	179
<i>Яфаева Р.Р., Ситникова Л.Д.</i> Проведение курсов повышения квалификации с использованием LMS Moodle	185
<i>Кулямин В.В., Хорошилов А.В., Петренко О.Л.</i> Возможности свободного программного обеспечения для обучения студентов	194
<i>Дякина В.А.</i> Формирование информационно-технологической культуры будущих специалистов с использованием векторной графики	200
<i>Воробъев С.В.</i> Практика использования свободного программного обеспечения при подготовке будущих экономистов	203
Сведения об авторах	211

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ДЕЯТЕЛЬНОСТИ ОРГАНИЗАЦИЙ И КОММЕРЧЕСКИХ СТРУКТУР

УДК 004.3'1

БЫСТРОЕ РАЗВЁРТЫВАНИЕ КОМПЬЮТЕРНОГО КЛАССА НА БАЗЕ LINUX

Г.В. Курячий

Московский государственный университет им. М.В. Ломоносова
г. Москва, Россия

E-mail: george@altlinux.org

Аннотация

Быстрое развёртывание мобильного Linux-окружения в компьютерном классе строится по технологии LiveCD с использованием технологий сетевой загрузки/NFSROOT и последующей доработки штатными средствами дистрибутива Linux. Полученная технология позволяет решать гораздо более широкий круг задач, чем LiveCD, в том числе задачу быстрой подготовки произвольного профиля ОС, загружаемой на клиентские машины.

Ключевые слова

Linux-окружение, компьютерный класс.

Общая постановка задачи

В современной практике часто встаёт задача быстрого развёртывания временного и/или часто меняющегося окружения Linux в компьютерном классе. Это может потребоваться, например, в следующих случаях:

- временное развёртывание Linux-окружения для демонстрационных или учебных целей, а также для персональной работы;
- постоянное развёртывание Linux-окружения без модификации содержимого жёстких дисков (например, при невозможности/нежелании иметь двойную установку ОС);

- постоянная модификация одного или нескольких Linux-окружений, загружаемых на клиентские компьютеры (например, при подготовке информационного пространства для определённых курсов или при постоянной доработке окружения).

Все три подзадачи могут быть сравнительно неплохо решены с помощью методики LiveCD — специализированного дистрибутива Linux, загружаемого с CD или Flash и не требующего ни установки на жёсткий диск, ни модификации содержимого этого диска.

Однако загрузка дистрибутива-«прошивки» с медленного носителя проигрывает традиционной установке как в скорости дискового обмена, так и в гибкости настройки.

Между тем полноценное Linux-окружение можно загрузить по сети, используя методику NFSROOT. Встают вопросы, насколько LiveCD можно «скрестить» с NFSROOT и какие будут от этого преимущества?

LiveCD, NFSROOT и дистрибутивы ALT Linux

Начиная с ветки дистрибутивов ALT Linux 3.0, компанией и сообществом ALT Linux регулярно выпускаются Live-дистрибутивы. Носитель с дистрибутивом ALT Linux включает в себя т.н. *propagator* — небольшую подсистему, предоставляющую возможность загрузки и установки дистрибутива с произвольного носителя, в том числе и по сети с использованием NFS.

Несмотря на то, что установка дистрибутива с помощью *propagator* с различных носителей тестируется сообществом регулярно, работа Live режима по NFS проверяется нерегулярно.

Так, в дистрибутивах ALT Linux ветки 3.0 достаточно было настроить (с некоторыми уловками) серверную часть, подсунуть содержимое LiveCD в качестве NFSROOT и включить на клиентской машине сетевую загрузку, и полный аналог LiveCD мог быть развёрнут за считанные секунды.

В дистрибутивах веток 4.0 и 4.1, в частности в ПСПО, образ LiveCD непригоден для непосредственной загрузки по сети, но содержимое этого образа, после незначительной модификации, вновь даёт работоспособную конфигурацию [1].

В дистрибутивах Пятой Платформы, в частности в «ALT Linux Ковчег 5.0», настройка серверной части автоматизирована. Одно из свойств «Ковчега» — быстрое развёртывание установочного сервера: достаточно включить несколько функций посред-

ством Web-интерфейса и «подложить» образ установочного диска, чтобы вся схема исправно заработала [2].

Частично восстановлена функциональность работы LiveCD по сети. Фактически, работает всё, кроме автоматизации сетевых настроек. Для создания полноценного окружения применяется уже опробованный метод развёртывания содержимого LiveCD в NFSROOT и последующей модификации [3]. Главным образом модификация касается самой функциональности LiveCD: запрет монтирования разделов найденного жёсткого диска на запись, добавление пароля суперпользователя, тонкая настройка сети и т.п.

Планы на будущее

Получившаяся схема подготовки NFSROOT, в добавок к решению основной задачи быстрого развёртывания, даёт более удобный инструмент решения подзадач, заявленных в начале. В самом деле, системному администратору достаточно развернуть содержимое NFSROOT, а дальнейшие действия — установку, удаление и обновление пакетов, настройку особенностей системы, создание контента и т.п. — он может производить на сервере как на отдельном компьютере с помощью chroot. Внедрение подобной методики на факультете ВМиК МГУ им. М.В. Ломоносова включает в себя загрузку нескольких вариантов окружения, в зависимости от текущих потребностей в классе.

Главные недостатки NFSROOT, создаваемого по разработанной схеме:

- всё ещё низкая скорость файлового обмена по сравнению с работой с жёсткого диска и вообще зависимость от пропускной способности сети (особенно в момент системной загрузки);
- усложнённость полного процесса подготовки окружения: сначала кто-то собирает LiveCD, затем из него добывается содержимое, модифицируется (в том числе на предмет запрета некоторых особенностей LiveCD);
- некоторые типовые доработки внедрения нужны почти всегда, а в LiveCD отсутствуют (централизованная авторизация, домашние каталоги на сетевом диске и т.п.).

Что можно сделать по этим трём направлениям?

1. Реализовать кеширование NFS на клиентском диске (по образцу Sun Autoclient®). Подобную функциональность предоставляет, например, проект OpenAFS.

2. Использовать для развёртывания NFSROOT тот же инструмент, который используется для создания LiveCD. В ALT

Linux традиционно используется mkimage, предоставляющий все возможности для предварительной настройки и наполнения создаваемого образа.

Типовые доработки можно оформить в виде отдельных пакетов и т.н. hook-ов для mkimage В любом случае к выходу Live-дистрибува следующей ветки планируется целиком восстановить функциональность загрузки Live-образа по сети.

Используемые источники:

1. Г. Курячий. «Сервер "толстых" терминальных клиентов на ПСПО», <http://uneex.ru/FrBrGeorge/PspoThickClient>.
2. Г. Курячий. «Сетевая установка и сетевая загрузка», <http://uneex.ru/FrBrGeorge/BookP5/NetworkInstall>.
3. Необходимые модификации описаны там же.

УДК: 34.037:004.42

НАРУШЕНИЕ АВТОРСКИХ И СМЕЖНЫХ ПРАВ И ОПЫТ ВНЕДРЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ТЕРРИТОРИИ ГОРОДА-КУРОРТА АНАПА

В.Б. Демурин, Д.И. Канарёв

АФ ГОУ ВПО «Московский государственный гуманитарный университет им. М.А. Шолохова»

г. Анапа, Россия

E-mail: vbili@yandex.ru

Аннотация

В данной статье авторы дают обзор правонарушений, связанных авторскими и смежными правами, а также описывают опыт внедрения свободного программного обеспечения на территории города-курорта Анапа.

Ключевые слова

Авторское право, правонарушение, программное обеспечение.

В соответствии с п.1 ст. 7.12 Кодекса об административных правонарушениях (КоАП) РФ «ввоз, продажа, сдача в прокат или иное незаконное использование экземпляров произведений или фонограмм (в том числе и ПО) в целях извлечения дохода влечёт

наложение административного штрафа на граждан в размере от одной тысячи пятисот до двух тысяч рублей с конфискацией контрафактных экземпляров произведений и фонограмм, а также материалов и оборудования, используемых для их воспроизведения, и иных орудий совершения административного правонарушения; на должностных лиц - от десяти тысяч до двадцати тысяч рублей с конфискацией контрафактных экземпляров произведений и фонограмм, а также материалов и оборудования, используемых для их воспроизведения, и иных орудий совершения административного правонарушения; на юридических лиц — от тридцати тысяч до сорока тысяч рублей с конфискацией контрафактных экземпляров произведений и фонограмм» [1, с.72].

Что же касается подобных деяний, причинивших ущерб в крупных размерах, то согласно статье 146 ч.2 УК РФ они влекут за собой наказание «штрафом в размере до двухсот тысяч рублей или в размере заработной платы или иного дохода осуждённого за период до восемнадцати месяцев, либо обязательными работами на срок от ста восьмидесяти до двухсот сорока часов, либо лишением свободы на срок до двух лет» [2].

На территории города-курорта Анапа за последние 3 года было возбуждено свыше 10 уголовных дел по статье 146 УК РФ. Ущерб, причинённый правообладателям, исчисляется в сотнях миллионов рублей. И это только в одном небольшом городе. Можно представить себе, какие цифры мы получим, если просмотреть все подобные дела, совершённые на территории Российской Федерации.

Что же касается характера этих правонарушений, то они в основном связаны с продажей и использованием нелицензионного программного обеспечения. В числе наиболее популярных программ — Microsoft Windows XP и Microsoft Office 2003, без которых не обходится почти ни одна организация.

В связи с ужесточением борьбы с использованием нелегального программного обеспечения, которое в народе называется «пиратство», перед многими пользователями нелицензионных программных продуктов сегодня встаёт выбор: рискованная работа с взломанными версиями Microsoft Windows и привычного им пакета Microsoft Office, приобретение лицензии либо поиск более дешёвых или бесплатных аналогов. Например, помимо общепризнанных офисных пакетов Microsoft Office 2003 и Microsoft Office 2007, споры о преимуществах и недостатках которых ведутся и по

сей день, на рынке имеется ещё один продукт – OpenOffice, который становится всё более и более популярным.

На территории города-курорта Анапа уже многие учебные учреждения и коммерческие организации сделали свой выбор и отказались от использования офисного пакета Билла Гейтса. Среди наиболее известных из них — ЧОУ СПО Анапский индустриальный техникум, Анапский филиал ГОУ СПО «Новороссийский медицинский колледж», фирма недвижимости «Гранд» и ООО «Пожарзащита».

Что же касается операционных систем, то анапчане не спешат отказываться от Windows и переходить, например, на Linux.

Используемые источники:

1. Уголовный кодекс Российской Федерации (по состоянию на 1 апреля 2009). - Новосибирск: Сиб. унив. изд-во, 2009. - 160 с.
2. Консультант Плюс - законодательство РФ: кодексы, законы, указы, постановления Правительства Российской Федерации, нормативные акты. [Электронный ресурс] // Кодекс Российской Федерации об административных правонарушениях. – 2010. – URL: <http://www.consultant.ru/>

УДК 34.096; 004.9

КРИЗИС КОПИРАЙТА

С.Г. Бессонов

ООО "ЭЛСИС"

г. Нижний Новгород, Россия

E-mail: mail@aceler.ru

Аннотация

Я не буду рассказывать о том, что копирайт умрёт. Копирайт не может умереть — его просто никогда не существовало. То, что сегодня называется копирайтом, это сплетение правды и лжи, законов и беззакония, оно противоречит само себе, но за счёт этого чертовски жизнеспособно и прекрасно приспособляется к любым условиям. Такой копирайт не копирайт вовсе, но он не умрёт сам, он умрёт только тогда, когда кто-то наконец решится или довести его до логического конца, или отменить совсем. Оба варианта означают

смерть копирайта, каким мы его знаем, и рождение копирайта, каким он должен быть.

Ключевые слова

Копирайт, пиратство, свободное ПО.

Копирайтная модель и настоящая копирайтная модель

Копирайт — дословно «право на копирование», т.е. право автора или его представителя распоряжаться результатом своей интеллектуальной деятельности. В современном мире копирайт — это прежде всего бизнес-модель, по которой издатели, режиссеры или разработчики, зарабатывают деньги. Как большинство представляет себе обычную копирайтную (проприетарную) модель, т.е. модель, основанную на непосредственной продаже произведений? Написал программу — продал коробку. Написал песню — продал диск. Написал книгу — продал книгу. Снял фильм — продал фильм. Назовём эту модель «официальная копирайтная модель».

Если эта модель работает, то я королева английская, а вам меня читать дальше не надо, у вас и так всё хорошо. Эта модель не работает. Она никогда не работала и, по-видимому, никогда работать не будет. Какая модель работает вместо неё? Вместо неё работает настоящая копирайтная модель: написал программу — добрые люди раздали всем желающим, 10% желающих купило коробку. Написал песню — добрые люди раскидали по файлообменникам, 5% купило диск (цифры взяты с потолка). Написал книгу, снял фильм — сущности не меняются, в современном мире бесплатная версия того же товара имеет куда больший спрос.

В области программного обеспечения настоящая копирайтная модель выглядит ещё жёстче. Никогда не думали, зачем любая уважающая себя копирайтная программа изобретает собственный, ни с чем не совместимый формат хранения документов? Только не надо повторять маркетинговые лозунги о том, что эти форматы эффективны, компактны и, вообще, хороши для хранения. Сохраните один и тот же документ в doc и в odt и почувствуйте разницу.

Нет, закрытый формат нужен совсем для других целей — сначала добрые люди, раздающие всем желающим, подсаживают всех желающих на программу и на формат. Именно затем, чтобы все документы, которые были сделаны в этом формате, держали как якорь пользователей замечательной копирайтной программы

с закрытым форматом данных. И чтобы в один прекрасный момент пришёл дядя с дубинкой и потребовал заплатить за эту программу. И сколько раз это было!

Вот эта модель работает, и работает она прекрасно.

Ничего не напоминает? Да, наркобизнес. Самый рентабельный и прибыльный бизнес. С той лишь разницей, что в наркобизнесе нелегальна и первая и вторая дозы, а в копирайтном — только первая, суть от этого не меняется.

Едем дальше: как ведут себя наркоманы, которые не понимают, что они наркоманы? А никак: они покупают наркотик в известных им местах, используют и живут обычной жизнью. От передоза ПО не умирают, кстати, хотя были случаи передоза конкретно компьютерных игр. Но это несколько другая история, с копирайтом не связанная. А вот от излишней траты денег на наркотик вполне можно и помереть.

Кроме того, рейды за соблюдение авторских прав на программное обеспечение проводятся куда чаще и с куда большей помпой, чем рейды на наркопритоны. Во-первых, тут не надо даже никого выслеживать и вылавливать — «пасутся» вполне приличные фирмы, которые никуда не денутся. Во-вторых, сам рейд можно проводить одним неприметным сотрудником в присутствии понятых, никакого маски-шоу, никаких взрывов дверей. И если вы откажетесь пустить проверяющего в серверную, он так и напишет в журнальчике — «отказались пустить в серверную». И тогда вы автоматически станете подозреваемым и ждите масок. Со дня на день. Неплохо начать сушить сухари и запасти денег на второй комплект серверов.

Более того, никогда не задумывались, почему у копирайтных программ столь сложные взаимоотношения с версиями? Ту же Microsoft Windows можно купить десятком способов, по разным лицензиям и корпоративным программам, среди которых нет ни одного правильного. Про серверные версии или, тем более, учебные и другие «специальные» даже сами представители Microsoft вам не смогут объяснить, ссылаясь на своих «экспертов по лицензированию», которых в глаза, разумеется, никто не видел. (Партнёры Microsoft предлагают услуги этих специалистов иногда за отдельную плату. Естественно, законной силы такие рекомендации иметь не могут, ниже мы это обсудим.)

А зачем эти эксперты компании? Чтобы кто-то смог наконец купить лицензионное ПО? Это им совершенно не выгодно, куда

выгоднее собирать миллионные штрафы. Можно вообще не бороться с пиратством, раздавать диски на перекрёстках, чтобы потом собирать миллионные штрафы. И это, кстати, даже более привлекательная модель, чем настоящая копирайтная [1], не говоря уже об официальной.

Как ведут себя наркоманы, которые понимают, что они наркоманы? Ну, во-первых, это надо понять. Наркоманы обыкновенные, живые, понимают это по всё увеличивающимся расходам и по ломке. В мире ПО ломка та же. В один прекрасный день один из моих знакомых, фотограф, обнаружил, что его любимый софт помер. И что ему в ближайшем будущем светит переходить на что-то новое. И вот тут начинается ломка — надо перетаскивать фотографии и коллекцию из старого формата в новый, переделывать настройки, изучать новое ПО... На деле, кстати, изучение нового ПО заняло от силы день, всё остальное время ушло на подбор настроек. Сейчас он сидит на другом сорте этого софта, но суть от этого не меняется.

В области музыки ситуация иная. Здесь нельзя подсадить на формат, хотя есть подвижки и в этой области — например, активно насаждается миф о том, что звук с проигрывателя компакт-дисков будет всяко лучше звука с файлов, даже сжатых т.н. Lossless кодеками, т.е. без потерь в качестве. Чего уж там говорить про какой-то убогий звук из mp3. А по факту именно в mp3 распространяется абсолютное большинство музыки в сети, поскольку именно mp3 позволяет без проблем обмениваться этой музыкой.

Так вот, подсаживание производится другим способом — через чарты, телевизоры и другие СМИ. Если в чарте написано, что большинство слушателей проголосовало за песню XYZ — во-первых, кто вам сказал, что это правда, во-вторых, а за что они ещё будут голосовать на данном радио, если песня XYZ крутится там каждые пятнадцать минут? Вот ваша первая доза, кушайте. Мелодия делается прилипчивой и въедливой, совсем не такой, как у Моцарта. И чтобы постоянно жевать эту мелодию, надо пойти и купить. Благо только, никто не стоит с палкой за то, что вы слушаете не лицензионную музыку.

Зато с палкой стоят там, где этой музыкой обмениваются. Первая доза-то звучала из радиоприёмников бесплатно, а вот за вторую надо отдавать деньги. И специальные надзорные органи-

зации, наподобие RIAA в США, удостоверятся, чтобы деньги были заплачены. В большом количестве. В очень большом.

И эта модель тоже работает.

С фильмами ситуация примерно та же, просто вместо чартов вам покажут три трейлера актуальных к показу в кинотеатрах фильмов. Трейлеры, естественно, составляются так, чтобы вам хотелось посмотреть прямо сейчас, и развешиваются даже на остановках, транслируются по радио и телевидению, вешаются в интернете. В общем, вы уже знаете, что стоит посмотреть, осталось пойти и посмотреть...в лапы МРАА. Потому что легально посмотреть фильм не в кинотеатре нельзя — официальные DVD/BD выходят заметно позже проката в кинотеатрах. Чем вызвана такая задержка, кроме желания половить и поштрафовать «миллионеров»?

С книгами же, как мне кажется, ситуация относительно приличная. То, что покупают на лотках, продаётся за счёт обложек. Нарисуйте красивую картинку на обложку и книгу купят. Единственный способ подсадить для издательств — печатать серии, чтобы покупатели покупали каждую книгу серии. Серии делают, да, но это не такой тяжёлый наркотик, как ПО. В остальном покупатели относительно свободны.

Блеск и нищета копирайта

Копирайтная, настоящая копирайтная модель в области софта предполагает, что компания, забившая свою нишу, останется в этой нише на веки вечные. Случаи смещения «королей» крайне редки, а кампании по такому смещению длительны и кровопролитны. И далеко не всегда побеждает лучший продукт, поскольку в ход идут совсем не конкурентные методы борьбы — конкурентные тут совсем неэффективны. В результате мы видим только то, что развитие проприетарных продуктов только изредка стимулируется обычно успехом свободных на этом же поле. Если никакой стимуляции нет — никакого развития нет. Лидером рынка графических редакторов является один и тот же продукт вот уже 15 лет (Adobe Photoshop, великий и ужасный, стал лидером рынка в версии 3.0, 1994-й год [2]). Лидером рынка офисных пакетов тоже является не менее известный пакет Microsoft Office, с долей под 80% [3]. Лидером рынка операционных систем для домашних компьютеров является одна и та же фирма больше двадцати пяти лет (Microsoft начинала ещё с MS-DOS, а это 1981-й год. Можно я не буду приводить ссылку?). Где конкуренция?

Немного есть на рынке антивирусов [4], но там просто две компании, а не одна.

Копирайт (настоящий, естественно) рождает монополии — больше ничего. Сколько в мире издателей аудиопродукции? Четыре [5]. Остальные есть, да, но они настолько малы и мало-численные, что ситуация на рынке описывается только одним словом — олигополия. Сколько компаний снимают кассовые фильмы? Тоже четыре, и они все сидят в Голливуде [6]. Остальные компании хорошо, если на местный рынок выпустят какое-нибудь кино (типа индийского, или российского, или польского), но скорее всего, ничего серьёзного не выйдет.

И при этом издательства не перестают жаловаться на пиратов. Постоянно, из года в год, идут жалобы. «Если вы не будете покупать наши диски, мы сдохнем, а авторам нечего будет кушать!». Мы уже тридцать лет не покупаем ваши диски. Когда ж вы сдохнете?

Не сдохнут. Потому что гораздо лучше жаловаться на убытки от пиратов, а потом под это дело заказывать всё более драконовские законы наказания этих самых пиратов и жить на выплату компенсаций от нарушения авторских прав. Кстати, тут же о компенсации. Никогда не думали, почему красивые, профессионально оформленные и качественно сломанные дистрибутивы популярных проприетарных программ лежат на торрентах всегда в максимальной комплектации? Если виста, то Ultimate, если MS Office, то в максимальной редакции, а если 1С, то со всеми конфигурациями?

Всё просто: во-первых, потому что пользователю приятнее работать в неограниченной версии, он так проще подсядет, а, во-вторых, сумма компенсации пропорциональна стоимости ПО.

В результате сегодня сложилась ситуация, когда некий усреднённый расчётный убыток Microsoft от нелегального ПО в России составляет более миллиарда долларов [7]. И почему Microsoft никак не разорится? Ясно же, что эта сумма абсолютно виртуальна, потому что, во-первых, платить её никто не будет, во-вторых, стоит заменить пиратский офис на лицензионный или на другой бесплатный, как этой суммы уже нет. Но на самом деле эта сумма — это кредит, причём кредит, который мы даём США, процитирую РБК: «Столь большие объёмы потерь от пиратства в России, к сожалению, это не просто оценки “антипиратского” альянса. ПРА ежегодно сдаёт отчёт в Министерство торговли США,

которое в свою очередь урезает льготы на экспорт для самых «пиратских» стран».

(<http://rating.rbc.ru/article.shtml?2008/02/18/31826058>)

Вам кажется, что спиратив, вы ничего особого не сделали — а вас посчитали и вы за это заплатите всё равно. Потому что вы наркоман, укравший дозу. И потому что кто-то хочет ваших денег. Всякий раз, когда вы пользуетесь проприетарным продуктом — вы платите, вне зависимости от того, легальный у вас продукт или нелегальный. Даже если спиратить отечественный проприетарный продукт, он попадёт в статистику, и вы всё равно будете платить, причём в США.

И ваша проблема в том, что тот софт и та музыка, которую вы взяли бесплатно, она вам и не нужна вовсе. Не нравится слушать Моцарта, хотите Бритни Спирс? А почему? Почему так совпало, что нравится вам только современная кассовая музыка? Почему так получилось, что вы смотрите только кассовые американские фильмы с одинаковыми сюжетами? Почему вы пользуетесь профессиональными офисными и графическими пакетами для трёхстраничных документов и удаления красных глаз на фотках?

Как вы думаете, что является показателем качества и успешности музыкальной группы? Количество платиновых альбомов? Количество проданных копий, количество заработанных денег? Садитесь, два. Единственным пока доступным критерием качества культурной ценности является время. Если через 25 лет будут слушать группу Queen, значит это хорошая группа. Деньги и альбомы — это показатель качества менеджера группы и её коммерческого директора.

Подумайте о своих музыкальных вкусах и об используемых программах. Может, стоит соскочить разок?

Страшно? Страшно. Вы боитесь за совместимость, за вовремя сданную работу, за наработанные навыки и привычки. Вам очень неуютно будет в новой программе — но не только потому, что это новая непонятная для вас программа. Гораздо хуже то, что вы резко станете меньшинством. Все люди как люди, а вы один с макинтошем. Знакомо?

Я уже не говорю о том, что вы можете потерять в эффективности из-за использования другого инструмента, если речь идёт о ПО. И вот это давление толпы, давление «стандарта де-факто» и «мышления де-факто», а также «музыкальных вкусов де-факто» и

«де-факто хороших фильмов» делает их человека покупателя десятков никому не нужных наборов байт. Даже если он конформист и не покупает все эти вещи, а честно крадёт первую дозу — он всё равно её покупает. Почему — смотри выше.

А чем, собственно, плохи вот эти инструменты, вот эти фильмы и вот эта музыка? Да ничем. Сами по себе программы, фильмы и музыка ни в чём не виноваты. Проблема именно в копирайте, всё, чего он касается, превращается из золота в грязь, из шедевра в призрак, образ шедевра, который можно продать сотни раз. Симулякр этого рынка похож на панель — одни и те же покупатели приходят за тем, чтобы получить свой миг наслаждения. А на следующий день они же придут за следующей порцией. А кто купил надувную женщину — тот всё равно платит владельцу борделя.

Пиратство поддерживает копирайт — это две стороны одной медали. Блеск и нищета идут рука об руку, и чем больше объёмы пиратства, тем богаче корпорации.

Как соскочить

В данной ситуации интересен вопрос: а почему вы в любом случае оказываетесь виноваты? Как вы думаете, кто может определить, контрафакт у вас на компьютере или нет? Скопирован фильм или выпущен официально? Распространяется ли музыка на вашем плеере бесплатно или нет?

Определяет это всё издатель. То есть только издатель имеет право казнить и миловать, определяя по своим критериям, правильно ли вы под него прогнулись и достаточно ли заплатили. И если вы крупный уважаемый клиент, то вам лучше сказать, что у вас всё нормально, чтобы записать себе в портфолио и продавать своё произведение дальше. А если с вас лучше взять денег, то лучше усмотреть в легально купленном ПО признаки контрафакта. А, вы не знали, что наше ПО нельзя использовать, как вы его используете? Надо покупать версию постарше, в два раза подороже. И штраф не забудьте. Наш консультант вам сказал, что можно? Где подпись консультанта? Впрочем, консультант гарантий не даёт.

Вот представьте себе типичный офисный десктоп, забитый программным «обеспечением де-факто», и попробуйте представить все лицензионные соглашения на проприетарные программы, которые там используются. А ещё не забудьте, что на многие программы издатели оставляют за собой право изменять это согла-

шение в одностороннем порядке, вывешивая обновления на сайте. Вы сможете разобраться во всех этих лицензиях со всеми этими обновлениями? Вряд ли. И никто не сможет, потому что многие лицензионные соглашения вообще противоречат действующему в России законодательству. Разумеется, издателю, если ему не будет выгодно засуживать вас за честно купленный диск, нет интереса поднимать этот факт. Вам, как человеку, которому работать надо, тоже. Поэтому всё остаётся в руке великого и ужасного издателя, бога и дьявола вашего компьютера. А ещё в руках BSA, ОБЭП-а и других интересных организаций, знакомство с которыми вы бы заводите не хотели.

Про фильмы и музыку ещё смешнее. Диск есть. Голограмма есть. Что ещё есть? Вы никогда не узнаете, лицензионный диск или нет. Зато есть РАО, которое по записи песни может установить, есть ли данная запись в её реестре — и вот уже вы должны много денег за лицензионный диск, который вы просто используете неудобным издателю (или, точнее, РАО) образом. А есть ведь ещё смежные права (т.е. права исполнителей, авторов аранжировок и др.), из-за которых вы не можете нормально прослушать даже Мендельсона [8], который умер в позапрошлом веке и на которого предъявлять права просто неприлично, как бы некоторым ни хотелось.

В интересах издателя чтобы вы платили за его собственность всегда. Налог на воздух такой, что вы можете косвенно оплатить, например, просмотр кино по телевизору. Да, никто не спрашивает у вас, смотрели ли вы то или иное кино, но вы отсмотрите рекламу, а потом по всплеску продаж рекламируемых товаров рекламодатель увидит что да, кино отсмотрено, можно давать ещё. И даст ещё, специально для тех, кто год назад не сходил в кино, полгода назад не купил DVD или не скачал бесплатно. В любом случае, как бы вы ни посмотрели кино, вы за него платите. А иногда и несколько раз.

И как всего этого избежать? Можно, конечно, взять красный флаг, забраться на броневилок, отформатировать винчестер и поставить православный Linux. По крайней мере за Linux вы никому ничего не должны (если это не Linux XP или что-нибудь в этом духе, полукоммерческая пародия на коммерческое ПО (безотносительно качества)).

Фильмы придётся смотреть авторские, возможно на языке оригинала. Нет, польские и финские фильмы ничем особо не ху-

же американских, а сюжеты там куда как более жизненные, это я вам как знаток говорю. Но вот обсудить всё это вам будет уже не с кем, все ваши знакомые будут смотреть «Аватар» в то время, как вы в очередной раз пересмотрите «Я обслуживал английского короля». С другой стороны, можно надеть на себя интеллектуальный образ и писать обо всех этих фильмах в своём блоге. Со ссылками на торренты, разумеется.

С музыкой проще — сайты независимых интернет-издательств вам в помощь (<http://www.jamendo.com/ru/>, <http://www.kroogi.ru/>, <http://www.freemusic-project.ru/> и другие). Иногда попадаются крайне приятные вещи, уж не хуже попсы. Кроме того, многие группы уже начинают практиковать модель бесплатного официального распространения музыки, отбивая её на концертах, «виниловых» тиражах или просто SMS-сками благодарности. Упомяну свежий альбом группы «Пелагея» и выложенные в сеть альбомы группы «Несчастный случай». Всегда проще использовать торренты, чем бороться с ними, ведь альбомы всё равно на этих торрентах будут — воюй не воюй.

С книгами, опять же, вообще всё хорошо — есть множество хороших авторов, помеченных как «неформат» у издательств, которые (авторы, не издательства) мечтают о том, чтобы вы их прочитали. Ищите в любых сетевых издательствах. Причём. Причём авторы в таких издательствах нередко получают больше, чем в больших и красивых [9], потому что интернет-издательства не тратятся на рекламу и офлайновые каналы, а просто передают все пожертвованные деньги авторам, забрав фиксированный процент на собственное развитие.

Проблемами, с которыми вы столкнётесь при таком образе жизни, будут отсутствие какой-либо ориентации во всём многообразии Indie музыки, фильмов и литературы (хотя бывают местные хит-парады и литературные чтения, ищите), причём с линуксом даже проще — это уже почти индустрия. Ну и вторая проблема — на какие темы вы будете разговаривать с друзьями и коллегами? Они потребляют совсем другую музыку, фильмы, книги и операционные системы. И готовьтесь, что вы станете фанатом, даже если фанатом вы не станете. Просто вы единственный, кто будет говорить про Linux, творчество Ким-Ки-Дука и песни Светланы Косенко, когда вокруг все будут говорить про Windows, фильмы Кэмерона и кто-там-сегодня-вместо-Бритни. Потому что их много, а вы один.

Кризис копирайта

Есть способ проще. Чтобы убить настоящую копирайтную модель, достаточно использовать официальную копирайтную модель. Обычная копирайтная модель, в том виде, в каком она преподносится для покупателей, — это сладкая пилюля, утопия, она совершенно нежизнеспособна. Так используйте это свойство.

Для начала определитесь с программами. Оставьте то, что вам нужно, и поинтересуйтесь, сколько это стоит. Тщательно выполняйте все указания из лицензии. Это, в конце концов, забавно. У вас быстро появится желание вместо покупки ПО воспользоваться бесплатными или свободными аналогами. Давите. Лучше воспользуйтесь урезанными версиями — они дешевле и интоксикация от них происходит гораздо быстрее. Windows стоит восстановить ту, которую туда поставил производитель, вместе с теми тоннами говен, которые он туда положил. В конце концов, зачем покупать коробку с Windows, если на свежкупленном компьютере уже стоит великолепная виста? Узнаете через пару недель.

Если вы музыкант, найдите на дисках, прилагаемых к оборудованию, урезанные рекламные версии «больших» программ, таких как Cubase LE. Фотографы наверняка найдут PhotoShop Elements в фотоаппаратах, а вашим детям наверняка понравятся демо-версии игр с дисков, приложенных к видеокarte. Ну и что, что двухлетней давности.

Через пару недель вы приобретёте стойкую ненависть к копирастам — это я вам обещаю. Разница между использованием пиратских программ и «настоящих» — как между туризмом и эмиграцией. Уже через пару дней какой-нибудь с трудом развиваемый Muse (<http://www.muse-sequencer.org/>) вам будет казаться вполне неплохим музыкальным редактором на фоне абсолютно убогого Cubase LE. Станет проще выучить интерфейс GIMP, чем заставлять себя работать в глубоко урезанном фотошопе. Демо-версии игр, понятное дело, ничто по сравнению с 15-ю длиннейшими кампаниями Battle for Wesnoth.

Что ж, если PhotoShop вам действительно нужен, купите себе немного PhotoShop. Чем больше денег вы потратите, тем лучше привяжете себя к копирайтной модели, тем проще вам будет отказаться от очередной покупки, заменив какую-то не очень важную программу на бесплатную. Не так страшен платный фотошоп, как его обновления, цены на которые поднимут в вашей груди волну справедливого народного гнева (Порядка 7500 р. за

обновление: http://www.softkey.ru/catalog/program_ver.php?ID=66291&CID=3940).

Когда гнева будет достаточно, ставьте Linux. Да, он другой, после многолетних привычек Windows он кажется неудобным, но если сравнивать не Vista Ultimate, а с предустановленной Home Basic, то Linux покажется вам самой функциональной системой в мире, да там ещё и куча софта на все случаи жизни. Того самого софта, который вы же, возможно, осуждали месяц назад как никому не нужный, потому что есть сладкая пилуля пиратского полнофункционального проприетарного нечто. И не забывайте о последней подлянке копирастов — подержанное ПО очень трудно будет продать. А уж обновления и вовсе продать на вторичном рынке нереально.

С музыкой, как я уже говорил, проще — просто выкидываете всё лишнее и начинаете формировать свои новые вкусы. Заведите себе ещё одну папку, кроме папки «моя музыка», заведите папку «моя правильная музыка», куда будет попадать только правильная, оплаченная из вашего кармана или вовсе бесплатная музыка. И потихоньку привыкайте слушать музыку из второй папки. Через пару месяцев вам уже будет жалко те два бакса, что вы потратили на хит, поскольку хит уже приелся, а денег не вернёшь. И вы начнёте спокойнее относиться к музыке и ценить то, что действительно ценно.

То же с фильмами, только через некоторое время вы обнаружите, что переплатили 100\$ производителям винчестеров, потому что их теперь нечем забить. Вот незадача...

Зачем всё это надо? Если честно, вам это нафиг не надо. У вас и так всё хорошо, проводить революцию в отдельно взятом доме вы не собираетесь. Даже если у вас уже везде линукс, вы всё равно не Столлман и не будете покупать себе новый OLPC, просто потому что там свободный BIOS. Правда? Ну если уж купили Ричард Мэттьювич, очень приятно, что вы меня читаете. Всё это написано здесь для того, чтобы вы сами представили себе жизнь по закону и оценили масштабы огромного надувательства под названием «копирайт». Потому что нарушение копирайта касается каждого. И потому что настоящий копирайт не может работать в рамках закона.

Всё это надо делать на уровне предприятий, а лучше — на уровне государства, а не отдельных домохозяйств. По одной причине — сегодня от настоящей копирайтной модели выигрывают

только ограниченное число монополистов. Они перечислены в сносках к первой главе, и российских, кстати, там нет. И всё это, к счастью, делается государствами и крупными компаниями, умеющими считать деньги.

Ежегодный убыток от пиратства в России, повторяюсь, оценивается в 4 млрд.\$ [7]. Этого уже достаточно, чтобы перевести всю государственную инфраструктуру РФ на свободное ПО, включая доработку этого ПО под отечественные стандарты делопроизводства (или стандарты делопроизводства под свободное ПО), обучение сотрудников на местах, внедрение и сопровождение. И это будет куда эффективнее прямого стимулирования американской экономики путём покупки ПО от иностранных производителей, проверено европейцами [10]. Потому что чем больше покупаешь, тем больше должен. Только деньги виртуальны, их нет, вместо них иногда, изредка, на местах покупают иностранное лицензионное. Очень хороша в качестве приличного гвоздя в гроб копирастии и копирайт-зависимости от США программа по внедрению свободного ПО в образование. Да, на линуксе сейчас мало кто работает. Но если сразу целое поколение выйдет из школы с каким-никаким знанием линукса и особенно OpenOffice.org, ему уже не придётся краснеть за свою «инаковость», они будут мейнстримом.

В чём кризис копирайта? Кризис копирайта в том, что чем дальше, тем сложнее копирайту противостоять свободному творчеству и самому себе. Копирайт стоит на той же основе самовыражения, что и творчество, он не может запретить творчество и устраивать суды с самодеятельными авторами. Максимум, до чего может дотянуться настоящий копирайт, — запрет на переработку собственных образов, создание коллажей, запрет на использование даже «изначального образа».

Именно свободное ПО, музыка и авторские фильмы — главные противники настоящей копирайтной модели. Именно они, а не пиратство, потому что пиратство — это карманный наркодилер, работающий на копирайт, а свободное ПО и творчество работает против копирайта.

С другой стороны, копирайт не может справиться с самим собой правильным. Когда государство по-настоящему крутит гайки, настоящая копирайтная модель вырождается в официальную, а официальная модель абсолютно нежизнеспособна, она разваливается даже без конкуренции.

А сейчас мы наблюдаем оба процесса: с одной стороны, возрастает влияние и уровень свободных разработок, как программ, так и музыки и фильмов, появляется какая-никакая инфраструктура, сообщества, а с другой, — в попытках выжать больше денег с помощью судебных процессов законодательно копирайт всё более притягивают к обычному, заменяя настоящий. Копирайт не умрёт, он уже умер. Умрёт настоящий копирайт, но по нему никто не будет плакать, потому что его не было. И только пройдёт череда банкротств и слияний крупных компаний, изыскивающих другие бизнес-модели и увольняющих бывших звёзд. Впрочем, звёзд они увольняют и без всякого кризиса.

СС-BY-SA

Используемые источники:

1. <http://www.securitylab.ru/news/386438.php>
2. http://artjunkies.net/2007/07/13/istorija_sozdanija_adobe_photoshop.html
3. <http://www.networkworld.com/news/2009/060409-forrester-microsoft-office-in-no.html>
4. <http://www.channelinsider.com/c/a/Security/Cracking-Open-the-Antivirus-Market/>
5. http://ru.wikipedia.org/wiki/Большая_четвёрка_лейблов_звукозаписи
6. http://en.wikipedia.org/wiki/Concentration_of_media_ownership
7. <http://worldquality.ru/index.php?name=news&op=view&id=198>
8. <http://www.newsru.com/russia/08oct2009/mendelson.html>
9. http://community.livejournal.com/za_lib_ru/176823.html
10. http://ec.europa.eu/enterprise/sectors/ict/files/2006-11-20-flossimpact_en.pdf

УДК 323.213; 004.9

ФАКТОР СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УСИЛЕНИИ ИНФОРМАЦИОННОЙ ЛЕГИТИМАЦИИ ВЛАСТИ РФ

М.А. Губин

ГОУ ВПО «Елецкий государственный университет
им. И.А. Бунина»

г. Елец, Россия

E-mail: gubinma@yandex.ru

Аннотация

В статье рассматриваются вопросы корректировки политического курса в сторону свободного программного обеспечения как базовой основы информационной безопасности РФ, а также вопросы усиления информационной легитимации власти в контексте использования свободных программных систем в социальной и информационной деятельности государства.

Ключевые слова

Свободное программное обеспечение, информационная политика, информационная безопасность, легитимность власти.

Проблема информационной безопасности в современном мире в полной мере может быть актуальна не только для «развитых» стран мира, таких как США, Япония, Германия, Великобритания, Сингапур и Китай, но и для национальной безопасности России. Процесс демократизации политической системы в России привнёс существенные изменения в оборот информации. Национальные интересы в начале 1990-х гг. позиционировались малозначимыми для большого количества политических «тяжеловесов», предпринявших реализацию и защиту партикулярных интересов в условиях постсоветской трансформации. Увеличение количества политиков является объективным следствием, по мнению Э.Тоффлера, «распада политических структур» [1, с.161], что несёт в себе явную угрозу информационной безопасности.

Такая проблема имеет место и в современной России. В период президентства Б.Ельцина информационная политика ключевых акторов сосредотачивалась исключительно вокруг их собственных интересов, связывающихся с проблемами политической легитимации. Тем не менее, государственная власть оставалась весьма значительным актором, сумевшим сформулировать приоритеты информационной политики и обезопасить себя от различных делегитимационных вызовов, обозначившихся в период демократизации, сохранить свою безопасность.

Действительно, президентство В.Путина и политика нового президента России Д.Медведева, значительно изменили подход к пониманию и реализации информационной политики. Заметно обозначился акцент на национальный компонент в порождаемом правящим классом политическом тексте. Возможность примене-

ния данных мер в своё время отмечалось Э.Гидденсом, определявшим каждое государство как информационное. По его мнению, «государственная власть подразумевает рефлексивный сбор, хранение и управление информацией, которая необходима для администрирования» [2, С.178]. Усиление национальной компоненты политического текста, что и происходит в настоящий момент в России, в теоретических построениях автора, требует «высокой степени интеграции его административных функций, а это в свою очередь требует более высокого уровня информационного обеспечения» [2, С.179].

Корректировки государственного курса в информационной политике происходят в свете изменения дефиниции правящего режима. Усиление правящего режима в России, присутствие заметных и логичных акцентов на консолидацию и совпадение интересов основных политиков в сфере информационной политики означает и реализацию одновременно более действенных и силовых решений по обеспечению безопасности информационного пространства.

Тем не менее, даже усиление правящего режима не гарантирует ему информационной безопасности. Разумеется, данная проблема не является актуальной только для России. Другие политические системы также в полной мере могут испытывать подобные вызовы. Даже относительная стабильность политической системы не может обещать отсутствие проблем, связанных с оборотом информации.

Одним из основных факторов, влияющих на информационную безопасность государства, может являться применение в госсекторе тех или иных программных продуктов, основанных на собственных или свободных правах их пользования. Ситуация может ещё и усугубляться тем, что использование собственных решений «порождает угрозу технологического замыкания» [3, с.96]. Так называемый «Мюнхенский проект», начавшийся в ноябре 2001 года, можно считать началом массового использования свободного программного обеспечения (СПО) в государственном секторе. В рамках этого проекта муниципалитет города Мюнхена начал искать альтернативы используемому в городском хозяйстве проприетарному программному обеспечению и перевёл на СПО более 14 тыс. рабочих мест. Результаты этого проекта позволили Европейскому Союзу сформировать чёткое отношение к СПО, которое было оформлено Еврокомиссией в 2003 году в ви-

де рекомендательного документа «The IDA Open Source Migration Guidelines» [4].

Примеру Германии в 2004-2010 годах последовали Австрия, Франция, Бельгия, Норвегия, Дания, Венгрия, Италия, Малайзия, Индия, Бразилия, Китай, некоторые муниципальные образования США.

В прошлом Д.Медведев был противником свободных систем, однако в последнее время его мнение резко изменилось, и он стал одним из сторонников перехода на СПО в российских государственных и образовательных учреждениях. Выступая в Кремле 12 февраля 2009 года с вступительным словом на заседании Совета по развитию информационного общества в России, президент высказал свою точку зрения на СПО в РФ: «Ещё одна тема — это информационные технологии в социальной сфере. Сейчас нужно начинать массовое обучение школьных учителей новым технологиям. Мы, собственно, пытались это делать в рамках национального проекта. Наверное, кое-что удалось, но пока это только самое начало. Надо подумать и о том, чтобы двинуться дальше — к использованию отечественного свободного программного обеспечения. Я этой темой занимался, результаты у нас есть, мы подготовили уже свои программы, которые позволяют создать, по сути, продукт абсолютно качественный, на основе свободного программного обеспечения, но привязанный уже к нашим реалиям» [5].

Чем же мотивируется «желательный» переход на свободные продукты? В условиях всё более широкого применения информационных технологий в органах власти и огромных сумм, выделяемых из бюджета на закупку программ, многообразие, сложность и непрозрачность лицензионных политик фирм-производителей проприетарного ПО приводит к созданию прекрасной среды для развития коррупции. В отличие от собственного ПО, СПО не требует лицензионных отчислений за каждый установленный экземпляр программы. Проведя открытый конкурс, государство может заплатить фирме-разработчику за поставку необходимого варианта программного обеспечения и дальше тиражировать его без ограничений, не заботясь о лицензионных выплатах.

К сожалению, в России все попытки создания и внедрения именно «русской оси» с «русскими программами» натываются на бюрократический аппарат не способных «делиться отпущенными сверху средствами» чиновников. Чего стоит, к примеру, статьи та-

кого плана, как «СПО в школах: посредники превратили проект в грустный анекдот» [6].

В идеале прозрачная схема платежей и отсутствие сверхприбылей у разработчиков ПО позволят существенно оздоровить атмосферу на государственном IT рынке.

Не секрет, что и многие проприетарные приложения от известных производителей содержат недокументированные возможности — так называемые «пасхальные яйца». В то же время в Доктрине информационной безопасности Российской Федерации прямо указано, что «угрозами безопасности информационных и телекоммуникационных средств и систем, как уже развёрнутых, так и создаваемых на территории России, могут являться... внедрение в аппаратные и программные изделия компонентов, реализующих функции, не предусмотренные документацией на эти изделия» [7]. Возможность доступа к исходным текстам (кодам) СПО даёт возможность контролировать наличие в программах «задних дверей» и недокументированных функций.

Другой вид и ещё более серьёзная угроза — это возникновение зависимости от владельца и поставщика проприетарного ПО [8]. При наличии претензий к ним или их отказе от дальнейшего сотрудничества на приемлемых условиях, а также их ухода с рынка, при применении СПО государственные органы будут в состоянии нанять для доработки другие компании, поскольку открытые исходные тексты программ и условия свободного лицензионного договора позволяют это сделать.

Несмотря на то, что разработка свободных программ интернациональна и неотделима от мирового сообщества разработчиков, сервис-ориентированные услуги по их адаптации, внедрению, поддержке и развитию, как правило, оказывают национальные фирмы. Государству и обществу выгоднее создавать рабочие места для высококвалифицированных программистов на своей территории и платить деньги за поддержку и развитие программного обеспечения отечественным фирмам, чем зарубежным. Локализация СПО на языки народов РФ может быть проведена самостоятельно носителями языка, что будет способствовать сохранению культуры многонационального народа Российской Федерации.

Открытый доступ к исходным кодам большого количества уже готовых свободных приложений позволяет легко их адаптировать под конкретные нужды пользователей и создавать на их

основе новые необходимые решения. Благодаря своей открытости и доступности, СПО обладает рядом преимуществ с точки зрения развития информационного общества в России (согласно Стратегии развития информационного общества в Российской Федерации).

Важным является и вопрос цены лицензии. Если принять условие, что совокупная стоимость владения проприетарного и свободного ПО приблизительно равны, СПО всегда будет дешевле проприетарного ПО на стоимость, потраченную на приобретение лицензий.

Можно признать, что сегодня применение СПО в государственном секторе может стать реальностью. Оно экономически выгодно, дружелюбно к гражданам, позволяет вкладывать средства в развитие национального IT рынка и противостоять коррупции и информационным угрозам, что подтверждается многочисленными успешными внедрениями по всему миру. Ожидается дальнейшая активизация усилий по изучению позитивного опыта использования свободного программного обеспечения в государственных структурах и предприятиях Российской Федерации, а также стимулирование его тиражирования и применения во все более широких масштабах. Эти действия будут способствовать обеспечению конкурентоспособности России, развитию IT-отрасли РФ, совершенствованию системы государственного управления на основе использования информационных и телекоммуникационных технологий, укреплению информационной безопасности политической системы и усилению позиций акторов информационной политики РФ.

Используемые источники:

1. Тоффлер Э. Война и антивоенная. М. 2005.
2. Giddens A. The Nation State and Violence: Volume Two of a Contemporary Critique of Historical Materialism. Cambridge. Polity. 1985.
3. Губин М.А. Проблемы перехода на стандарты СПО в документообороте // Информационные технологии на базе свободного программного обеспечения: материалы научно-практического семинара. - Елец: ЕГУ им. И.А.Бунина, 2009.
4. <http://www.netproject.com/docs/migoss/>
5. <http://archive.kremlin.ru/text/appears/2009/02/212834.shtml>
6. <http://www.cnews.ru/news/top/index.shtml?2009/07/06/353142>

7. http://www.rg.ru/oficial/doc/min_and_vedom/mim_bezop/doctr.shtm

8. Более подробно см. Опыт внедрения международного стандарта ISO/IEC 26300:2006 в органах государственной власти и местного самоуправления Российской Федерации. Материалы семинара. - Алексин, 2007.

УДК 004.725.7

ПРАВОВЫЕ АСПЕКТЫ ГОСУДАРСТВЕННОЙ ПОЛИТИКИ ПО ОБЕСПЕЧЕНИЮ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРИ ВНЕДРЕНИИ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

В.А. Сплендер, А.П. Табакман

СПБГУ ИТМО, МИРЭА

г. Москва, Россия

E-mail: sva-777@mail.ru, alexeytabakman@yandex.ru

Аннотация

Правительством РФ одобрен переход в государственных учреждениях и некоммерческом секторе экономики на СПО. Сохранится ли необходимый уровень безопасности автоматизированных систем при миграции на СПО? Не ставит ли ПО с открытыми исходными кодами под угрозу эти системы и информацию, которую они обрабатывают? Какие принципиальные различия в подходе к обеспечению безопасности в системах с открытым исходным кодом и коммерческих системах?

Ключевые слова

Безопасность, свободное программное обеспечение, открытый код.

В соответствии с постановлениями Правительства РФ «О Министерстве связи и массовых коммуникаций Российской Федерации» от 02.06.2008 г. №418 и «О некоторых мерах по обеспечению информационного взаимодействия государственных органов и органов местного самоуправления при оказании государственных

услуг гражданам и организациям» от 25.12.2007г. №931 в нашей стране были выработаны мероприятия государственной политики в сфере информационных технологий, первоочередными из которых можно назвать:

- формирование государственных информационных ресурсов и обеспечение доступа к ним;
- установление требований к формату данных в государственных информационных системах в целях защиты интересов государства в области авторского права в сфере информационных технологий;
- создание и обеспечение функционирования информационных систем, задействованных в информационном взаимодействии по оказанию государственных услуг федеральными органами исполнительной власти; определение требований к информационной безопасности (ИБ) в государственных информационных системах.

В качестве основы в государственных информационных системах предлагается использовать так называемое свободное программное обеспечение (СПО).

В самом общем смысле СПО — это такое программное обеспечение (ПО), которое широкий круг пользователей может свободно запускать, копировать, распространять, изучать, изменять и улучшать.

В различных зарубежных и отечественных источниках под «распространением ПО» понимаются любые действия, конечным результатом которых является предоставление широкому кругу пользователей возможности использования программы по её прямому функциональному назначению. Однако такая трактовка понятия «распространение ПО» вступает в противоречие с юридическим значением этого термина, означающего продажу или иное отчуждение экземпляров произведения (пп.2) п.2 ст.1270 ГК), что применительно к программе для ЭВМ и, как правило, означает продажу материальных носителей (CD, DVD и т.п.) с записью дистрибутива программы.

Термин же «СПО» является понятием более высокого уровня по отношению к таким понятиям, как «ПО с открытым исходным кодом», «бесплатное ПО», «коммерческое ПО» и др. Однако необходимо отметить, что суть термина «СПО» находится в противоречии с понятием «несвободное ПО», хотя в случае, когда для классификации используется такой критерий, как наличие или отсут-

ствие в лицензионном договоре условий, ограничивающих свободу пользователей ПО, в совокупности они составляют полный объем понятия «программное обеспечение».

Кроме того, для данной категории ПО не существует установленного набора лицензионных условий в сфере информационных технологий, достаточных для того, чтобы отнести его к разряду «свободных». Поэтому в настоящее время присутствует ряд неопределённостей, которые не позволяют чётко определить понятие СПО посредством указания исчерпывающего набора признаков, что в обязательном порядке необходимо учитывать при подготовке нормативной документации, посвящённой вопросам использования «свободного ПО».

Тем не менее, Правительством РФ выбран и одобрен переход в государственных учреждениях и некоммерческом секторе экономики на СПО. Помимо перечисленных выше критериев СПО, в пользу его выбора можно отнести такие положительные качества, как возможность наличия у пользователей четырёх видов свободы:

- свободы запускать программу для любых целей (свобода 0);
- свободы изучать, как программа работает, и адаптировать её для своих нужд (свобода 1), при этом доступ к исходному коду является необходимым для этого условием;
- свободы повторно распространять копии с целью оказания помощи другому пользователю (свобода 2);
- свободы улучшать программу и опубликовывать результаты работы по улучшению программы для пользы всего общества (свобода 3), при этом доступ к исходному коду также является необходимым для этого условием.

В 2008-2009 годах о планах по расширению использования СПО сообщили:

- Министерство связи и массовых коммуникаций Российской Федерации (начата разработка нормативной и правовой базы по внедрению и использованию СПО, ведётся разработка рекомендаций для органов власти по использованию СПО, проведён открытый конкурс по внедрению системы электронного документооборота на базе СПО);
- Министерство экономического развития и торговли Российской Федерации (ведётся разработка рекомендаций по использованию СПО, проведено большое количество открытых конкурсов, связанных с разработками ПО на базе СПО);

- Министерство образования и науки Российской Федерации (продолжен проект по внедрению в школы GNU/Linux; начато обучение учителей работе в GNU/Linux);

- Минздравсоцразвития РФ (разработан реестр информационных систем здравоохранения, социального развития и трудовых отношений. Несмотря на то, что реестр включает и коммерческие решения для социальной сферы, в качестве одной из целей данной системы заявлено обеспечение плавного перехода учреждений социальной сферы к преимущественному использованию свободного программного обеспечения).

В связи с этим наблюдается активизация деятельности заинтересованными организациями по юридическому сопровождению разработки и внедрения СПО.

Так, в период с 2008 по 2009 годы Министерство связи и массовых коммуникаций РФ разрабатывает и публикует следующие документы:

- Концепцию развития разработки и использования свободного программного обеспечения в Российской Федерации – первый официальный документ, определяющий понятие «свободное программное обеспечение» в России;

- проект приказа Министерства связи и массовых коммуникаций Российской Федерации «Об утверждении Методических рекомендаций по разработке и приобретению программного обеспечения для использования в органах государственной власти и бюджетных учреждениях»;

- проект Плана перехода органов государственной власти и бюджетных учреждений на использование свободного программного обеспечения.

Таким образом, был дан старт активному внедрению СПО в государственных учреждениях и некоммерческом секторе.

С 2005 года в государственных учреждениях различных уровней СПО начинает внедряться и использоваться в рамках Федеральной Программы административной реформы Российской Федерации и других пилотных проектов, проводимых в федеральных министерствах, ведомствах, субъектах Российской Федерации, районных и областных администрациях, образовательных и медицинских учреждениях, научно-исследовательских институтах и многих других государственных образованиях.

Но при внедрении СПО справедливо возникает достаточно много вопросов. Сохранится ли необходимый уровень безопасно-

сти автоматизированных систем при миграции на СПО? Не ставит ли ПО с открытыми исходными кодами под угрозу эти системы и информацию, которую они обрабатывают? Какие принципиальные различия в подходе к обеспечению безопасности в системах с открытым исходным кодом и коммерческих системах?

Одним из важных рыночных требований, предъявляемых к операционным системам, является их сертификация в различных организациях и комиссиях. В особенности это касается государственных учреждений, деятельность которых связана с обработкой и хранением персональных данных, конфиденциальной информации и сведений, составляющих государственную тайну.

Наиболее влиятельные международные стандарты информационной безопасности объединяются под названием Общих критериев (Common Criteria). В разработке положений Common Criteria принимают участие представители более чем 20 стран (США, Европа, Япония и другие развитые страны), стандарты безопасности принимаются в рамках организации ISO.

Отечественные методы сертификации безопасности операционных систем постепенно приближаются к западным. С 2004 года введён стандарт ГОСТ Р ИСО/МЭК 15408 «Общие критерии оценки безопасности информационных технологий», представляющий собой перевод стандарта Common Criteria [1].

Более широкое использование Linux-систем в госсекторе и появившиеся перспективы масштабных государственных проектов в этой сфере закономерно приводят к более активной сертификации программно-аппаратных решений на базе СПО. В 2007-2008 годах сертификаты ФСТЭК получают Red Hat Enterprise Linux AS на всех серверных платформах IBM, ALT Linux 4.0 Server Edition и ALT Linux 4.0 Desktop Professional, ряд продуктов Mandriva (Mandriva PowerPack+, Corporate Server, Corporate Desktop, Mandriva Flash) в 2009 г. проходит сертификацию решение Open Referent on Software United, включающее Red Hat Enterprise Linux 5.2, платформу совместной работы IBM Lotus Domino 8.0.1, систему контроля исполнительной дисциплины Сервер Open Referent 3.1.4. [2]

Защита информации всегда очень волновала военных. Именно в недрах Министерств обороны возникли первые критерии и стандарты безопасности программ и операционных систем. В числе таких изобретений и метод доступа к информации, именуемый мандатным доступом (MAC, Mandatory Access Control) [1].

Концепция информационной безопасности предоставляет стандартизированную модель политики безопасности, применяемую в государственных учреждениях и включающую описание уровней и классов секретности, права доступа различных субъектов и специфику ввода и вывода информации из системы. Это позволяет применять в автоматизированных системах универсальные методы обеспечения безопасности информации, в том числе и в различных операционных системах.

В современных открытых системах используется сочетание классической дискреционной модели безопасности UNIX с мандатной и ролевой моделями доступа, которые реализуются специальными модулями безопасности, такими, как SELinux (Security-Enhanced Linux), RSBAC (Rule Set Base Access Control) и grsecurity.

Существует также значительное количество средств обеспечения безопасности ПО с открытым исходным кодом, которые можно условно разделить на несколько типов:

- шифрование и защита данных при помощи криптографии;
- защита учётных данных пользователей и усовершенствованные современные и перспективные модели аутентификации;
- ограничение возможностей пользователей и аудит их активности;
- защита локальных и сетевых файловых систем;
- проверка целостности и предотвращение компрометации программ;
- сканирование уязвимостей системы;
- обнаружение вторжений и атак;
- построение сложных систем маршрутизации и межсетевого экранирования.

В совокупности эти технологии образуют многоуровневую систему доступа и контроля АС, которая позволяет на практике реализовать практически любую политику безопасности.

Немалую роль в достижении высокого уровня безопасности свободных систем играют открытость исходных текстов и принципы разработчиков, проповедующих использование только открытых стандартов.

Широко применяемые в военных системах обработки и хранения информации методы мандатного и многоуровневого доступа сейчас доступны создателям коммерческих и других систем,

чувствительных и критичных к утечкам информации и (или) её потере [1].

Проведённый анализ позволяет дать ответ на вопрос о том, почему несмотря на достаточные преимущества, в нашей стране только за последние 5-7 лет начали внедрять и применять современные информационные технологии на базе СПО, в отличие от ряда других стран, которые в течение десятков лет уже реализуют централизованные программы внедрения и развития СПО.

Таким образом, к основным факторам, сдерживающим распространение СПО, по-нашему мнению, можно отнести ряд специфических для России особенностей, к которым относятся:

- законодательная неопределённость статуса СПО (отсутствие определения СПО в Гражданском кодексе РФ);
- отсутствие чёткого понимания роли и места СПО в современной информационно-коммуникационной инфраструктуре со стороны государственных учреждений [3];
- масштабное распространение компьютерного пиратства;
- мощное лоббирование использования проприетарного ПО со стороны производителей информационных технологий и продукта;
- слабая (до недавнего времени) поддержка развития СПО со стороны государства [2].

Однако важной положительной тенденцией, наметившейся в России в последнее время в сфере информационных технологий, становится начало внедрения СПО в инновационном стиле, которое характеризуется законодательной деятельностью использования СПО, разработки связанных с этим концепций и проектов в отраслевых ведомствах и, одновременно, с практической реализацией внедрения информационных технологий на базе СПО в государственных учреждениях и некоммерческом секторе.

При этом необходимо отметить, что, несмотря на государственную поддержку использования СПО, на быстрые результаты его внедрения рассчитывать не приходится. Это связано прежде всего с длительностью периода процедур по прохождению и утверждению подобных государственных программ, но факт обращения государства к данной теме информационных технологий уже является важным сигналом для многих ведомственных учреждений [2].

Основанные на открытых технологиях отечественные решения использования СПО, в том числе и в части информационной

безопасности, превосходят зарубежные коммерческие аналоги не только низкой ценовой политикой, но и принципами открытой разработки и активного участия сообщества пользователей.

Используемые источники:

1. SELinux: теория и практика безопасности, Федосеев Алексей, IBM developerWorks Россия, 2010, <http://www-128.ibm.com/developerworks/ru/library/selinux/>.
2. Аронова Л. Исследование российского рынка СПО, 2010, <http://www.opennet.ru/>.
3. Беспалов П. В. Опыт повышения квалификации госслужащих в области использования СПО. - Российская академия государственной службы при Президенте РФ, 2009.
4. Власова А. Разработка свободного программного обеспечения в интересах государства. - ООО "КОРУС Консалтинг", www.korusconsulting.ru

УДК 004.451.52.056.5; 004.056.53

ОПТИМИЗАЦИЯ СХЕМ ДИСКРЕЦИОННОГО ДОСТУПА К ФАЙЛАМ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ ГОСУДАРСТВЕННОГО УПРАВЛЕНИЯ, ОСНОВАННЫХ НА СВОБОДНОМ ПРОГРАММНОМ ОБЕСПЕЧЕНИИ

И.Б. Саенко, А.В. Баранов

Санкт-Петербургский институт информатики и автоматизации
Российской академии наук
г. Санкт-Петербург, Россия
E-mail: ibsaen@mail.ru

Аннотация

Рассматриваются механизмы дискреционного управления разграничением доступа к файлам операционной системы МСВС, используемой в автоматизированных системах государственного управления. Приводится постановка задачи оптимизации схемы дискреционного доступа к файлам и алгоритм её решения.

Ключевые слова

Автоматизированная система, защита информации, несанкционированный доступ, файловая система.

В автоматизированных системах государственного управления (АСГУ) одним из приоритетных принципов построения является обеспечение повышенных требований по обеспечению безопасности обрабатываемой информации. Для реализации этих целей в качестве базового элемента общего программного обеспечения, разрабатываемых и совершенствуемых АСГУ, применяется операционная система МСВС. Система МСВС разработана на основе системы *Linux*, поэтому она вобрала в себя многие технические решения, свойственные операционным системам свободного программного обеспечения.

Одной из характерных черт МСВС следует считать наличие дополнительных средств защиты информации от несанкционированного доступа (НСД), имеющих комплексную организацию. В состав данного комплекса средств защиты информации (КСЗИ) входят следующие компоненты: средства управления доступом, средства аутентификации, средства контроля целостности данных, средства управления печатью документов, средства аудита, средства администратора обеспечения безопасности информации и другие средства. Средства управления доступом, входящие в состав КСЗИ от НСД, играют ключевую роль в данном комплексе и включают в себя средства дискреционного управления доступом и средства мандатного управления доступом. Средства дискреционного доступа применяются практически во всех АСГУ. Средства мандатного управления доступом применяются только в случае, когда обрабатываемая информация имеет несколько уровней конфиденциальности. Поэтому настоящая статья ограничивается только дискреционным доступом.

Для реализации дискреционного управления доступом используются традиционный для *Linux* механизм битов доступа, и, дополнительно к нему, предусмотрена реализация механизма списков прав доступа (*ACL – Access Control List*). Оба механизма реализуются на уровне файловой системы и служат для задания прав доступа субъектов (пользователей и процессов) к её объектам, в качестве которых рассматриваются как файлы, так и каталоги. Однако совместное использование данных механизмов, как показывает опыт их применения в АСГУ, требует от администраторов безопасности и разработчиков принятия рациональных решений по построению схем разграничения доступа. Схема разграничения доступа представляет собой совокупность векторов, списков и матриц, элементами которых полностью определяются

полномочия доступа субъектов к объектам файловой системы. Недостаточная проработка схемы разграничения доступа приводит к появлению уязвимостей безопасности, которые могут привести к нарушению либо конфиденциальности, либо доступности информации, содержащейся в файлах и/или каталогах.

Для постановки задачи оптимизации схемы дискреционного доступа в АСГУ вначале следует рассмотреть сущность указанных выше механизмов.

Механизм битов доступа предполагает предоставление субъектам доступа трёх видов прав доступа в отношении файлов и каталогов: «чтение», «запись» и «выполнение» [1]. Традиционно указанные виды прав доступа обозначаются определёнными буквами соответствующих английских слов: «*r*» (*Read* – чтение), «*w*» (*Write* – запись) и «*x*» (*eXecute* – выполнение). Дополнительно к перечисленным видам прав доступа существует особое право – «владелец». Право «владелец» обеспечивает субъекту возможность изменения прав доступа к объекту.

По отношению к каждому объекту доступа все множество пользователей системы подразделяется на три категории: «владелец» (*user owner*), «группа-владелец» (*group owner*) и «все» (*others*). К категории «владелец» может быть отнесён только один пользователь, имеющий право «владелец» по отношению к этому объекту. Наличие категории «группа-владелец» предполагает наличие механизма объединения пользователей в группы. К категории «все» относятся все пользователи системы, кроме пользователя, назначенного владельцем объекта, и пользователей, входящих в группу владельцев объекта.

Для хранения информации о правах доступа каждому объекту файловой системы сопоставлена запись, содержащая соответствующие поля для хранения идентификаторов «владельца» и «группы-владельца» и поле назначенных соответствующим категориям пользователей прав. Последнее поле включает три группы по три бита, которыми закодированы права доступа (рис. 1).

Наличие соответствующего права доступа кодируется значением «1», отсутствие – значением «0».

Механизм битов доступа применяется при задании прав для большинства объектов файловой системы, одинаковым образом используемых основной частью пользователей. Однако данный механизм не является достаточно гибким и в ряде более сложных

ситуаций не обеспечивает возможность задания индивидуальных прав доступа к объекту для нескольких пользователей.

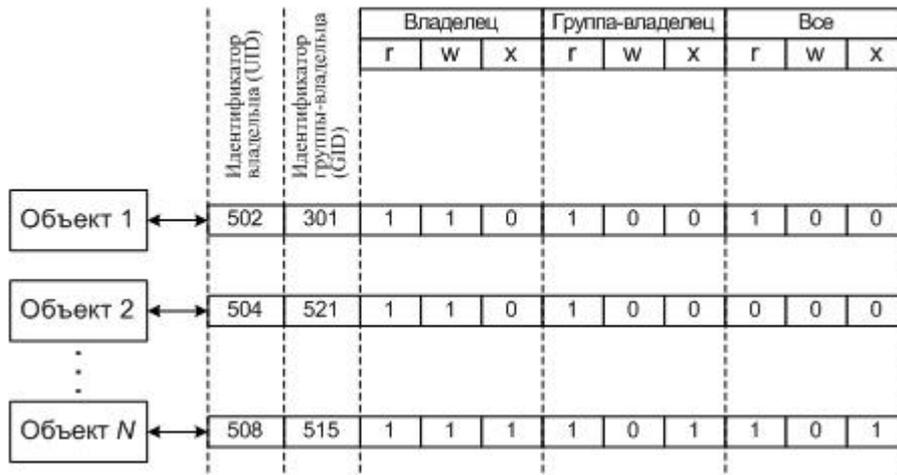


Рис.1. Биты доступа

Сложные комбинации прав доступа могут быть установлены путём использования механизма списков контроля доступа (ACL) [2]. С помощью механизма списков контроля доступа можно задавать права на уровне отдельных пользователей и/или групп пользователей и тем самым достичь существенной детализации в задании прав. Списки применяются при работе с файлами, для которых требуется, например, задать разные права доступа для нескольких определённых пользователей.

Для реализации механизма ACL с каждым объектом файловой системы (файлом или каталогом), помимо записи, обеспечивающей функционирование битов доступа, дополнительно ассоциирован список пользователей с указанием их прав доступа к объекту (рис. 2).

Совместное применение механизмов битов защиты и списков ACL предполагает наличие правила, обеспечивающего учёт результатов функционирования указанных выше дискреционных механизмов при окончательном принятии решения о предоставлении доступа субъектов к объектам. В качестве такого правила используется следующее. Запрашиваемый субъектом вид доступа к объекту будет разрешён, если указанный вид доступа разрешён одним из механизмов дискреционного управления доступом.

Постановка задачи оптимизации схемы дискреционного доступа к файлам (каталогам) в АСУ сводится к следующему. Исходными данными являются: множество субъектов доступа, множество объектов доступа и требуемая схема дискреционного доступа.



Рис. 2. Списки контроля доступа (ACL)

Требуемая схема доступа задаётся в матричном виде и определяет требуемые полномочия доступа субъектов по отношению к объектам файловой системы, которые необходимо установить в АСГУ.

Критерием оптимизации в постановке задачи является минимум количества элементов в списках доступа. Данный критерий целесообразен, исходя из необходимости снижения трудозатрат администратора безопасности и, тем самым, уменьшения вероятности появления ошибки в его действиях, связанных с формированием схемы разграничения доступа.

Предложим алгоритм решения данной задачи.

Пусть $\mathbf{Z}=\{Z_{ij}\}$ – матрица, задающая требуемую схему доступа. Значение Z_{ij} определяет набор полномочий доступа, которые необходимо установить между субъектом i и объектом j . Положим, что Z_{ij} является группой, состоящей из трех битов: $Z_{ij} = \{z^{(r)}_{ij}, z^{(w)}_{ij}, z^{(x)}_{ij}\}$, где $z^{(r)}_{ij}$ – бит, задающий право субъекта i на чтение объекта j , $z^{(w)}_{ij}$ – право на запись, $z^{(x)}_{ij}$ – право на выполнение.

Пусть $\mathbf{X} = \{X_j\}$ – вектор, определяющий биты доступа всех пользователей к объектам файловой системы. Значение X_j также представляется тройкой битов: $X_j = \{x^{(r)}_j, x^{(w)}_j, x^{(x)}_j\}$, где $x^{(r)}_j$ – бит, определяющий право всех субъектов на чтение объекта j , $x^{(w)}_j$ – право на запись, $x^{(x)}_j$ – право на выполнение.

Списки ACL определим формально в виде матрицы $\mathbf{Y} = \{Y_{ij}\}$. Значения Y_{ij} имеют вид и смысл, аналогичные значениям Z_{ij} . В

каждом i -ом столбце матрицы \mathbf{Y} содержатся полномочия, которые субъект i имеет относительно всех объектов, а в каждой j -ой строке – полномочия, которые все субъекты имеют над объектом j .

Задача заключается в том, чтобы найти такие \mathbf{X} и \mathbf{Y} , чтобы, во-первых, реальная схема разграничения доступа, образованная совместным применением механизмов \mathbf{X} и \mathbf{Y} , полностью совпадала с требуемой схемой \mathbf{Z} . Во-вторых, количество ненулевых элементов в \mathbf{Y} должно быть минимальным.

Предлагаемый алгоритм решения данной задачи включает следующие шаги.

Шаг 1. Присваиваем матрице \mathbf{Y} значение матрицы \mathbf{Z} .

Шаг 2. Организуем перебор объектов доступа и выбираем объект j в качестве текущего.

Шаг 3. Организуем перебор значений Z_{ij} по индексу i .

Шаг 4. Находим значение X_j в соответствии со следующими выражениями: $x^{(r)}_j = \min\{z^{(r)}_{ij}\}$, $x^{(w)}_j = \min\{z^{(w)}_{ij}\}$, $x^{(x)}_j = \min\{z^{(x)}_{ij}\}$.

Шаг 5. Организуем перебор значений Y_{ij} по индексу i .

Шаг 6. Уточняем и при необходимости модифицируем значение Y_{ij} в соответствии со следующими выражениями: $y^{(r)}_{ij} = z^{(r)}_{ij} - x^{(r)}_j$, $y^{(w)}_{ij} = z^{(w)}_{ij} - x^{(w)}_j$, $y^{(x)}_{ij} = z^{(x)}_{ij} - x^{(x)}_j$. В результате количество нулевых элементов в матрице \mathbf{Y} увеличивается за счёт наличия ненулевых элементов в \mathbf{X} , причём количество ненулевых элементов в \mathbf{Y} достигает минимально возможного значения.

Шаг 7. Фиксация \mathbf{X} и \mathbf{Y} в качестве результата решения задачи.

Применяя предлагаемый алгоритм к спискам ACL , представленным на рис. 2, и полагая, что эти списки соответствуют матрице требуемой схемы доступа, получим, что $X_1 = (101)$ и $X_N = (100)$. В списке ACL объекта 1 нулевые значения получают элементы, соответствующие субъектам с $UID = 502$ и $UID = 502$, а в списках объекта N – элементы субъектов с $UID = 505$ и $UID = 517$.

Отметим, что, исходя из найденных значений \mathbf{X} , биты, приведённые на рис. 1 для объекта N , являются некорректными. Рис. 1 устанавливает для X_N значение (101). В результате этого появляется уязвимость безопасности, заключающаяся в реализации всеми субъектами запрещённого права на выполнение объекта N , приводящая к нарушению конфиденциальности информации в АСГУ.

Предложенный алгоритм оптимизации схемы дискреционного доступа к файлам был реализован на языке C++, вошёл в арсенал средств защиты информации администратора безопасности и хорошо зарекомендовал себя в ходе функционирования АСГУ.

Используемые источники:

1. Симоненко С.Н. Обзор дискреционных механизмов управления доступом применительно к информационным системам // [Электронный ресурс]. – Режим доступа: http://www.philipovich.ru/Library/Books/ITS/wwwbook/IST7/sp_lit/liter.htm#172. – 08.05.2010.
2. Корт С.С. Теоретические основы защиты информации: Учебное пособие. – М.: Гелиос АРВ, 2004. – 240 с.

УДК 007:681.3.06(075.8)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Н.А. Насташук

Омский государственный педагогический университет
г. Омск, Россия

E-mail: nat_lion@mail.ru

Аннотация

Данная статья посвящена проблеме программной реализации интеллектуальных информационных технологий. Представлены определённые категории программного обеспечения, реализующего интеллектуальные информационные технологии и носящего статус «free» software (свободное программное обеспечение).

Ключевые слова

Искусственный интеллект, интеллектуальные информационные технологии, свободное программное обеспечение.

Возрастающая роль информации в жизни современного общества и использование в различных сферах профессиональной деятельности как традиционных информационных технологий, так и интеллектуальных информационных технологий (ИИТ) позволяют выделить информационную компетентность в качестве ба-

зовой. При этом процесс развития науки информатики в сторону интеллектуализации информационных технологий делает значимым их применение в различных областях деятельности человека. В этой связи возникает объективная потребность у современных специалистов в овладении знаниями о структуре, методах, средствах и функциональных возможностях программного обеспечения, реализующего ИИТ, методик организации интеллектуальных информационных систем на базе ИИТ, а также в умении применять его для решения слабо формализованных и не формализованных задач.

Однако в последнее время наблюдается ограничение использования не лицензионного программного обеспечения, подкреплённое соответствующими юридическими законами. Последнее вынуждает применять как в образовательных учреждениях, так и на рабочих местах программное обеспечение, носящее статус *«free» software*, т.е. *свободное программное обеспечение*, под которым будем понимать модель, при которой базовый набор имущественных прав передаётся («лицензируется») владельцу / пользователю каждого экземпляра произведения.

Учитывая вышесказанное, целью данной статьи является аналитический обзор свободного программного обеспечения (СПО), реализующего ИИТ, и предложение конкретных программных средств ИИТ к использованию в образовательных учреждениях. При этом сюда также будут отнесены и программные средства, выступающие как демонстрационные версии (или демо-версии). Названия программных средств, которые носят статус «free» будут отмечены символом «*», а демо-версии и «free trial» (подобные демо-версии) – «%».

Под *интеллектуальными информационными технологиями* будем понимать технологии обработки информации и решения задач с помощью компьютера, опирающиеся на достижения научной области «Искусственный интеллект». *Искусственный интеллект* – это раздел информатики, изучающий возможность обеспечения разумных рассуждений и действий с помощью вычислительных систем и иных искусственных устройств. Целью данного научного направления является разработка аппаратно-программных средств, позволяющих пользователю-не программисту ставить и решать свои, традиционно считающиеся интеллектуальными задачи (т.е. слабо формализованные и не формализованные

задачи), общаясь с ЭВМ на ограниченном подмножестве естественного языка [2].

По мнению исследователей научной области «Искусственный интеллект» (Э.В. Попова, Г.С. Поспелова, Д.А. Поспелова и др.), противопоставление не формализованных методов искусственного интеллекта формальным методам математического моделирования весьма условно, так как реализованное на компьютере программное обеспечение ИИТ полностью формализовано и однозначно интерпретируется процессором.

В соответствии с направлениями искусственного интеллекта – программно-прагматическое и бионическое – выделены следующие основные линии исследований в научной области «Искусственный интеллект», а также соответствующие им ИИТ и программное обеспечение, реализующее данные технологии (табл. 1).

Таблица 1

Программное обеспечение, реализующее ИИТ

ИИТ	ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
программно-прагматическое направление	
представление и обработка знаний	
технология экспертных систем	<ul style="list-style-type: none"> •оболочки экспертных систем: <i>FuzzyCLIPS 6.02*</i>; <i>Малая экспертная система 2.0*</i>; <i>EXSYS%</i>, <i>EXSYS Corvid%</i>; •экспертные системы: <i>VisiRule Graphical Business Rules Free Trial%</i> (графическая экспертная система); •программные средства, имеющие в своем составе функциональный компонент, который реализует механизм работы экспертной системы: <ul style="list-style-type: none"> – системы поддержки принятия решений / информационно-аналитические системы: <i>Project Expert Trial%</i> (модуль «Экспертное заключение»); <i>Sales Expert 2 Demo%</i> (модуль «База знаний»); – программные средства общего назначения: <i>табличный процессор Open Office.org Calc*</i> (логическая функция If());
инструментальные сред-	<ul style="list-style-type: none"> •логическое программирование: <i>Visual Prolog 5.2 Personal Edition*</i> (рассматри-

ИИТ	ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ства разработки интеллектуальных информационных систем	вается как СПО только для образовательных целей); <i>LPA PROLOG Professional 3.83*</i> ; <i>WIN-PROLOG 4.800 28-Day Free Trial%</i> ; •язык С, интегрированная производственная система: <i>CLIPS*</i> ; •производственный язык: <i>OPS5*</i> ;
бионическое направление	
машинное обучение и самообучение	
технология интеллектуального анализа данных (Knowledge Discovery in Databases, Data Mining и OLAP-технологии)	•программные средства, имеющие в своем составе функциональный компонент, который реализует данную технологию: <i>Deductor Studio Academic 5.2*</i> (рассматривается как СПО только для образовательных целей; содержит функциональные компоненты «Дерево решений», «Ассоциативные правила», «Линейная регрессия», реализующие методы и алгоритмы Data Mining; реализует OLAP - многомерный оперативный анализ данных); •информационные системы класса ERP: <i>Microsoft Ахapta%</i> (реализует OLAP - многомерный оперативный анализ данных); •универсальные программные системы интеллектуального анализа данных: <i>Recognition-Demo.1.00%</i> ;
нейросетевые технологии	•программные надстройки (add-ins), которые могут быть интегрированы в программные средства общего назначения (интегрированные нейропакеты): <i>Alyuda Forecaster XL%</i> ; <i>NeurOK Excel Neural Package%</i> ; •информационно-аналитические системы, имеющие в своём составе функциональный компонент, который реализует данные технологии: <i>Deductor Studio Academic 5.2*</i> (рассматрива-

ИИТ	ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
	<p>ется как СПО только для образовательных целей; содержит функциональные компоненты «Нейросеть» и «Карта Кохонена», реализующие многослойную нейронную сеть и самоорганизующуюся карту Кохонена соответственно);</p> <ul style="list-style-type: none"> •библиотека компонентов, встраиваемая в некоторую среду программирования: <i>NeuralBase*</i> (библиотека компонентов для Delphi реализует сеть Хопфилда и многослойную нейронную сеть обучаемую по алгоритму обратного распространения); <i>SOMBase*</i> (библиотека компонентов для Delphi, реализующая самоорганизующиеся карты Кохонена).
эволюционное моделирование, генетически алгоритмы	<ul style="list-style-type: none"> •библиотека компонентов, встраиваемая в некоторую среду программирования: <i>GeneBase*</i> (библиотека компонентов для Delphi реализует генетические алгоритмы);
обработка текстов на естественном языке и распознавание образов	
автоматизированное извлечение знаний из текста	<ul style="list-style-type: none"> •интеллектуальные программные средства для работы с текстовыми документами (анализаторы текста): <i>TextAnalyst 2.01*</i>;
автоматическая классификация и кластеризация документов	<ul style="list-style-type: none"> •интеллектуальная поисковая система: <i>Nigma*</i> (результаты поиска группируются в так называемые кластеры);
интеллектуальный поиск информации	<ul style="list-style-type: none"> •интеллектуальная поисковая система: <i>Nigma*</i>; <i>Exactus*</i>.

Кроме представленных выше ИИТ (табл.1) существуют *интегрированные ИИТ*, объединяющие в комплекс множество информационных технологий (как традиционных, так и интеллектуальных) [1]: 1) базы данных, хранилища данных, базы знаний; 2)

управление документооборотом; 3) автоматизированное извлечение знаний из текста; 4) автоматическую классификацию и кластеризацию документов; 5) автоматическое реферирование и аннотирование; 6) приобретение знаний от экспертов; 6) машинные перевод; 7) интеллектуальный анализ данных; 8) автоматическое распознавание образов; 9) поддержку принятия решений и др.

В настоящее время интегрированные ИИТ получили широкое распространение в управлении знаниями, и, как следствие этого, появилось программное обеспечение, реализующее данные технологии, которое получило название *системы управления знаниями* [1, 3]. На сегодняшний момент возможно найти только демо-версии подобных программных средств, например, к таким системам можно отнести XpertRule Knowledge Builder.

Представленное в данной статье программное обеспечение ИИТ используется при организации учебного процесса студентов следующих специальностей в рамках специальностей:

– (030100) «Информатика», дисциплина «Основы искусственного интеллекта»,

– (080801) «Прикладная информатика» (в образовании, в менеджменте) и направления подготовки (080800) «Прикладная информатика» с дальнейшей специализацией в образовании, дисциплины «Интеллектуальные информационные системы» и «Информационные системы образовательного назначения».

Несмотря на существующие в данный момент препятствия в изучении и применении программного обеспечения ИИТ как в образовательных учреждениях, так и на рабочих местах, связанные, в первую очередь, с финансовой стороной данного вопроса, будем надеяться, что это не остановит дальнейшее развитие информационной компетентности будущих и настоящих специалистов в области ИИТ. И, в свою очередь, предлагаемые программные средства позволят восполнить пробел в знаниях современного специалиста о методах, средствах и функциональных возможностях программного обеспечения ИИТ и умении применять его для решения слабо формализованных и не формализованных задач в соответствующей деятельности.

Используемые источники:

1. Башмаков А.И. Интеллектуальные информационные технологии / А.И. Башмаков, И.А. Башмаков. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2005. – 304 с.

2. Гаврилова Т.А. Базы знаний интеллектуальных систем / Т.А. Гаврилова, В.Ф. Хорошевский. – СПб.: Питер, 2001. – 382 с.
3. Джарратано Д. Экспертные системы: принципы разработки и программирования, 4-е издание.: Пер. с англ. / Д. Джарратано, Г. Райли – М.: ООО “И.Д. Вильямс”, 2007. – 1152 с.

УДК 004.41/.42

КОНЦЕПЦИЯ И ТЕХНОЛОГИИ ПОСТРОЕНИЯ СЕТЕВОГО УНИВЕРСИТЕТА

А. Н. Горошкин, М.М. Фисков, А.М. Дербень

ГОУ ВПО «Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнёва»

г. Красноярск, Россия

E-mail: fisk@sibsau.ru

Аннотация

Рассмотрена концепция построения сети крупного предприятия, реализующая возможности мобильности пользователей, гибкости, снижения затрат на содержание сети при одновременном повышении защищённости системы.

Ключевые слова

Сетевое хранилище, распределённая файловая система, информационная безопасность, защита информации, модульная архитектура.

Работы выполнены при поддержке г/к №02.740.11.0621 «Информационно-вычислительное моделирование сложных систем».

Широкое использование информационных технологий требует повышенной защищённости систем. Электронная информация как продукт, используемый в производстве, имеет уникальные особенности, отличающие её от физического продукта:

1. Прежде всего это возможность копирования, при котором исходный продукт не меняет своих свойств.
2. Копирование или дублирование информации не требует существенных материальных затрат.
3. Неконтролируемое распространение информации может нанести существенный вред производственному процессу.

Все это выводит на первый план задачи информационной безопасности систем. В настоящее время Россия взяла курс на построение информационного общества, и совершенно очевидно, что это совершенно невозможно выполнить без реализации многоуровневой и весьма совершенной системы безопасности. Но реализация любых мер по информационной безопасности вступает в противоречие с необходимостью неограниченного расширения функциональных возможностей информационных систем, их гибкости и доступности. Преодоление этих трудностей только организационными мерами весьма затруднительно. Вместе с тем повышение роли информационных технологий в производственном процессе требует обеспечения мобильности пользователей, позволяющей предотвратить «расползание» защищаемой информации.

Ряд классических технологий имеют свои недостатки. Так, например, классическая реализация рабочих станций в сетевых информационных системах является одним из самых уязвимых мест с точки зрения информационной безопасности. В этой технологии отсутствует возможность полного централизованного контроля программного обеспечения (ПО) функционирующего на рабочем месте пользователя. Попытки такого контроля, конечно же, предпринимаются. В качестве примера можно привести повсеместное требование применения антивирусных систем и применения технологий удалённой настройки, как в Active Directory Service (AD) и некоторых других системах. Однако потенциальные уязвимости отсутствия полного реального контроля ПО на рабочем месте пользователя регулярно проявляются.

Ещё одной существенной проблемой применения классической технологии рабочей станции является низкая мобильность пользователей. Переход пользователя с одного компьютера на другой лишает его возможности использования «своих» документов и своего программного обеспечения без ущерба для безопасности системы, потому что при реализации технологий перемещаемых профилей происходит их неконтролируемое размножение на компьютерах, с которых выполнялся сетевой доступ. Решением этой проблемы могло бы стать применение мобильных компьютеров, получающих все большее и большее распространение, однако они оказываются ещё более уязвимы. Они, как правило, регулярно перемещаются за пределы контролируемой зоны и пределы контролируемой сети, а потому возможности для реализации

угроз информационной безопасности в таких системах резко возрастают.

Другим примером классической технологии является применение терминальных серверов и рабочих станций. Казалось бы, все указанные выше недостатки преодолеваются при помощи терминальных технологий. Администраторы такой сети имеют полный контроль над ПО терминальных серверов, пользователи приобретают необходимые возможности по мобильности. Однако в полной мере эти преимущества реализуются только при использовании на рабочих местах пользователей специализированных терминальных систем, типа *Sun Ray*. В других же случаях мы опять не имеем возможности для полного контроля ПО, реализующего технологии тонкого клиента. А значит, становится возможной не только утечка информации, но и перехват удалённого управления тонким клиентом. Кроме того, при реализации технологии тонких клиентов проявляется ещё одно её слабое место: ограничения на возможности применения. В частности, возникают проблемы при использовании ПО, требующего значительных ресурсов центрального процессора, а также значительных ресурсов графической подсистемы, а то и другое практически всегда требуют современные CAD-системы. Возникают проблемы при работе с мультимедиа, поскольку передача видеoinформации от сервера к клиенту даёт значительную загрузку сети. Решение указанных проблем требует значительных финансовых затрат и существенно проигрывает по стоимости классическим решениям рабочих станций.

Идеальным решением вопросов безопасности и мобильности могла бы стать реализация технологии удалённой загрузки ОС, позволяющая администраторам полностью контролировать используемое на рабочем месте ПО. При этой технологии появляется возможность полностью использовать вычислительные ресурсы рабочей станции как для центрального процессора, так и для графической подсистемы. Ввиду полного контроля ПО можно реализовать меры защиты передаваемого по сети трафика и аутентификации доступа. Однако размещение всего программного обеспечения и данных пользователей на сетевых ресурсах значительно увеличивает сетевую нагрузку. Решением было бы применение распределённой файловой системы, позволяющей сохранять файловые ресурсы на жёсткие диски самой рабочей станции. Данные на таких дисках, конечно же, должны храниться в шиф-

рованном виде, а доступ к ним предоставляться только при помощи централизованно контролируемого ПО.

Удалённая загрузка операционной системы (ОС) рабочего места пользователя также даёт значительную загрузку сети, поэтому должно быть разделение ПО рабочего места на несколько слоёв:

1. Аппаратно зависимое ПО, а это собственно ядро ОС, сетевая подсистема и ПО графической подсистемы. При этом они могли бы устанавливаться на жёсткий диск рабочей станции.

2. Слой ПО, выполняющий инициализацию системы и контролирующий её функционирование. При этом он должен принадлежать сетевым ресурсам и централизованно контролироваться.

3. Слой прикладного ПО пользователя, предоставляемого ему с сетевых распределённых ресурсов. Он должен иметь возможность индивидуальной настройки для каждого пользователя, включая создание индивидуального набора ПО.

Таким образом, разделение ПО ОС на отдельные слои и их размещение на различных носителях могут реализовать задачу снижения сетевого трафика при загрузке системы и сохранении полного контроля над ней. В дополнение к этому нужно сказать, что сетевая архитектура системы должна сочетать в себе не только полный контроль исходного программного обеспечения, загружаемого на рабочее место пользователя, но и централизованное управление им в процессе работы, обеспечиваемый системами, аналогичными AD. Безопасность в данной технологии должна обеспечиваться применением современных систем аутентификации доступа и шифрования трафика к защищаемым ресурсам, удовлетворять российским требованиям в области защиты информации.

Вместе с тем, вычислительные возможности современных компьютеров позволяют полноценно использовать на рабочих местах средства виртуализации и не только повысить защищённость систем, но и сделать возможным перемещение вслед за пользователем между компьютерами сети программного обеспечения, запускаемого в виртуальной машине.

Реализация такой распределённой и полностью централизованно управляемой системы актуальна для крупных предприятий, насчитывающих более тысячи рабочих мест. Для таких предприятий внедрение подобной технологии даёт существенное снижение

трудозатрат на содержание системы и обеспечение её безопасности. Но вместе с тем это выдвигает дополнительные требования к надёжности и гибкости. Становится необходимым иметь инструментарий, позволяющий реализовать гибкое разделение прав доступа пользователей к ресурсам сети — должна быть реализована технология рабочих групп, объединяющих пользователей в разрезе их отличий в правах доступа к ресурсам сети, используемого программного обеспечения и т.д. Реализация этих требований возможна при помощи централизованной системы управления типа AD. Обеспечение надёжности системы складывается из нескольких составляющих — надёжности сетевого хранилища и надёжности функционирования самой сети, а значит должно предусматривать дублирование всех критичных элементов. Должна учитываться возможность потенциального подключения стороннего аппаратного обеспечения, и в целях безопасности необходим контроль аппаратного обеспечения пользователей, т.е. необходимо иметь механизмы верификации аппаратного обеспечения, которое может использовать распределённую информационную среду. Это можно сделать при помощи формирования некой цифровой подписи, характеризующей однозначно набор аппаратного обеспечения. А система в целом должна обеспечивать простое подключение разрешённого оборудования и делать невозможным подключение потенциально опасного или неизвестного аппаратного обеспечения.

В качестве ПО пользователя предлагается использовать специализированные модули, построенные на базе сжатой файловой системы *squashfs+lzma*, представляющей собой преобразованные *RPM*, *DEB* пакеты. Система строится таким образом, чтобы делать подключение модулей «на лету». В результате чего установка ПО отличается от традиционной тем, что достаточно просто подключить предварительно настроенное ПО по любому разрешённому сетевому протоколу и оно сразу готово к работе. Модули подгружаются путём монтирования сетевого ресурса с последующим монтированием на *loop* устройство. Поддержка ПО пользователей осуществляется через централизованное хранилище модулей, путём своевременного обновления и устранения ошибок. Сделанные изменения сразу становятся доступны всем пользователям системы. Достаточно просто переподключить модуль. Это даёт возможность полного контроля над тем ПО, которое доступно пользователю и делает систему очень гибкой. Базовая часть си-

стемы может быть построена на основе *livecd* технологии с применением комбинации файловых систем, доступных только для чтения (*squashfs*), и файловых систем, развёрнутых в ОЗУ (*aufs*, *tmpfs*, *unionfs* и т.д.). Поскольку ПО построено на основе специализированных модулей и доставляется пользователю по сети с интеграцией в файловую систему развёрнутую в ОЗУ, то ему доступны все аппаратные ресурсы компьютера пользователя, что даёт возможность выполнять ресурсоёмкие приложения. Применение сжатой высокоскоростной файловой системы *squashfs+lzma* обеспечивает прирост производительности файловых операций на 1ГГц процессоре с 1 Гб ОЗУ примерно 1,5-3 раза в сравнении с обычным режимом работы реального устройства хранения (жёсткие диски, флеш-память и другие). Применение этой технологии является обоснованным и обеспечивает высокую производительность. Вместе с тем, как говорилось ранее, основной задачей распределённого хранилища является возможность надёжного хранения больших объёмов информации, а также обеспечение безопасного доступа. При этом немаловажную роль занимает быстродействие хранилища. В системе предлагается использовать распределённую сетевую файловую систему *AFS* на базе *OpenAFS*. Она позволяет достичь тех требований, которые предъявляются к системе. Безопасность в этом случае обеспечивается поддержкой *Kerberos*. Кроме того, *OpenAFS* позволяет строить высокомасштабируемые решения, содержащие тысячи узлов. Ключевым моментом является возможность распределённого хранения информации, расположенной на серверах (как можно ближе к клиентскому приложению), т.е. хранение информации между серверами распределяется таким образом, что данные расположены физически ближе к клиентскому приложению, которое их использует. А в тех случаях, где ближайший сервер расположен далеко, информация кешируется клиентской частью *AFS* непосредственно на компьютере пользователя. Система *AFS* позволяет также осуществлять репликацию данных и множественное хранение одной и той же копии на различных серверах, что делает хранение надёжным и не требует резервного хранения отдельно от самой системы. Следовательно, хранение специализированных модулей и пользовательских данных в таком хранилище является эффективным, производительным и надёжным. Для использования *AFS* в системе подключаются модули серверной и клиентской части, компьютер пользователя получает возможность работы с

распределённым хранилищем и становится его частью. В целях обеспечения доступа к сетевому хранилищу, наряду с авторизацией посредством *Kerberos*, предлагается использовать каталог *LDAP* на базе *OpenLDAP*. Он содержит информацию об аппаратном обеспечении, пользователях и подразделениях, программном обеспечении (наборы специализированных модулей), которое доступно пользователю, а также о всех настройках. Для работы с данным каталогом создаётся специализированный административный центр управления на основе *Web*-ориентированного интерфейса, позволяющий управлять данными в каталоге, а значит управлять все системой, включая ПО пользователей, серверов и т.д.

Одним из недостатков системы может быть то, что программное обеспечение «ходит» в след за пользователем, перемещаясь на компьютер только на время выполнения, и выполняется в некой, как бы виртуальной, среде. Реализация таких принципов применения ПО не предполагается лицензионными соглашениями многих производителей, поэтому трудности лицензирования в таких системах приходится относить к недостаткам. К преимуществам и перспективам применения предложенной технологии можно отнести следующее. Как уже указывалось ранее, полноценная реализация информационного сообщества невозможна без обеспечения надлежащего уровня безопасности. Децентрализованные технологии, применяемые в настоящее время, достигли вершины своего развития и фактически себя исчерпали. Свидетельством этому является общий низкий уровень безопасности информационных систем предприятий. Не повысив уровень защищённости систем, применяемых на предприятиях, мы не сможем реализовать концепций информационного общества, а не реализовав их, не сможем качественно повысить производительность труда и уровень жизни. В предлагаемом решении это обеспечивается за счёт централизованного контроля ПО и безопасного хранения данных и доступа ко всем ресурсам сети вне зависимости от того, в каком месте происходит подключение пользовательского аппаратного обеспечения. Технологическое повышение защищённости систем всегда более простой путь реализации задач в сравнении с организационными мерами и т.п. Поэтому именно с данной технологией и подобными ей связано дальнейшее развитие всех информационных технологий.

БАЗЫ ДАННЫХ ПО СВОЙСТВАМ НАНОРАЗМЕРНЫХ ОБЪЕКТОВ: ПРАКТИКА ИСПОЛЬЗОВАНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.О. Еркимбаев, В.Ю. Зицерман, Г.А. Кобзев,

М.С. Трахтенгерц

ОИВТ РАН

г. Москва, Россия

E-mail: 1941@vz.ru

Аннотация

Описана информационная система по свойствам наноструктур и наноматериалов, включающая фактографическую и документографическую базы данных (БД). Система построена с использованием свободно-распространяемых продуктов: СУБД PostgreSQL, CDS/ISIS, Web-сервера Apache и языка PHP. Рассмотрены преимущества работы с нелицензируемыми средствами применительно к научным данным.

Ключевые слова

Наноструктуры, базы данных, СУБД, структура данных.

Предмет работы — информационные технологии, используемые в Объединённом институте высоких температур (ОИВТ РАН) для задач хранения, обработки и распространения данных по физико-химическим свойствам наноструктур и наноматериалов. Хотя подобных систем пока ещё нет в сфере научно-технической информации, сама по себе компьютеризация фонда справочных данных имеет давнюю историю. В ОИВТ РАН с 1973 года существует Теплофизический центр, специальный орган, занятый разработкой, наполнением и поддержкой БД по свойствам определённой группы веществ: неорганических и простейших органических соединений. Эксперты проводят обработку отечественных и зарубежных изданий, извлекая из них сведения, отвечающие выделенным тематическим границам, и вводят в БД формализованную информацию, включающую библиографию, численные данные, а в отдельных случаях, полнотекстовые документы. В ходе этой работы были освоены технологии, опирающиеся преимущественно на свободное программное обеспечение.

щественно на свободно-распространяемое программное обеспечение (ПО). Стратегия с ориентацией на *free soft* имела ряд серьёзных оснований, определяемых спецификой области и возникающих задач. Прежде всего, теряют актуальность многие из требований, предъявляемых к коммерческим продуктам в части производительности, скорости обработки запросов, многопользовательского режима и т.п. Место технологических требований, возникающих в бизнесе, управлении производством или инфраструктурой, занимают совершенно иные, характерные для научных данных объекты. Среди них: многообразие форматов (числа, графика, программные коды), плохо формализуемая структура, многомерность данных, согласование данных и закономерностей, наделение численных данных информацией по их неопределённости и ряд других. Особенности научных данных по свойствам и возникающие отсюда требования к выбору программных средств рассмотрены в работах [1-3]. Именно **свободное ПО**, созданное за пределами компьютерной индустрии, оказалось наиболее адекватным указанным требованиям. Важно при этом, что в ходе работ по созданию фонда данных разработчики получают возможность беспрепятственно (без специальных лицензий) устанавливать разработанные продукты на все компьютеры, вне зависимости от их местоположения. Возможна также их установка на компьютерах заказчиков (потребителей данных), которые получают возможность самостоятельно поддерживать и обновлять БД по интересующей их тематике.

Переход от традиционной тематики к новым задачам по свойствам наноструктур и наноматериалов потребовал определённой методической работы, расширяющей функциональные возможности используемых программных средств. В отличие от многочисленных БД по свойствам традиционных веществ и материалов здесь приходится в самой структуре данных отобразить специфику объектов: промежуточное положение между молекулой и макроскопическим веществом; сложность идентификации по комплексу параметров, условиям приготовления, влиянию среды и проч.; (3) зависимость номенклатуры свойств от вида объекта; (4) изменчивость структуры данных, проявляемую в различиях их объёма и типа, в появлении новых и/или утрате смысла прежних характеристик, в изменении правил идентификации и т.п. Ключевым требованием к таким БД является их способность поддерживать полуструктурированные данные (ПСД).

Была создана информационная система, включающая два относительно автономных продукта: **DATA_N** и **DOCS_N**. Основным элементом информационной системы является фактографическая БД **DATA_N**, включающая отобранные и рекомендованные численные данные по свойствам наноструктур. Наряду с таблицами численных данных, БД пригодна для хранения других типов данных: текстовых, графических, исполняемых файлов и др.

Вторая БД, **DOCS_N**, является документографической БД, содержащей описания первичных документов по свойствам наноструктур. Описание документа включает сведения о его локализации: на сервере пользователя или ОИВТ РАН, сети INTERNET, научных библиотеках страны. Если документ доступен в электронном виде, в описание включена гиперссылка с указанием адреса на сервере или в сети и именем файла. Ресурсное наполнение документографической БД включает совокупность логически связанных управляющих файлов, которые определяют структуру записей, состав полей БД, поисковых образов документов, форматов печати и т.п.

БД DATA_N создана с использованием объектно-реляционной СУБД PostgreSQL [4], языка PHP для программирования оболочки БД и размещена на Web-сервере Apache. Все три продукта (PostgreSQL, PHP, Apache) относятся к категории свободно-распространяемого ПО с открытым кодом.

Современная СУБД PostgreSQL ведет происхождение из проекта POSTGRES, который разрабатывался под руководством Майкла Стоунбрейкера, профессора Калифорнийского университета в Беркли и нашел широкое применение в научном сообществе: терабайтное хранилище астрономических данных (www.astronet.ru), проект ALADDIN (A Labelled Atomic Data Interface.), поддержка федеральных порталов Минобразования РФ. Проект имел целью преодолеть ограниченность реляционной модели, обеспечивая создание и управление сложными объектами. Уникальные свойства PostgreSQL позволяют сочетать традиционные модели с задачей хранения данных с «размытой» структурой. Отличительная особенность PostgreSQL - богатство типов данных: символьных, числовых (произвольной точности), «больших объектов» (графика, файлы, программные коды и проч.), возможность создания новых типов данных, в частности, композитных типов (объединяющих элементарные типы для представления сложных объектов) и ряд других. Технология обобщённого поискового дерева (Gener-

alized Search Trees for Date) позволяет эксперту по свойствам наноструктур, не владея сведениями по БД, создавать специализированные типы данных и обеспечить доступ к ним. В целом, функциональные возможности PostgreSQL оказались адекватны специфике данных по свойствам наноструктур, включая наличие блоков данных с внутренней иерархической структурой, эклектичность типов (числа, таблицы, файлы и проч.), вариации логической структуры. С практической точки зрения построение систем на основе PostgreSQL облегчается за счёт: (1) доступности кодов, документации и сведений по развитию системы; (2) размещения средств управления БД на Web-сервере при минимальных требованиях к программному обеспечению пользователя (архитектура «клиент-сервер»); (3) развитых средств визуального проектирования БД, что для пользователя исключает необходимость в специальных знаниях по компьютерной технологии.

PHP (Personal Home Page Tools). Язык программирования, созданный для генерирования HTML-страниц на Web-сервере и работы с БД [4, 6]. Входит в стандартный набор для создания Web-сайтов (Linux, Apache, MySQL, PHP (Python или Perl)). Группа разработчиков PHP состоит из множества людей, добровольно работающих над ядром и расширениями PHP и смежными проектами, такими как документация языка.

Apache HTTP-сервер. Один из самых популярных HTTP-серверов в сети [5, 6], на рынке Web-серверов занимает примерно 60%. Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Web-сервер Apache разработан и поддерживается открытым сообществом разработчиков под эгидой Apache Software Foundation и включён во многие коммерческие продукты, например, СУБД ORACLE. Является бесплатным, свободно распространяемым продуктом. Краткие характеристики Web-сервера Apache по пятибалльной шкале: надёжность (5), производительность (5), простота использования (3), техническая поддержка (4.5).

Документографическая БД DOCS_N построена с использованием **СУБД CDS/ISIS**, системы, разработанной, поддерживаемой и свободно распространяемой ЮНЕСКО [7, 8]. В отличие от реляционных СУБД, ISIS в своей основе имеет текстовый формат, адекватно представляющий структурированные не числовые данные в форме записей произвольной длины. Библиографическая информация и описания произвольных документов могут быть

представлены наилучшим образом именно в такой форме, что соответствует рекомендациям стандарта ISO-2709, предназначенного для обмена библиографической информацией на магнитных носителях. CDS/ISIS имеет механизм для назначения смысловых функций полей индивидуально для каждой БД. Он позволяет при необходимости вписать вновь создаваемую БД в существующую информационную структуру, назначив ей такие же поля. В то же время при разработке самостоятельной системы может быть выбран свой набор полей, оптимальный для решаемой задачи. В данной разработке был применён второй подход как пионерский для БД по свойствам наноструктур.

В отношении объёма хранимой информации CDS/ISIS имеет весьма слабые ограничения: число одновременно поддерживаемых БД не ограничено; максимальное количество записей в БД - 16 млн.; максимальный размер записи - 8000 символов; максимальное количество полей - 200. Система идеально приспособлена для создания различного рода каталогов и указателей. Возможно построение любого каталога и указателя как по всему массиву записей БД (или его части), так и по массиву записей, отображенных в результате тематического поиска. После ввода очередной порции записей система выходит в основное меню и актуализирует инверсный файл (генерируемый упорядоченный словарь ключевых слов), при помощи которого система имеет возможность создавать для каждой записи фактически неограниченное число терминов доступа.

Поиск в БД осуществляется по элементам, определяемым пользователем в таблице выбора полей, которая содержит инвертируемые поля и метод индексирования, используемый для каждого поля. Содержание основных полей в БД идёт в инверсный файл. При поиске информации запрос представляет собой перечень терминов, связанных логическими операторами. В системе существуют следующие (основные) операторы: «ИЛИ» (объединение), «И» (пересечение), «НЕ» (отрицание), оператор усечения терминов справа. Возможна комбинация прямого (по словарю) и последовательного поиска, а также полнотекстовый поиск по тексту (в том числе, и по полям информация из которых не поступает в словарь).

БД проста и удобна для исследователя. Реализованные в СУБД ISIS функции позволяют: вводить новые записи в имеющиеся БД; редактировать существующие записи; автоматически

строить и поддерживать файлы быстрого доступа (словарь и инвертированный файл) с целью уменьшения времени поиска; осуществлять поиск записей в режиме полнотекстового поиска или с использованием развитого поискового языка; выдавать записи или их части в соответствии с требованиями пользователя; печатать каталоги и/или указатели с глубиной сортировки до четырёх уровней; подключать к системе прикладные программы, используя интегрированные средства программирования СУБД (встроенный Паскаль).

При том, что ISIS формально предназначен для библиографических БД, он позволяет соединять каждую из записей (описаний документа) с файлами, включающими фактографическую информацию, которые размещаются либо на компьютере пользователя, либо в сети (локальной или ИНТЕРНЕТ). Малые габариты СУБД и весьма ограниченные требования к ресурсам компьютера позволяют распространять построенные на ней БД среди пользователей. Файлы, определяющие структуру БД, поставляются вместе с СУБД, а содержимое БД в виде файла описаний в формате ISO-2709, предназначенном для обмена данными. При наличии в БД фактографической информации дополнительно поставляется совокупность электронных документов в форматах *.pdf, *.html, *.mht или произвольных графических форматах.

Таким образом, на базе свободного ПО удалось построить распределённую систему, элементы которой размещены на сервере приложений ОИВТ РАН и компьютерах пользователей. Web-сервер Apache обеспечивает доступ клиентов через сеть INTERNET к БД DATA_N. Все её компоненты, включая СУБД POSTGRES и внешнюю оболочку для управления БД, размещены на сервере. Вторая БД (DOCS_N), содержащая описания документов, размещается на компьютерах пользователей. На файловом сервере размещают обновления БД в виде файла формата *.iso, специального формата для обмена структурированной текстовой информацией. Там же размещается репозиторий с электронными документами, адрес которых включен в описания документов в БД DOCS_N. Работа выполнена при поддержке Гранта РФФИ №10-08-00623а.

Используемые источники:

1. Зицерман В.Ю., Кобзев Г.А., Фокин Л.Р. Возможности и перспективы информационных технологий в подготовке и распространении справочных данных: свойства веществ и материалов. Научно-техническая информация. Серия 1. 2004. №2. С. 7.

2. Еркимбаев А.О., Зицерман В.Ю., Кобзев Г.А. Роль метаданных в создании и использовании информационных ресурсов о свойствах веществ и материалов. Научно-техническая информация. Серия 1. 2008. №11. С. 14.
3. Еркимбаев А.О., Зицерман В.Ю., Кобзев Г.А., Фокин Л.Р. Логическая структура физико-химических данных. Проблемы стандартизации и обмена численными данными. Журнал физической химии. 2008. Т.82. № 1. С. 20.
4. Шенинг Г.Ю., Гешвинде Э. Разработка Web-приложений на PHP и PostgreSQL. Руководство разработчика и администратора. М: «ДИАСОФТ», 2003.
5. Фролов А.В., Фролов Г.В. Практика применения PERL, PHP, Apache и MySQL для активных Web-сайтов. М.: «РУС. РЕДАКЦИЯ», 2002.
6. Скотт Хокинс. Администрирование Web-сервера Apache и руководство по электронной коммерции - Apache Web Server Administration and e-Commerce Handbook. М.: «Вильямс», 2001.
7. Трахтенгерц М.С. Новый эффективный инструмент для текстовых баз данных : CDS/ISIS for Windows. Научно-техническая информация. Серия 2. 2006. № 6.
8. Пакет прикладных программ CDS/ISIS/M, версия 2.3. Методические материалы и документация по пакетам прикладных программ. М.: МЦНТИ, Вып.70, 1991.

УДК 004.41/.42

ПРОБЛЕМЫ АДАПТАЦИИ ПОЛЬЗОВАТЕЛЕЙ К ТЕХНОЛОГИЯМ, ПРИМЕНЯЕМЫМ В ПРОЕКТЕ «СЕТЕВОЙ УНИВЕРСИТЕТ»

А.М. Дербень, М.М. Фисков, Е.А. Вейсов

ГОУ ВПО «Сибирский аэрокосмический университет

им. ак. М.Ф. Решетнёва»

г. Красноярск, Россия

E-mail: derben@sibsau.ru

Аннотация

Рассматриваются проблемы перехода пользователей ОС семейства MS Windows на СПО. Сформулированы основные критерии при выборе графической подсистемы. Предложена модель проектирования и создания

основных элементов «Рабочего стола» пользователей СПО.

Ключевые слова

Графическая подсистема, Gnome, темы Рабочего стола, 3D моделирование, Blender.

Работы выполнены при поддержке г/к №02.740.11.0621 «Информационно-вычислительное моделирование сложных систем».

При проектировании сложной сетевой инфраструктуры, такой как «Сетевой университет», неизбежно возникновение проблемы с перемещением не только сетевого профиля пользователя, но и программного обеспечения. При использовании проприетарных операционных систем (семейства MS Windows, напр.) лицензией, как правило, запрещается производить какие-либо манипуляции с системой. Иными словами, установка у пользователя фактически привязывается к конкретному компьютеру и, в лучшем случае, при переносе на другой компьютер предписывается уничтожить предыдущую копию ОС. Кроме того, используются разного рода механизмы защиты от такого «несанкционированного копирования», которые существенно затрудняют любую миграцию ПО. Существует и другой немаловажный фактор: разработчики проприетарного ПО стремятся использовать свои закрытые протоколы хранения и передачи данных. В случае, когда под давлением общественности и судебных исков производителей принуждают открыть свои спецификации, то они не полностью соответствуют реализованному функционалу (достаточно вспомнить ситуацию с Microsoft Office 2007: это пакет программ сохраняет файлы в специальном формате, отличающемся от спецификации OOXML). Актуальной остаётся угроза утечки информации: например, для ОС семейства Windows ежедневно появляется не менее 300 — 400 новых вредоносных программ, большинство из которых классифицируются как программы удалённого управления и взлома (т.н. «трояны»).

Все вышеперечисленные проблемы мешают прогрессу в построении сложных вычислительных систем. В последние годы появились высокопроизводительные рабочие станции с многоядерными процессорами и гигабайтами ОЗУ, высокоскоростные каналы передачи данных, быстрыми темпами развивается рынок мобильных устройств. Однако схема построения крупных сетевых подсистем остаётся неизменной с середины 90-х годов: канал до-

ступа в Интернет (или выделенный канал, VPN и т.п.) – шлюз, маршрутизатор, файерволл – прокси-сервер – конечный пользователь. Частично решить проблему создания мобильных рабочих групп можно при помощи технологии терминального доступа. Но при этом имеются другие ограничения: «тяжёлые» графические приложения для моделирования трёхмерных объектов, такие как AutoCAD или Компас-3D требуют, кроме серьёзного процессорного ресурса, также и ресурсов видеокарты. Очевидно, что такие приложения не смогут в полном объёме функционировать на терминальном сервере. Кроме того, ресурсы рабочих станций оказываются практически не задействованными во время терминальной сессии.

В связи с вышеописанными факторами, выглядит перспективной модель построения распределённой сетевой системы типа «облако», с файловой системой типа AFS, на базе СПО. Но у такой модели есть одна проблема, без решения которой не может быть успешного внедрения: большинство существующих пользователей персональных компьютеров имеют навыки работы только в ОС семейства MS Windows.

В настоящей статье ставится цель рассмотрения проблем субъективного характера: непривычной рабочей среды, неправильной настройки «Рабочего стола» пользователя.

Проблемы объективного характера также существуют: это и нереализованность некоторой необходимой части функционала программ, неполная эмуляция Windows – окружения, зачастую в работе СПО наблюдается некоторая потеря производительности. Процедуры решения таких задач на настоящий момент более или менее отработаны.

Но что делать, если пользователь постоянно жалуется на непривычную среду работы? Многие системные администраторы склонны недооценивать серьёзность таких жалоб, списывая все на односторонность мышления, низкую квалификацию, да и на элементарную лень пользователя. Но если просто понаблюдать за действиями таких пользователей хотя бы полчаса, то выясняется иная ситуация.

За годы работы у пользователя формируются определённые навыки. Например, кнопка закрытия окна всегда была красного цвета с белым крестиком посередине. Это удобно. И вот установили Linux, в нем графическая среда Gnome, а там — та же кнопка (просто серая) ничем не выделяется. В реальной работе пользова-

тель начинает просто «промахиваться» мимо этой кнопки. И таких примеров достаточно много.

Первоначально возникло предположение, что достаточно просто создать тему Рабочего стола с пиктограммами, напоминающими пиктограммы, используемые в ОС MS Windows XP. Но такой эксперимент не привёл к каким-либо заметным результатам. И тогда было принято решение провести более масштабное исследование. Оно было начато в начале 2007 г. и продолжается по настоящий момент.

На первом этапе была собрана статистика активности использования ПО с 2-х работающих терминальных серверов под управлением MS Windows 2003 Server Std, которые обслуживали сеть работников бухгалтерии, планово-финансового управления, отдела кадров университета (всего порядка 48 пользователей). Эксперимент длился более 2-х месяцев. Результат эксперимента показал, что в 78-86% случаев использовались узкоспециализированные АРМ'ы, такие как «1С», «Налогоплательщик ЮЛ», «ДокПУ», «2НДФЛ» и т.д. При этом пиктограммы запуска этих приложений, как правило, сразу выводятся на «Рабочий стол» и из главного меню (кн. «Пуск») не вызываются. Текстовый редактор Word и приложение для работы с электронными таблицами Excell из пакета программ MS Office 2003 использовались крайне редко, в основном для проверки и коррекции полученных из основных АРМ'ов документов. При замене MS Office 2003 на OpenOffice никаких существенных изменений в работе пользователей не произошло. Для работы с архивами программа WinRAR, как правило, не вызывалась из главного меню, распаковка архивов осуществлялась либо простым кликом мышки на архив, либо вызовом контекстного меню правой кнопкой мышки. В настройках «Проводника» по умолчанию в ОС MS Windows 2003 расширения файлов не показываются, поэтому пользователи различали тип файла только по пиктограмме. Главные выводы из данного эксперимента:

1. Первостепенное внимание при адаптации пользователя следует уделить пиктограммам АРМ'ов и иных узкоспециальных программ.

2. Пиктограммы часто используемых типов документов и файлов должны быть максимально похожими на пиктограммы в ОС семейства MS Windows, содержать логотипы производителей ПО, которые используются при обработке данного типа файла, либо должны позволять каким-либо иным способом идентифици-

ровать тип файла. Так, например, в теме по умолчанию в графической подсистеме Gnome типы файлов *.doc и *.odt визуально никак не различаются, что часто приводило к ошибкам.

3. Рамка, оформление кнопок оконного интерфейса, панель задач и кнопка главного меню должны быть максимально похожими на аналогичные элементы оформления «Рабочего стола» в ОС семейства MS Windows.

На втором этапе был произведён выбор графической подсистемы. Графическая подсистема должна удовлетворять следующим условиям:

1. Внедрение новых стилей и тем для рабочего стола должно производиться без программирования или изменения функционала.

2. Концепция и идеология пользовательского интерфейса должны быть максимально близки к концепции и идеологии ОС семейства MS Windows.

3. Установка и конфигурация рабочего стола должны производиться без вмешательства системного администратора, иметь удобный и интуитивно понятный интерфейс.

Из всего многообразия графических подсистем за основу дальнейших экспериментов была выбрана Gnome, поскольку она удовлетворяла всем вышеизложенным требованиям.

После этого было произведено 2 пробные темы для рабочего стола, стилизованные под пиктограммы и оформление Windows XP и Windows Vista. После нескольких месяцев пробной эксплуатации методом опроса порядка 50 пользователей был выявлен ряд иных существенных деталей:

1. Фоновое изображение с логотипом производителя (применялся дистрибутив MandrivaLinux 2007.1 — 2008.1) воспринимается большинством пользователей негативно. Положительно оценивается любое изображение с преобладанием сине-зелёного фона без отдельных объектов, по размеру сравнимых с размерами пиктограмм. При дальнейших экспериментах выяснилась следующая зависимость: цвета должны быть дополняющими по отношению друг к другу, т.е. при их смешивании должна появляться иллюзия белого или грязно-серого цвета.

2. Пиктограммы для работы с системными функциями ОС: Не должны полностью копироваться аналогичные пиктограммы из оформления ОС семейства Windows, но должны быть интуи-

тивно понятными, также они не должны привлекать излишнего внимания пользователя, иметь нейтральные тона.

Для сокращения времени создания отдельных элементов темы была разработана следующая технология:

1. В 3-D редакторе «Blender» создаётся трёхмерная модель фонового изображения (как правило, это изображение в стиле «Безмятежность» в Windows XP).

2. Создаются трёхмерные модели значков «Мой Компьютер», папок, клавиатуры, мыши и др. элементов, которые в дальнейшем будут использоваться для создания отдельных пиктограмм.

3. Все созданные элементы компоуются в одной 3-D сцене, подбираются наилучшее освещение, размер, угол зрения.

4. Создаются вспомогательные элементы: кнопки, панели, пиктограммы типов файлов и запуска приложений.

5. 3-D изображения конвертируются в тему для «Рабочего стола».

Созданные по такой технологии графические темы по сравнению с традиционными имеют ряд преимуществ:

- легко модифицируются, видоизменяются такие параметры пиктограмм и фоновых изображений, как настройка яркости, контрастности, цветности, бликов, теней, угла зрения;
- существует возможность полуавтоматической генерации тем для Рабочего стола на языке программирования «Python», используемого в редакторе «Blender»;
- достигается максимальная реалистичность как для отдельных элементов Рабочего стола, так и для темы в целом;
- появляется возможность проектировать концептуально новый, трёхмерный вид графического интерфейса.

В результате внедрения новых тем для Рабочего стола были достигнуты следующие результаты:

- снижение времени адаптации пользователя ОС семейства MS Windows при переходе на СПО до 3-5 дней;
- снижение ошибочных действий пользователя, связанных с низкой эргономикой СПО;
- получение принципиально новой модели проектирования и создания пользовательских интерфейсов для СПО.

Направления для дальнейших исследований:

1. Исследование критериев оптимизации тем «Рабочего стола» для отдельных групп пользователей (таких как пол, возраст, др. социальные признаки).

2. Создание экспертных систем для автоматизированного создания графических тем для Рабочего стола по признакам, изложенным в п.1.

3. Применение систем распознавания образов с целью получения наилучшей эргономики пользовательского интерфейса. Системы распознавания образов могут быть использованы при автоматической регулировке контрастности темы. Например, в течение рабочего дня у пользователя повышается утомляемость, вследствие чего увеличиваются ошибки, проявляемые в работе с элементами рабочего стола, далее – система распознает ошибки и при превышении определённого порога адаптирует тему «Рабочего стола» для обеспечения наилучшей производительности. Также системы распознавания образов для определения оптимальных параметров для пользователя по фото или непосредственно через Web-камеру (через измерение отдельных пропорций лица определяются признаки, изложенные в п.1).

УДК 004.9

ПОСТРОЕНИЕ СИСТЕМ ОБЛАЧНЫХ ВЫЧИСЛЕНИЙ НА ОСНОВЕ ОТКРЫТЫХ ТЕХНОЛОГИЙ

А.О. Шалеев

«Siemens IT Solutions and Services»

г. Воронеж, Россия

E-mail: anton.shaleev@siemens.com

Аннотация

В статье рассматриваются основы построения эластичных облачных архитектур (Elastic Clouds), подходы к созданию IaaS, PaaS и SaaS решений на основе проектов с открытым исходным кодом. Дается текущий обзор состояния и перспектив развития и внедрения.

Ключевые слова

Открытый код, облачные вычисления, эластичная облачная архитектура.

Развитие IT-инфраструктуры сегодня подошло к критическому порогу. Информационные технологии меняются очень быстро, и сегодня они образуют своего рода невидимый слой, который все ощутимее влияет на все аспекты нашей жизни. Электроэнергети-

ческие сети, управление дорожным движением, здравоохранение, водоснабжение, пищевая промышленность, а также большая часть финансовых транзакций в современном мире – все это сегодня зависит от информационных технологий. Новая модель оказания ИТ-услуг под названием cloud computing («облачные вычисления») способна значительно уменьшить ИТ-расходы и сложность инфраструктур, а также оптимизировать рабочие нагрузки и функционирование ИТ-сервисов. Cloud-решения обладают высокой степенью масштабируемости, обеспечивают пользователям широчайшие возможности и создают основу для построения новой, Интернет-ориентированной экономики.

Уже сейчас можно найти множество Cloud-решений, предлагаемых гигантами ИТ-индустрии, но высокая стоимость на данные решения определяет своеобразный порог вхождения, который подчас высок даже для среднего бизнеса. К счастью, уже сейчас доступны технологии построения Cloud-решений на основе открытого исходного кода всех трёх основных концепций: IaaS, PaaS и SaaS.

Прежде, чем приступить к непосредственному описанию программных продуктов, введём определения трёх основных концепций Cloud-решений:

- IaaS (Infrastructure as a Service) — инфраструктура как сервис — это предоставление компьютерной инфраструктуры (как правило, в форме виртуализации) в качестве услуги. Данная концепция состоит из трёх основных компонентов:

- 1) аппаратные средства (серверы, системы хранения данных, клиентские системы, сетевое оборудование);

- 2) операционные системы и системное ПО (средства виртуализации, автоматизации, основные средства управления ресурсами);

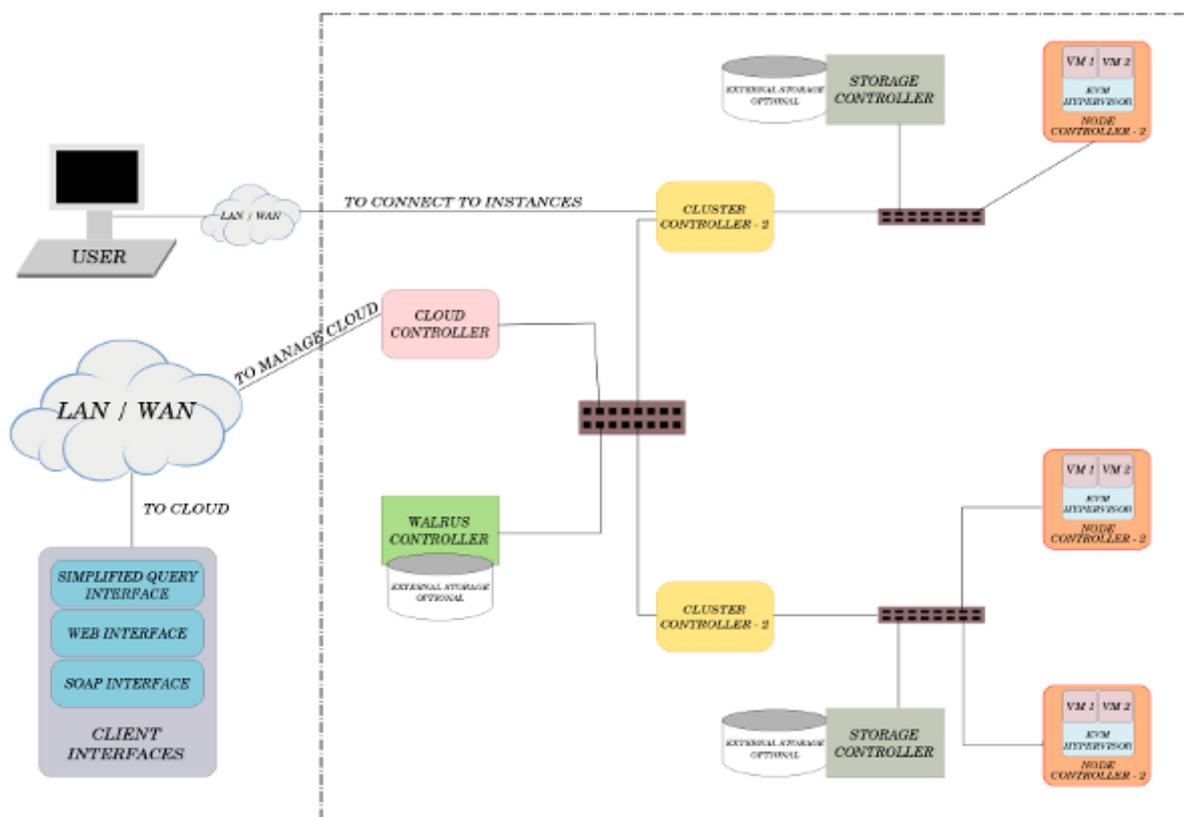
- 3) связующее ПО (контроллер кластера).

- PaaS (Platform as a Service) — платформа как сервис — предоставление интегрированной платформы для разработки, тестирования, развёртывания и поддержки Web-приложений как услуги.

- SaaS (Software as a Service) — программное обеспечение как сервис — бизнес-модель продажи программного обеспечения, при которой поставщик разрабатывает Web-приложение и самостоятельно управляет им, предоставляя заказчикам доступ к программному обеспечению через Интернет.

Наиболее ярким представителем открытых IaaS-решений является Eucalyptus. Разработанное в университете Калифорнии приложение первоначально предназначалось для запуска только на одном типе кластеров, теперь же установочный комплект доступен для всех популярных дистрибутивов Linux. Более того, на основе Eucalyptus такие компании, как Canonical, предлагают OpenSource решения корпоративного уровня.

Eucalyptus представляет собой свободную реализацию облачной инфраструктуры, полностью совместимую с известным сервисом Amazon EC2.



EUCALYPTUS BASED CLOUD

Рис.1. Модель облачной инфраструктуры

На рис.1 можно ознакомиться с моделью облачной инфраструктуры. Это:

- 1) контроллер облака;
- 2) контроллер кластера;
- 3) контроллеры нод;
- 4) Walrus Storage контроллер.

В качестве среды для тестирования Eucalyptus достаточно всего трёх машин (рис. 2).

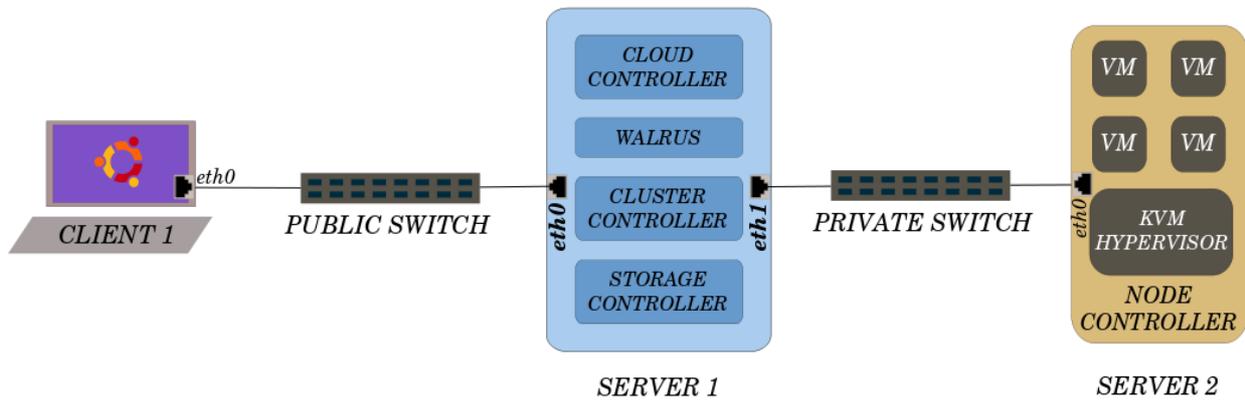


Рис. 2. Среда тестирования Eucalyptus

1. Клиент (машина с любой операционной системой, поддерживающий протокол RDP).

2. Сервер-контроллер облака (серверная машина с аппаратной поддержкой виртуализации), совмещающий в себе функции:

- Cloud Controller

CLC является интерфейсом для всего облака. На нём располагается Web-сервер с панелью администрирования и утилиты для взаимодействия с облаком.

- Walrus Storage Controller

WSC является системой хранения данных с использованием REST и SOAP API. Совместим Amazon S3 сервисом. Предполагается как простое файловое хранилище.

- Cluster Controller

CC осуществляет управление одним или более контроллерами нод, управляет размещением и работой инстансов, а также сетью в одном из доступных режимов.

- Storage Controller

SC представляет собой хранилище для перманентных данных инстансов. Аналогичен Amazon EBS.

3. Контроллер нод (серверная машина с аппаратной поддержкой виртуализации). NC (Node Controller) устанавливается совместно с KVM-гипервизором и служит для размещения инстансов операционных систем.

Eucalyptus позволяет не только реализовать масштабируемую среду, но и наиболее эффективно использовать ресурсы благодаря мощнейшему механизму экономии энергопотребления: нода переходит в спящий режим после 5 минут простоя.

Другим ярким представителем Cloud-технологий является проект Apache Nadoop. Это свободная реализация концепции PaaS, где основным является наличие универсального API. Nadoop

изначально разрабатывался для поискового механизма Nutch, но сейчас это полноценный инструмент для распределённых вычислений, независимый от операционной системы.

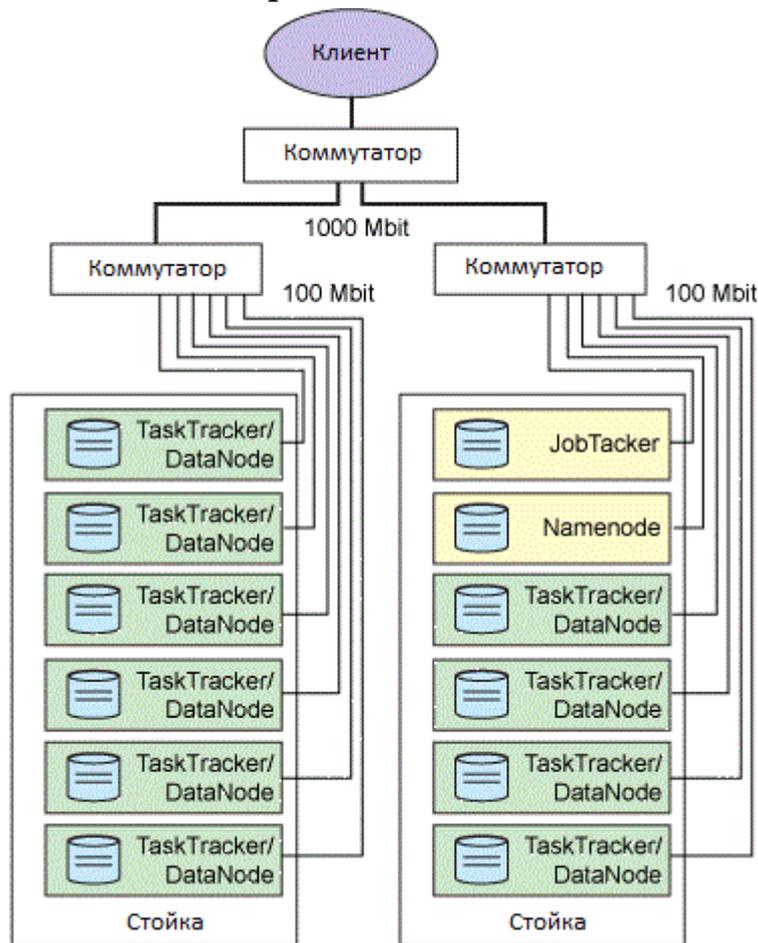


Рис. 3. Распределённая работа в кластере

Как эта функциональность реализована в Hadoop? Приложение MapReduce запускается по требованию клиента на единственном управляющем узле, который называется JobTracker. Как и узел NameNode, это единственный узел данного типа в кластере Hadoop, а его задачей является управление приложениями MapReduce. Когда приложение запущено, ему предоставляются входные и выходные директории, содержащиеся в файловой системе HDFS. Узел JobTracker использует информацию о файловых блоках (количество блоков и их месторасположение), чтобы решить, сколько подчиненных задач необходимо создать на узлах типа TaskTracker. Приложение MapReduce копируется на каждый узел, содержащий входные файловые блоки. Для каждого файлового блока заданного узла создается отдельная подчиненная задача. Каждый узел TaskTracker докладывает о статусе работы и о завершении задачи узлу JobTracker. На рисунке 3 показан пример распределенной работы в кластере.

Разнообразие OpenSource представителей концепции SaaS настолько велико, что трудно выделить явного лидера. Реализации данной концепции очень сильно разбиты по сферам применения. Среди SaaS решений в сфере CRM можно выделить Compiere, а в среде облачных операционных систем несомненным лидером является ChromiumOS, которая делает возможным осуществление всех операций пользователя через унифицированные механизмы облака, из какой точки мира он бы не зашёл в систему. ChromiumOS также представляет огромный простор для разработчиков, делая доступным работу с облачными технологиями в среде настольных компьютеров: начиная от Flash-игр, заканчивая удалённой средой разработки, позволяющей работать с кодом из любой точки мира.

Облачные технологии на сегодня являются наиболее перспективным направлением развития систем, опережая в эффективности как старые терминальные, так и десктопные решения.

УДК 004.9

СЕРВЕР МАЛОГО ПРЕДПРИЯТИЯ НА ОСНОВЕ LINUX

Р.А. Голубев

ООО «Сова-софт»

г. Липецк, Россия

E-mail: yazichnic@yandex.ru

Аннотация

В статье представлен один из вариантов лицензирования сервера предприятия на основе свободного программного обеспечения.

Ключевые слова

Лицензирование, сервер предприятия, удалённое администрирование.

Давно известно, что стоимость программного обеспечения составляет значительные суммы и зачастую превосходит стоимость самого компьютера. Стоимость минимального набора программного обеспечения для офиса составляет сумму, сопоставимую со стоимостью офисного компьютера. В то же время ничего сложного данное программное обеспечение не производит. Поэтому создаётся ситуация, при которой необходимо максимально ми-

минимизировать затраты на программное обеспечение. Выходов из этой ситуации принципиально существует два: использование пиратского программного обеспечения (коммерческих продуктов без оплаты лицензий на их использование), использование свободных программных продуктов. В соответствии с Гражданским кодексом за использование пиратского программного обеспечения предусмотрена ответственность вплоть до уголовной.

Последнее время часто приходится решать задачи лицензирования программного обеспечения, в том числе и серверов предприятий. Среднестатистический сервер в небольшой организации используется для решения нескольких задач:

- файловый сервер;
- шлюз в интернет;
- сервер приложений.

Рассмотрим подобный сервер, организованный на базе свободных и проприетарных программных продуктов. В деятельности офисов часто используются такие программные продукты, как пакеты офисных программ, программы бухгалтерского учёта, информационно-правовые системы.

Первое, что необходимо обеспечить, — это возможность удалённого администрирования в Unix системах. Это реализуется довольно просто. Для этого мы используем консольный доступ через SSH и Web-конфигуратор Webmin.

SSH позволяет безопасно передавать в незащищённой среде практически любой другой сетевой протокол, таким образом, можно не только удалённо работать на компьютере через командную оболочку, но и передавать по зашифрованному каналу звуковой поток или видео. Также SSH может использовать сжатие передаваемых данных для последующего их шифрования, что удобно, например, для удалённого запуска клиентов X Window System.

Webmin — это программный комплекс, позволяющий администрировать операционную систему через Web-интерфейс, в большинстве случаев позволяя обойтись без использования командной строки и запоминания системных команд и их параметров.

Webmin обычно не входит в состав стандартных репозиториях, но сборки для большинства популярных платформ есть на сайте программы.

Для организации обмена файлами часто используют общие папки на компьютерах. При использовании выделенного сервера

было бы логично использовать часть его дискового пространства для организации файлового сервера. Для этого стоит рассмотреть возможность использования файлового сервера Samba.

Samba использует один конфигурационный файл, в котором прописываются общие папки и права доступа к ним, также Samba позволяет организовать контроллер домена и общий доступ к принтерам. В связи с этим стоит назвать некоторые сложности с принтерами. Существуют так называемые win-принтеры, которые используют исключительно ресурсы компьютера и драйвера для печати, т.е. сам принтер выполняет только непосредственно печать, а формирование задания печати происходит на компьютере пользователя. Пример такого принтера — Canon MF5770. К сожалению, печать на таких принтерах невозможно реализовать без использования Windows.

С принтерами, для которых существуют проприетарные драйвера для альтернативных Windows операционных систем или которые поддерживают Post Script, проблем с печатью обычно не возникает. Для реализации печати на win-принтере необходимо сделать ряд действий, связанных с созданием на windows-машине виртуального принтера, задания с которого будут через специальные обработчики передаваться на реальный принтер. Такая машина может быть как физической, так и виртуальной, но в любом случае понадобится доступ в интернет, чтобы получить необходимое программное обеспечение.

На данный момент услуги предоставления доступа в сеть internet для организаций достаточно дорогие и безлимитные тарифы не всегда доступны для организаций, в связи с этим возникает необходимость организовать контроль доступа и учёт потребляемого трафика.

Учёт расходования трафика пользователями организации является важным моментом, так как позволяет контролировать сетевую активность пользователей в рабочее время и ограничивать её необходимыми рабочими контактами. Это особенно важно при отсутствии безлимитного доступа в сеть. Изначально рассматривалась возможность использования биллинговых систем, но позже отказались в пользу прокси-сервера SQUID и оболочки для его администрирования SAMS.

SAMS представляет собой программное средство для администрирования доступа пользователей к прокси-серверу SQUID.

Возможности системы:

- администрирование системы через Web-интерфейс;
- ограничение объёма трафика пользователей на месяц;
- автоматическое отключение пользователей, превысивших лимит;
- блокировка доступа пользователей к запрещённым ресурсам интернет;
- настройка доступа пользователей через механизм шаблонов;
- разграничение пользователей на группы для удобства администрирования системы;
- ведение статистики посещённых пользователями ресурсов интернет;
- формирование отчётов по трафику пользователей за любой отрезок времени;
- поддержка видов авторизации SQUID: ntlm, ncsa, ip;
- поддержка использования редиректоров SQUID: rejik, squidGuard;
- ограничение скорости загрузки для групп пользователей;
- посылка сообщений администратору при отключении пользователей при превышении трафика;
- для хранения данных используется СУБД MySQL.

При отсутствии контроллера домена наиболее удобным способом авторизации на проксе-сервере является использование авторизации по ip компьютера-клиента. Таким образом, использование SAMS+SQUID является наиболее простым способом реализации контроля за трафиком и сетевой активностью пользователей в небольшой организации.

Сервер предназначался для использования в офисе, а соответственно возникла необходимость организации работы офисных приложений, в частности 1С:Бухгалтерия 7.7 и правовая система КонсультантПлюс.

Наиболее сложным этапом, на наш взгляд, является настройка сервера терминалов и приложений, работающих в нем. Это связано с несколькими моментами:

- выбор и настройка непосредственно сервера терминалов;
- установка клиентских приложений;
- настройка пользовательского окружения.

Организовать сервер терминалов можно различными средствами: VNC, FreeNX, XRDP. В результате анализа было решено использовать FreeNX в редакции от Etersoft. Главной причиной

данного выбора стало наличие клиента как для Linux, так и для Windows, а также подробное руководство на русском языке от разработчика. FreeNX использует для передачи данных канал, создаваемый с помощью SSH, поэтому можно организовать терминальный доступ как по локальной сети, так и через Internet. Для организации работы windows-приложений, к сожалению, вариантов не много и наиболее работоспособный из них — это использование модифицированного wine от компании Etersoft. Этот продукт состоит из открытой и закрытой части, закрытая — проприетарная часть, за которую и предлагается заплатить, содержит библиотеки, необходимые для успешного запуска таких приложений, как: 1С:Предприятие 7.7, Гарант, ГрандСмета, Консультант-Плюс. При этом в отличие от большинства других проприетарных продуктов срок техподдержки ограничен по времени и продлевается по подписке. Здесь используется модель получения прибыли на основе Open Source, но в то же время не даётся полной свободы модификации и дальнейшего использования кода в других продуктах.

Хотелось бы отметить, что эмуляция Windows-окружения не идеальна и возникают различные проблемы, например, падение базы 1С при работе терминальных и нативных клиентов под Windows по сети с одной и той же базой одновременно, — приходилось каждый раз переиндексировать базы. Etersoft предложил два варианта решения проблемы: завернуть терминалы по cifs на файловый сервер или перевести всех клиентов в режим терминала.

ОПЫТ ИСПОЛЬЗОВАНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СФЕРЕ СРЕДНЕГО ОБРАЗОВАНИЯ

УДК 004.416.6

ПАКЕТ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ШКОЛ. РАЗРАБОТКА И ПЕРСПЕКТИВЫ ВНЕДРЕНИЯ

В.Л. Черный

ООО «Альт Линукс»

г. Москва, Россия

E-mail: black@altlinux.ru

Аннотация

В работе излагается краткая история проекта ПСПО. Текущее состояние внедрения в ОУ РФ и его перспективы. Отдельное место занимает обзор нового комплекта ПСПО — Альт Линукс 5.0 школьный.

Ключевые слова

Пакет свободного программного обеспечения, план реализации Стратегии развития информационного общества в Российской Федерации до 2011 года.

Краткая история

В 2007 году был объявлен конкурс на разработку и внедрение пакета свободного программного обеспечения (ПСПО) для школ. Опорный документ — приказ Председателя Правительства (Зубков) №1447-р от 18 октября 2007 г.

Конкурс выиграла ГК Армада. ALT Linux — соисполнитель.

2007-2008 — разработка и пилотная апробация в трёх регионах: республика Татарстан, Пермский край и Томская область. Всего охватывала около 1100 ОУ. В процессе было разрешение ФАО на добровольное присоединение к проекту, и в итоге апробацией занималось более 2300 ОУ.

2009 – обучение учителей и распространение ПСПО выполняли АйТи и IBS — качество и того и другого оставляет желать лучшего. Оба исполнителя были оштрафованы, а ФАО за допущенные нарушения ликвидировано.

2010-2011 – должно пройти внедрение, поскольку с 01.01.2011 лицензии, полученные с пакетом Первая Помощь, прекращают своё действие. К сожалению, в первом полугодии 2010 года внедрение выродилось в принудительную установку «опытных зон».

И все -таки ближайшая цель, определённая президентом РФ, именно внедрение!

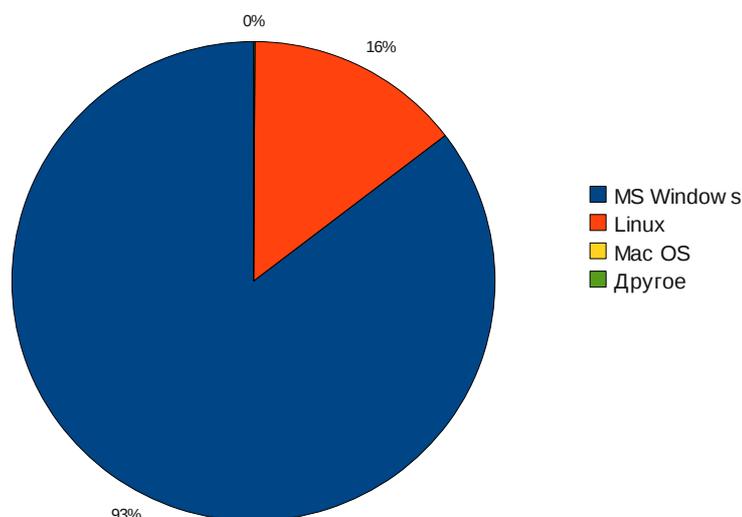
Президент России Д.Медведев утвердил План реализации Стратегии развития информационного общества в Российской Федерации до 2011 года.

В плане — «обеспечение перехода общеобразовательных учреждений на использование разработанного пакета свободного программного обеспечения (ПСПО)» (стр.13 Плана, п/п 2.3). В качестве результата работы по этому направлению к концу 2010 года 25% общеобразовательных учреждений каждого региона РФ должны использовать ПСПО не менее, чем на 50% имеющихся персональных компьютеров.

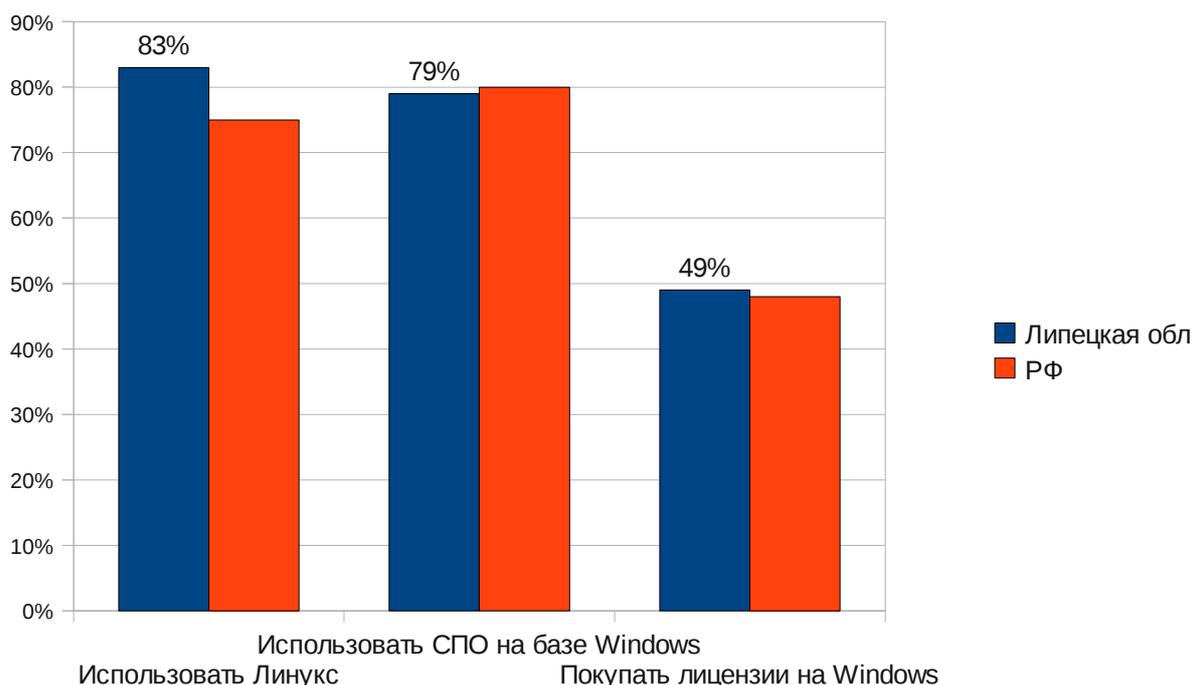
К концу 2011 года доля таких школ должна превысить 35%. Ответственными за достижение данных показателей назначены Минобрнауки России и органы исполнительной власти субъектов РФ. (<http://www.infosovet.ru/files/20100213%20Pr-357.pdf>)

Положение дел

ОС используемые в Липецкой обл.



Ближайшие планы



Основные применения СПО и проблемы

Административное – проблема обмена информацией с УО

Преподавание информатики – возможностей намного больше, чем раньше ЦОР, ЭОР — движок под Линукс сделан, но работает крайне плохо.

Новая версия ПСПО – Альт Линукс 5.0 Школьный

Состав:

Диски с ПО и методическими материалами

Книга «Руководство пользователя»

Лицензионное соглашение

Голографические наклейки

Плакат для оформления класса

Купон технической поддержки

Альт Линукс 5.0 Школьный

Диск с методическими материалами

Состав:

Книги по ОС Линукс

Лекционный материал и планы курсов, включая методические рекомендации

Учебные пособия по прикладным программам

Работы победителей конкурса «КОД СВОБОДЫ»

Документация по дистрибутивам комплекта

Встроенный сервер с moin-moin wiki для раздачи материалов сразу на весь класс

Книги из серии Библиотека Альт Линукс

Альт Линукс 5.0 Школьный

Диск с СПО под Windows

Состав:

Firefox – Web-браузер

Thunderbird – почтовый клиент

Lazarus – среда для программирования на Pascal

OpenOffice.org – офисный пакет

Audacity – редактор звуковых файлов

Blender – редактор 3D-графики

Scilab – пакет для инженерных расчётов

Scratch – среда для обучения программированию

Pidgin – клиент мгновенных сообщений

и многое другое...

Альт Линукс 5.0 Школьный или MS Windows?

	Проприетарное ПО MS Windows	ПСПО 5.0
Доступность	\$8-14 на один компьютер.	БЕСПЛАТНО
Условия распространения	Только для всех ОУ региона.	Сколько нужно
Техническая поддержка, плакат, наклейки и проч.	нет	Есть (3500 р для ОУ)

Альт Линукс 5.0 Школьный. Структура технической поддержки

Техническая поддержка строится на основе распределённой сети сертифицированных партнёров ALT Linux. Партнёры отвечают на вопросы по телефону и e-mail. Партнёры отвечают за 1 и второй уровни технической поддержки: поддержку пользователей (ФАК) и администраторов (вопросы настройки). ALT Linux контролирует весь процесс и отвечает за 3-й уровень поддержки — поддержку разработчиков (пересборка пакетов, исправление ошибок, добавление возможностей программ).

У каждого партнёра есть в штате, как минимум, один сертифицированный специалист ALT Linux, что обеспечивает качество услуг, предоставляемых ими конечным пользователям.

Обучением пользователей занимаются наши авторизованные Учебные центры. В Липецкой области такой центр тоже есть — это Елецкий государственный университет.

Кроме того, совместно с ФГУ ГНИИ «Информика» мы организовали бесплатную сертификацию специалистов по внедрению ПСПО в систему образования. Таким специалистом может стать любой человек, который помогает процессу внедрения ПСПО в школах. Подробности на сайте:

<http://www.altlinux.ru/partners/specialist-edu/>

НА СЕГОДНЯШНИЙ ДЕНЬ (ТРИ НЕДЕЛИ С МОМЕНТА ОБЪЯВЛЕНИЯ) УЖЕ ПОДАНО СВЫШЕ 60 ЗАЯВОК И ВЫДАНО 9 СЕРТИФИКАТОВ.

УДК 004.9

ИСПОЛЬЗОВАНИЕ JASENU CMS ДЛЯ РАЗРАБОТКИ САЙТОВ СРЕДНИМИ УЧЕБНЫМИ ЗАВЕДЕНИЯМИ

А.В. Логинов, С.А. Лиликович

Луганский национальный университет имени Тараса Шевченко
г. Луганск, Украина

E-mail: Livich@yandex.ru

Аннотация

В статье рассмотрены вопросы использования лицензионно чистого программного обеспечения в средних учебных заведениях, особое внимание уделено вопросу разработки сайтов с помощью CMS, освещён вопрос создания сайта с помощью авторской разработки Jasenu CMS.

Ключевые слова

Программное обеспечение, CMS, сайт, школа, контент.

Использование пиратских компьютерных программ и баз данных в Украине давно стало общенациональной проблемой [1]. Общий уровень пиратского программного обеспечения как в Украине, так и в России, остаётся крайне высоким – около 80% [2-5].

Говорить, что эта проблема касается только наших стран, было бы грубой ошибкой. К примеру, Георг Херрнлебен, директор BSA (Business Software Alliance) в Восточной и Центральной Европе, отмечает [2], что уровень пиратства в США в 1988 году был примерно на уровне 50%, в настоящее время – 20%. В то же время мало какие страны могут конкурировать с Украиной и Россией по масштабам распространения компьютерного пиратства. Особенно актуальным является вопрос использования лицензионного программного обеспечения в учебных заведениях. «Дело Поносова» в России является достаточно показательным примером остроты данной проблемы. С вступлением Украины в ВТО и стремлением России вступить в эту организацию (достаточно жёстко контролирующую вопрос защиты прав интеллектуальной собственности) актуальность использования лицензионного программного обеспечения, ПО с открытым кодом будет только возрастать.

Вопрос использования лицензионного программного обеспечения в образовательных учреждениях в настоящее время освещается достаточно широко. Но большинство работ на эту тему посвящено использованию операционной системы Linux, анализу наиболее распространённого ПО под эту ОС (офисные пакеты, языки программирования, антивирусные программы). Проблема использования узкоспециализированных программ, к примеру бесплатных CMS, освещена не достаточно.

Повышение количества сайтов учебных заведений (прежде всего, школ, лицеев, гимназий) является устойчивой тенденцией последних лет. При этом уровень знаний создателей сайтов в области Web-дизайна и Web-программирования остаётся не очень высоким. Эти причины являются одним из объяснений широкого использования CMS при создании сайтов учебными заведениями. Значительная часть школ разместила свои сайты на платформе ucoz.ru. Однако эта платформа обладает большими недостатками. Среди них не только невозможность использовать PHP, Perl и ASP, но и реклама, само наличие которой для учебного заведения часто является неприемлемым. Высокий процент сайтов построен с использованием иных CMS. При этом стоит отметить, что большинство популярных CMS являются коммерческими. Очень мало учебных заведений, в которых установлены лицензионные версии операционных систем и офисных пакетов, готовых платить достаточно большие деньги за покупку CMS. Выходом из данной ситуации является использование бесплатных CMS. При-

мером свободно распространяемой системы управления контентом сайта является Jasenu CMS. Данная система разработана слушателем Малой академии наук Украины при Ровеньковском факультете Луганского национального университета имени Тараса Шевченко Лиликовичем Сергеем. Разработанная система была представлена на Всеукраинском чемпионате по информационным технологиям «Экософт-2009» и победила в номинациях «Лучший функциональный продукт» и «Высокий профессиональный уровень программного продукта» в категории «Web-разработка».

Основными особенностями разработанной системы являются:

- кроссплатформенность, возможность работы под Windows и Linux;

- свободное распространение (freeware);

- простота, удобство работы пользователя;

- возможность создания как простых сайтов, так и сложных корпоративных сайтов, к которым предъявляются высокие требования;

- возможность работы пользователя с разным уровнем квалификации: от начинающих пользователей до профессиональных программистов;

- возможность расширения функций системы пользователем.

Jasenu CMS использует принцип полностью визуального редактирования, контента (при сохранении возможности редактирования кода «напрямую»), самоорганизации пользовательского, повышенной безопасности (уровень может быть изменён в сторону уменьшения).

Jasenu CMS поддерживает принцип построения системы администрирования, чтобы она являлась полностью автономной. Это значит, что админпанель может быть заменена или удалена вообще, что крайне редко, но может оказаться важной.

Интерфейсы подобного типа представляют собой сложные, зачастую тяжёлые для понимания структуры. С этой стороны Jasenu CMS заметно выигрывает у конкурентов. Её структура делится на исполняемые и взаимодействующие с пользователем модули – UI и EXEC. То есть два независимых модуля объединяются ядром, которое вызывает их.

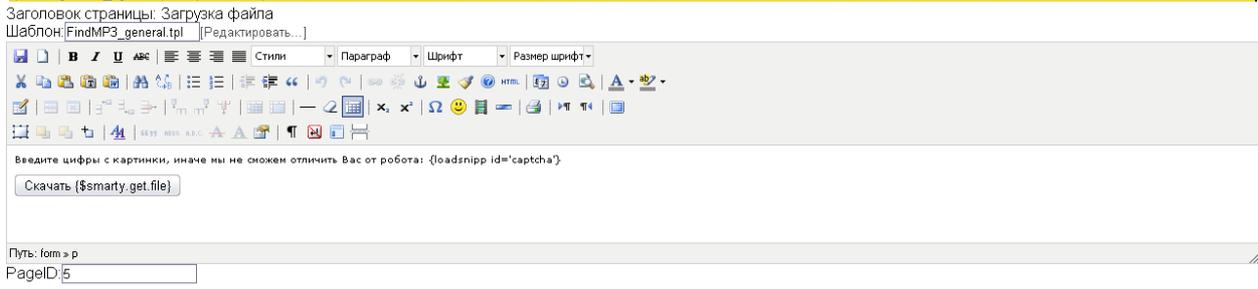
Расширить такую админпанель просто: достаточно скопировать файлы плагинов в нужные директории и зарегистрировать

их в системе, что может быть сделано через саму админпанель или с помощью установщика плагина.

Управление страницами сайта с сайта | Режим совместности (0)

Включён альтернативный режим редактирования!
 Файл данных: /data/pages/6_data
 Файл настроек: /data/pages/6_configs

```
array ( [varname_0] => content [varname_1] => title [varname_2] => alt_title [vartype_0] => string [vartype_1] => string [varval_0] => <form action="data/snippets/GetFile/action.php?stamp={Smarty.get.stamp}&file={Smarty.get.file}&path={Smarty.get.path}&mode={Smarty.get.pmode}" method="POST"> <p>Введите цифры с картинки, иначе мы не сможем отличить Вас от робота. {loadsnipp id='captcha'}</p> <input type="Submit" value="Скачать {Smarty.get.file}" /> </form> [varval_1] => Загрузка файла [varval_2] => Проверка бота )
```



Переменные, определённые на странице:

Переменная	Значение	Тип
content	<form action="data/snippets/GetFile/action.php?stamp={Smarty.get.stamp}&file={Smarty.get.file}&path={Smarty.get.path}&mode={Smarty.get.pmode}" method="POST"> <p>Введите цифры с картинки, иначе мы не сможем отличить Вас от робота. {loadsnipp id='captcha'}</p>	string
title	Загрузка файла	string
alt_title	Проверка бота	string

Добавить переменную:

Переменная	Значение	Тип
<input type="text"/>	<input type="text"/>	<input type="text"/>

Рис. 1. Модуль редактирования контента

Админпанель дружелюбна и к простым пользователям, и к программистам. Для тех и других она предоставляет разные режимы редактирования контента: визуальный и при помощи HTML-кода (Рис. 1).

В настоящее время ведётся разработка сайта проекта. Желающие ознакомиться с данной системой, бесплатно использовать ее для создания сайтов могут обращаться по адресу электронной почты: Livich@yandex.ru. По мнению авторов статьи, использование данного продукта является оправданным в средних учебных заведениях Украины, России и других стран СНГ.

Используемые источники:

1. Белоусов А. Вопросы легализации программного обеспечения в Украине //http://www.crime-research.ru/library/Belous1103.html
2. Херрнлебен Г. Программное обеспечение воруют не из-за его дороговизны, программы крадут только потому, что их возможно

украсть //http://itc.ua/node/34083/%5C'/node/34083/?page=10&s=a4472c1d021d791d66676655b4467ac4%5C'

3. Богапов Г. Пиратский вирус //http://expert.com.ua/8029.html

4. Экономить на программном обеспечении нужно и можно //http://www.exponenta.ru/educat/news/infoforvuz.asp

5. Романов Р.М. О лицензионном программном обеспечении в сельских школах //Информатизация сельской школы и жизнедеятельности молодёжи (Инфосельш-2009): Труды VI Всероссийского научно-методического симпозиума – Анапа. М.: РИЦ МГГУ им. М.А. Шолохова, 2009. – С. 204-209.

6. Лиликович С.А., Логинов А.В. Разработка Jasenu CMS6 //Перший крок у дослідженні: Збірник студентських наукових робіт-2009 р., м. Луганськ. – Луганськ: Phoenix, 2009. – С.74-77.

7. Фоменко А.В. Вимоги до системи управління контентом сайту //Сучасні тенденції розвитку інформаційних технологій в науці, освіті та економіці: Матеріали II Всеукраїнської науково-практичної конференції. 8-10 квітня 2008 р., м. Луганськ. – Луганськ: Альма-матер, 2008. – С.177-179.

УДК 372.8

ПРИМЕНЕНИЕ ГРАФИЧЕСКОГО РЕДАКТОРА TUX PAINT В ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

Л.А. Сафонова

Мордовский государственный педагогический институт

им. М.Е. Евсевьева

г. Саранск, Россия

E-mail: safonova.lan@yandex.ru

Аннотация

В статье рассматривается свободный графический редактор Tux Paint. Показаны его возможности в плане создания дидактических наглядных материалов.

Ключевые слова

Компьютерная графика, графический редактор.

Работа с компьютерной графикой – одно из самых популярных направлений использования персонального компьютера. В связи с этим современные графические средства разрабатываются с таким расчётом, чтобы не только дать удобные инструменты

профессиональным художникам и дизайнерам, но и предоставить возможность для продуктивной работы и тем, кто не имеет необходимых профессиональных навыков и врождённых способностей к художественному творчеству.

Для обработки изображений на компьютере используются специальные программы – графические редакторы. Графический редактор – это программа, предназначенная для автоматизации процессов построения на экране дисплея графических изображений. Одним из таких редакторов является Tux Paint.

Tux Paint – свободно распространяемая программа для рисования, ориентированная на детей. Эмблемой программы является пингвин – Tux символ операционной системы Linux, показанный на рисунке 1. Отсюда возникает первая часть названия программы. Вторая составляющая – слово Paint (с английского – «рисовать красками») — указывает на принадлежность программы к растровым графическим редакторам.

Графический редактор Tux Paint позволяет создавать изображения, сохранять их в собственном формате, печатать. Данный редактор содержит большой набор кистей, линий, форм, шрифтов, штампов, эффектов, некоторые из которых представлены на рисунке 2.



Рис. 1. Модальное окно графического редактора Tux Paint

Программа «Tux Paint» может служить замечательным средством развития изобразительных способностей детей, выполнять роль небольшой энциклопедии (прописано название каждого штампа, некоторые имеют звуковое сопровождение), да и просто являться прекрасным способом организации досуга.



Рис. 2. Шапки редактора Tux Paint и эффекты инструмента Магия

Данный редактор может быть полезным и преподавателям в качестве средства организации внеклассной работы и разработки дидактических материалов. Для примера рассмотрим технологию создания в графическом редакторе Tux Paint таблицы умножения и расписания уроков.

I. Таблица умножения.

Для её создания применяем следующий алгоритм.

1. Выбираем фон.
2. Выбираем инструмент *Линии*, проводим линию, состоящую из кошек, и получаем контур рисунка.
3. Выбираем кнопку *Текст* и прописываем строчку «1x1=1», нажимаем на *Enter* для перехода на следующую строчку. После нажатия кнопки *Enter* текст отредактировать нельзя. Продолжаем процесс до умножения на 10 и т.д.
4. Выбираем кнопку *Шапка* «миндаль» и ставим около столбца умножения на 1 и т.д. Результат показан на рисунке 3.

II. Рассмотрим алгоритм создания расписания уроков.

1. Выбираем фон.
2. Выбираем кнопку *Линии* и проводим контур.
3. Выбираем кнопку *Текст* и пишем заголовок «расписание уроков».
4. В пункте меню *Магия* организуем:
 - а) эффект *Граница*, который используем для обработки текста;

б) эффект *Заполнить*, с помощью которого заливаем буквы и область вокруг текста.

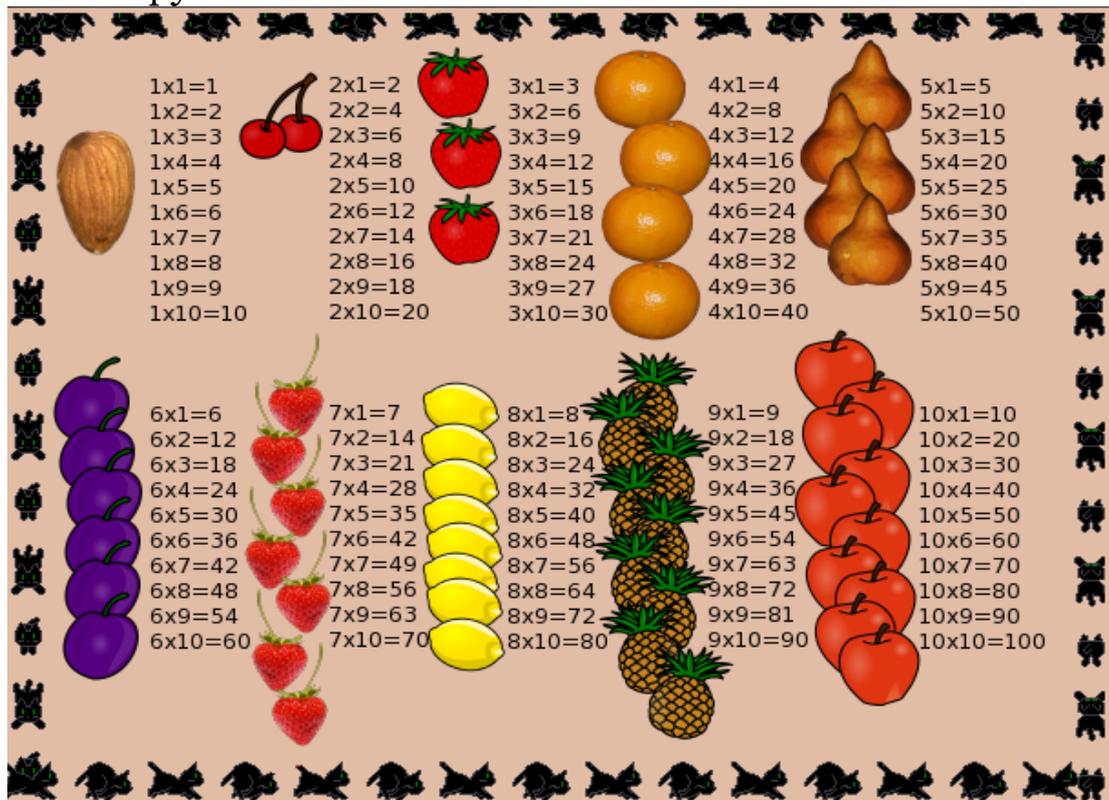


Рис. 3. Таблица умножения

5. Выбираем кнопку *Текст*:

- а) вводим названия дней недели;
- б) оформляем нумерацию уроков.

6. Выбираем кнопку *Линии* и проводим линии для названий уроков.

7. Выбираем кнопку *Магия* эффект *Резкость* и обрабатываем текст на рисунке, включая цифры.

8. Выбираем кнопку *Штамп* и вставляем изображения пингвинов.

9. Выбираем кнопку *Магия*, эффект *Цветок* и вставляем изображения цветов. Результат представлен на рисунке 4.

Таким образом, графический редактор Tux Paint может оказать помощь студенту, учителю, преподавателю для создания дидактических материалов, оформления документации. Достоинствами данной программы является также то, что она является бесплатной и легко устанавливается.

Однако у данной программы имеются свои недостатки.

1. После ввода текста его уже нельзя редактировать, то есть если допущена ошибка при написании, то придётся удалять слово (стереть ластиком) и прописывать его заново, после того как

текст установили в какой-либо области рабочего поля, его уже нельзя перемещать.



Рис. 4. Расписание уроков

2. Можно вводить только одну строку, нельзя прописывать текст в виде столбцов.

3. Выполнение работы должно осуществляться по строгому алгоритму, то есть нужно продумать весь путь создания проекта от начала до конца, так как что-то изменить будет невозможно.

4. Нельзя импортировать и экспортировать полученные рисунки, так как они сохраняются внутри самой программы и их нельзя использовать при работе в других программах. Решением данной проблемы может стать графический редактор, например, Paint, с помощью которого можно отредактировать извлечённые из Tux Paint объекты.

5. При вводе фигур их уже нельзя перемещать по рабочему полю, то есть мы можем вставлять их в какое-либо место рабочей области и только изменять их размеры, увеличивать или уменьшать.

6. Рабочая область строго ограничена, то есть весь наш проект нужно оформить на поле, которое мы видим на экране, чтобы никакие его части не выходили за установленные границы.

ИСПОЛЬЗОВАНИЕ СВОБОДНОГО РЕДАКТОРА 3D-ГРАФИКИ GMAX В УЧЕБНОМ ПРОЦЕССЕ

В.И. Сафонов

Мордовский государственный педагогический институт
им. М.Е. Евсевьева
г. Саранск, Россия

E-mail: wawans@yandex.ru

Аннотация

Статья посвящена рассмотрению возможностей свободного редактора трёхмерной графики GMax. Показаны его возможности, выделены достоинства и недостатки.

Ключевые слова

Компьютерная графика, графический редактор.

Современная система образования уделяет большое внимание использованию в обучении информационных технологий. Следует отметить, что это касается не только предмета «Информатика и информационные технологии», но и ряда других. Особое положение при этом занимает компьютерная графика, позволяющая сделать процесс обучения более наглядным и увлекательным. Способы применения компьютерной графики в обучении можно найти, например, в «Единой Коллекции цифровых образовательных ресурсов для учреждений общего и начального профессионального образования», доступ к которой осуществляется по электронному адресу <http://school-collection.edu.ru/>.

Компьютерная или машинная графика – это область деятельности, в которой компьютеры используются для построения и обработки изображений, а также результат этой деятельности. Данная область является относительно молодой, однако интерес к ней в последнее время значительно усилился.

Как показывает практика использования компьютерной графики в образовании, она позволяет развивать творческие и художественные способности учащихся, обогатить художественный инструментарий. Применение компьютерной графики в образовании связано с художественным образованием и эстетическим воспитанием.

Создание и широкое распространение программных средств, предназначенных для создания и редактирования изображений – графических редакторов, позволило говорить об их использовании в учебном процессе, чему в настоящее время имеется достаточно примеров. Подобные редакторы предлагается применять для построения чертежей, схем, планов и т.п. Они, как правило, обладают развитым инструментарием, а также дополнительными возможностями, которые не всегда могут быть реализованы традиционными изобразительными средствами. Учащиеся получают доступ в своеобразную изобразительную виртуальную лабораторию, предоставляющую им целый мир, в котором они могут творить, реализуя любые творческие фантазии.

Однако имеющееся многообразие подобных редакторов порождает новую проблему – проблему выбора среди обилия графических редакторов, обладающих различными возможностями. В связи с этим актуальным является чёткое представление их основных различий, чему и посвящена данная статья.

По способам задания изображений графику (и, соответственно, графические редакторы) разделяют на два вида: двумерная и трёхмерная графика. Двумерная, в свою очередь, подразделяется на векторную, растровую и фрактальную графику.

При использовании векторной графики изображение строится из геометрических примитивов (точки, прямые, окружности и др.), для которых задаются цвет линий и заливки и другие параметры. При сохранении изображения запоминаются данные параметры и математические уравнения линий, что занимает относительно небольшую область памяти. Векторное изображение можно трансформировать практически без потери качества, так как после преобразований изображение строится заново.

Все сказанное позволяет выделить некоторые положительные и отрицательные особенности векторной графики. С её помощью можно создавать достаточно объёмные по содержанию схемы, чертежи, анимацию, причём занимающие сравнительно небольшие объёмы памяти, что может оказаться необходимым при передаче данных в Интернет и др. Однако не каждое изображение (например, фотографию) можно построить из графических примитивов, что приводит к необходимости использования растровой графики.

К редакторам векторной графики относятся Corel Draw, Open Office Draw, Adobe Illustrator, Macromedia Freehand и др.

Растровое изображение строится из точек (точнее, из массива или матрицы точек), для каждой из которых сохраняется информация о расположении на экране, цвете, яркости и др. Растровые изображения отличаются от векторных, как правило, высоким качеством, но большие объёмы занимаемой при сохранении памяти. Кроме этого, следует отметить, что подобные изображения очень сложно редактировать. Так, при уменьшении растрового изображения особых искажений может и не наблюдаться, но при увеличении размеров станут увеличиваться составные его части – точки, что приведёт к ухудшению качества изображения: чем больше увеличение, тем больше ухудшение.

Основное преимущество растрового изображения заключается в том, что оно позволяет создавать любое изображение. Как нами было отмечено, этот вид графики имеет свои недостатки: большой объём памяти, требуемый для сохранения изображения, а также потери при редактировании, особенно при масштабировании.

К редакторам растровой графики относятся такие, как MS Paint, Adobe PhotoShop, GIMP и др.

Фрактальная графика подразумевает построение объекта, отдельные элементы которого наследуют родительские структуры. Описание меньших элементов происходит по некоторому простому правилу, что позволяет описать такой объект несколькими математическими уравнениями (снежинка, кривая Коха и др.).

С помощью фракталов можно создавать целые классы изображений, для хранения которых требуется относительно мало памяти. К недостаткам этого вида графики относится то, что для построения объектов, не относящихся к фрактальным классам, его применить либо весьма сложно, либо невозможно. К редакторам фрактальной графики относятся такие, как Art Dabbler, Ultra Fractal, Fractal Explorer и др.

В трёхмерной компьютерной графике объекты представляются, как правило, в виде совокупности минимальных поверхностей (полигонов), определяемых наборами из координат вершин. Созданные трёхмерные объекты можно использовать для создания сцен, в том числе анимационных. Трёхмерная компьютерная графика широко используется в кино, компьютерных играх. В настоящее время разрабатываются системы «Виртуальная реальность», которые планируется применять в обучении. Это, например, конструирование в трёхмерном пространстве с использова-

нием предметов виртуального мира; работа с использованием трёхмерных тренажёров и др.

Существует достаточно много программных пакетов, позволяющих создавать трёхмерные графические изображения, то есть моделировать объекты виртуальной реальности. Среди известных программ в данной области можно выделить такие, как, например, Autodesk 3ds Max, Newtek Lightwave, 3D Canvas, Sweet Home 3D, Maya и др. Следует также отметить использование трёхмерной графики в системах автоматизированного проектирования, где требуется создание объёмных моделей: деталей, заготовок, зданий и т.п.

Компьютерная графика становится одним из направлений изобразительного искусства, что приводит к необходимости как подготовки специалистов данного направления, так и формирования умения работы с компьютерной графикой как элемента информационной культуры. В связи с этим современный школьник должен представлять возможности создания изображений с использованием компьютерной графики, владеть средствами создания и редактирования компьютерных изображений. Это позволит ему формировать свою информационную культуру и развивать творческие способности.

Многие эффекты в современных клипах и фильмах создаются с помощью компьютера. При этом широко используется трёхмерная графика (3D, third dimension – третье измерение). С её помощью на плоском экране имитируется движение объёмных объектов в трёхмерном мире.

Впервые компьютерные эффекты были широко использованы в фильме «Звёздные войны» (режиссёр Джордж Лукас, 1977 год). В современных фильмах нередко некоторые персонажи моделируются на компьютере, существует множество полнометражных фильмов, которые полностью созданы с помощью трёхмерной графики и анимации, например, известная серия про Шрека.

Gmax – это программа для создания 3D-графики и анимации. Она представляет собой упрощённую версию всемирно известной программы 3ds Max, которая считается стандартом de facto в мире профессионалов трёхмерной графики. Однако 3ds Max – коммерческая программа, стоимость которой составляет более 150 000 рублей (версия 3ds Max 2008). В то же время Gmax – бесплатная программа, разработанная той же фирмой Autodesk (ранее она называлась Discreet) на основе версии 3D Studio 3.1.

По задумке фирмы Autodesk, основное назначение Gmax – разработка различных моделей для трёхмерных игр. Однако ее можно с успехом применять для начального освоения 3D-графики. Тем более, что интерфейс и основные методы работы в Gmax точно такие же, как и у профессиональной программы 3ds Max. К сожалению, в октябре 2005 года фирма Autodesk прекратила поддержку Gmax, однако эта программа доступна для скачивания на сайте www.turbosquid.com.

Программа Gmax позволяет:

- создавать трёхмерные модели;
- применять к моделям материалы;
- настраивать освещение;
- создавать анимацию с 3D-объектами.

При этом есть и недостатки:

- поддерживаются далеко не все возможности 3ds Max, особенно современных версий;
- отсутствует рендеринг (визуализация) – построение фотореалистичного изображения и анимации.

Усилиями энтузиастов вторую проблему удалось частично решить, используя бесплатную программу YafRay. Хотя создание анимационных роликов (формата AVI) она не поддерживает, но возможен рендеринг по одному кадру.

Файлы 3D-сцен, созданные в Gmax, имеют расширение *.gmax. Это особый формат, который не читают никакие другие программы. Однако можно установить дополнительные модули (экспортёры), которые позволяют записать сцены в других (более распространённых) форматах и таким образом передать модели в другую программу, например в 3ds Max.

Почему мы обратили своё внимание на Gmax? Во-первых, программа распространяется бесплатно и относится к продуктам трёхмерного моделирования с наиболее дружелюбным к пользователю интерфейсом (это значит, что для простого пользователя вполне реально изучить основные компоненты программы самостоятельно). Кроме того, Gmax по сравнению со своим многофункциональным родителем занимает значительно меньше места на диске, но обладает всеми необходимыми инструментами.

«МЫСЛИ О LINUX» ШКОЛЬНИКА ПРОВИНЦИАЛЬНОЙ ГИМНАЗИИ

Е. Мироненко

МОУ гимназия № 11

М.А. Губин

ГОУ ВПО «Елецкий государственный университет
им. И.А. Бунина»

г. Елец, Россия

E-mail: gubinma@yandex.ru

Аннотация

В данной статье делается попытка с точки зрения ученика гимназии провинциального города рассмотреть целесообразность использования операционной системы Linux в повседневной жизни. Также уделяется внимание «любимому» дистрибутиву гимназиста – Sabayon.

Ключевые слова

Операционная система, Linux, Sabayon.

Мир операционных систем огромен. Но в настоящее время на мировом рынке лидируют 3 системы — это Windows, Linux и MAC OS.

Главная особенность Linux заключается в том, что он относится к свободному программному обеспечению. Мы можем использовать его с любой целью: можем изучать, как он работает, и адаптировать под себя, можем улучшать и распространять свою улучшенную версию, в конце концов, можем просто дарить всем окружающим.

Так, чем же Linux отличается от остальных операционных систем?

Во-первых, высокой надёжностью и безопасностью. Система очень редко зависает и работает очень стабильно, а сбой программы не ведёт к зависанию всей системы. О потере данных можно практически не волноваться (конечно, в разумных пределах). Можно забыть про сторонние антивирусы и фаерволы — программ, которые уже включены в дистрибутив, будет вполне достаточно для безопасной работы.

Во-вторых, относительно малым использованием системных ресурсов. Ведь не секрет, что системные требования почти каждого дистрибутива меньше, чем у Windows Vista и Windows 7, которые в настоящее время распространяет компания Microsoft в качестве основных операционных систем. А «красоты» при этом, по крайней мере, не хуже.

В-третьих, практически не требует установки каких-либо драйверов. Все необходимые драйвера ставятся автоматически во время установки системы. Приходится с сожалением сознавать, что пользователи, привыкнув к «благам для пользователя» Windows, требующей для каждой «железки» драйвер, не могут понять, что в Linux большинство комплектующих будет работать уже «из коробки», и установка дополнительных драйверов может потребоваться лишь для относительно нового или редкого оборудования. Система сама определяет, какое у нас оборудование, ещё на стадии установки.

И, в-четвёртых, на наш взгляд, лучшее качество Linux — это то, что данная операционная система использует открытые стандарты и абсолютно бесплатна. Linux не просто свободная система — любой пользователь, будь он из какого-нибудь Нью-Йорка или Ельца, может сделать свой вклад в развитие дистрибутива. Также можно увидеть исходный код любой программы, отредактировать его и распространять эту программу. Есть, конечно, и платные дистрибутивы, но в них, в основном, плата берётся либо за дополнительные услуги (техподдержка, реклама, обучение), либо за проприетарные программы, включённые в данный дистрибутив.

В отличие от MAC OS и Windows, в Linux есть много графических сред, среди которых наиболее популярными являются KDE и GNOME. Каждая графическая среда представляет собой интерфейс и набор программ для работы под этим интерфейсом. Переходя с KDE на GNOME и наоборот, мы полностью изменяем внешнюю оболочку системы. С виду кажется, что мы используем уже совершенно другой дистрибутив. Многие считают, что дистрибутивы Linux выглядят не так красиво, как, например, Windows 7 с оболочкой Aero. Нет, это не так. Напомним, что Linux можно полностью настроить под себя — и стиль не исключение. Благодаря композитному менеджеру окон Composit или фреймворку Plasma можно добавить любой графический эффект для рабочего стола.

Многие думают, что в Linux очень мало программ, которыми мы привыкли пользоваться. Но данное утверждение не соответствует действительности — тут есть свои репозитории, с которых и ведётся скачивание программ. В Windows же нам придётся искать эти программы в поисковых системах, качать их себе на ПК и, вполне возможно, платить деньги за право пользования той или иной программой. При этом в репозиториях находится более миллиона приложений, отсортированных в сотни разных категорий, и все приложения бесплатны.

Стоит отметить тот факт, что даже обычный пользователь в Windows по умолчанию может сделать слишком много — настолько много, что достаточно написать какой-нибудь «вирус» и всё, мы взломали всю систему. В Linux такого не происходит. Перед тем, как изменить что-то в системе, у нас спрашивают пароль от суперадминистратора root. Например, если пользователь откроет какой-то подозрительный файл и у него спросят пароль от root'a, с достаточной долей уверенности можно сказать, что файл может оказаться вредоносный.

Самое интересное в Linux — это командная строка. Она является неотъемлемой частью системы. Во многих операционных системах настройка системы ведётся с помощью графической среды, но пользователи, не знакомые с конкретным дистрибутивом, чаще пользуются командной строкой. Она позволяет пользователю полностью перенастроить все функции системы. С помощью неё также можно восстанавливать систему, даже если графическая оболочка не загружается.

Кроме Linux существуют и другие операционные системы, которые имеют лицензию GNU GPL, разрешающую свободное копирование. Например, FreeBSD. Работа для специалиста в FreeBSD не составляет никакого труда, но для новичка более прост в освоении и использовании всё-таки Linux. Но обе системы — отличные. Одна чем-то хуже, чем другая, и наоборот. Какой бы выбор мы ни сделали — Linux, FreeBSD или, например, OpenSolaris — это будет правильный выбор.

Теперь остановимся на моём «любимчике». Дистрибутивов Linux множество, и если вы хотите перейти на один из них, то необходимо, для начала, определиться, какой вам больше подходит. Лично мне больше понравился Sabayon.

Sabayon создан на основе проекта Gentoo итальянским разработчиком Fabio Erculiani в апреле 2008 года. В данный момент

команда Sabayon и Gentoo тесно сотрудничают. Но в отличие от Gentoo, данный дистрибутив более дружелюбен к пользователю. Gentoo же построен по принципу «сделай сам», — пользователь сам устанавливает все из исходных пакетов. В Sabayon же простая графическая установка — многие пакеты устанавливаются сразу. Как и Gentoo, Sabayon работает по технологии Portage — это сердце системы, выполняющее множество функций. Через Portage собираются и устанавливаются пакеты, обновляется сам дистрибутив. Можно превратить систему в идеально защищённый сервер, рабочую станцию разработчика, систему для игр — во всё, что можно пожелать.

В последней версии Sabayon 5.2 оптимизировали графическую среду, обновили все системные пакеты, сделали новую распределённую систему управления пакетами Sulfur и Equo (в первую очередь, бинарными). По сравнению с 4.x, современная версия работает намного стабильнее.

Количество установленных программ приятно удивляет. У нас сразу же стоят такие программы, как Skype, Google Earth, множество аудио- и видеопроигрывателей, кодеков и других программ. Многие скажут «понапилили всё в кучу», но пользователи с медленным Интернетом или вообще без него скажут только «спасибо».

Sabayon оформлен в виде Live-дистрибутива. Дистрибутив снабжён возможностью лёгкой установки на жёсткий диск благодаря встроенному инсталлятору. Но это один из немногих дистрибутивов Linux, который очень требователен к «железу». Поэтому, к сожалению, без конфликтов с оборудованием не обойтись., но все проблемы довольно легко решаются, особенно если есть подключение к «всемирной паутине».

Сравнить данный дистрибутив с остальными дистрибутивами Linux достаточно тяжело. Например, Ubuntu ориентирован на лёгкость в освоении и использовании, openSUSE отличается удобством настройки и обслуживания, Fedora ориентирован на применение самых последних версий программ. Лично меня не удивил ни один из этих дистрибутивов — все они рано или поздно отказывали. Но не Sabayon.

Sabayon, с одной стороны, можно назвать «дружелюбным» дистрибутивом, а с другой — «суровым», который нужно долго настраивать под себя. Но в итоге он станет стабильной и надёжной системой — он специально для тех, кто хочет ощутить всю мощь

системы сборки/установки пакетов portage, но при этом не желает тратить много времени для компиляции всей системы, графического окружения, рабочего стола и необходимых приложений. Этот дистрибутив можно порекомендовать как начинающим, так и продвинутым пользователям.

Из всего вышесказанного можно сделать вывод о том, что переходить на Linux или нет — решать придётся самому пользователю. Чтобы «ладить» с системой, нужно совершать немало умственных усилий. Операционная система Linux не для тех, кто любит делать всё в один клик мыши, — она требует стремления, работы, желания, — систему нельзя понять сразу за пару минут. С другой стороны, разбирающийся человек сразу научится пользоваться Linux и делать на нём всё, что захочет.

УДК 004.451

ПРОБЛЕМЫ АДМИНИСТРИРОВАНИЯ КОМПЬЮТЕРНОГО ПАРКА ШКОЛЬНЫХ ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ НА ОСНОВЕ ТЕХНОЛОГИЙ «ВИРТУАЛЬНЫЙ КОМПЬЮТЕР», «ТОНКИЙ КЛИЕНТ», «БЕЗДИСКОВАЯ РАБОЧАЯ СТАНЦИЯ»

А.О. Козлов

ИИО РАО

г. Москва, Россия

E-mail: ole-kozlov@mail.ru

Аннотация

В данной статье предлагается один из подходов к осуществлению администрирования компьютерного класса на основе технологий «тонкий клиент» и «бездисковой рабочей станции».

Ключевые слова

Образовательной учреждение, компьютерный парк, тонкий клиент, бездисковая рабочая станция, виртуальный компьютер.

На данный момент техническая сторона школьного компьютерного парка внутри одного района, как правило, выглядит следующим образом. В каждой школе есть один или несколько

компьютерных классов, на компьютерах которых, как правило, установлено определённое программное обеспечение. Во время занятий и кружков ученики и учителя вносят какие-то изменения в систему: меняют настройки рабочего стола, создают новые файлы, ставят программы, ходят по Web-сайтам. В результате чего необходимо администрировать систему, т.е. выполнить следующие действия:

- поставить на все компьютеры новую программу;
- поставить на все компьютеры антивирусное ПО, дефрагментировать диск, вернуть настройки, поставить очередное обновление системы;

- «починить компьютер» — когда операционная система практически стала неработоспособной, кто-то должен с нуля переустановить её и все требуемые программы;

- аппаратная часть компьютера может сломаться. Чаще всего из строя выходят жёсткие диски — ведь в них есть движущиеся механические части. Современные жёсткие диски имеют низкую надёжность, так как предполагается что через два три года купят новые. Кто-то опять должен поставить на компьютеры все программы;

- наконец, последнее время кто-то в школе несёт ответственность за лицензионность установленных программ. Он должен проверять каждый компьютер на то, нет ли там каких либо нелегальных программ. Если какой-то школе дают новые компьютеры, то на них тоже нужно установить все требуемое ПО.

Получается, что обслуживание одного компьютера в несколько раз превышает стоимость его аппаратной части и установленных на него программ.

Каждый компьютерный класс работает сам по себе. Администрирование одного компьютерного класса не распространяется на администрирование других компьютерных классов. Каждый компьютер требует обслуживания в школе. Для установки ПО требуется определённое усилие со стороны школы. Для поддержания работоспособного класса тоже требуются определённые усилия. Добавление/замена компьютера требует внимания администратора. Администратор компьютерного парка в школе несёт ответственность за нелегальное ПО. Защита информации осуществляется каждым администратором самостоятельно и, как правило, будет на низком уровне.

Современное развитие информационных технологий позволяет целиком избавить школу от администрирования компьютеров. Предполагается следующий порядок работы компьютерного класса:

1. Учитель после включения своего компьютера выбирает в меню, какой урок он хочет сейчас провести в этом компьютерном классе. При этом в райцентре есть заранее утверждённое расписание, в какой школе, в каком компьютерном классе и в какое время какой идёт «компьютерный» урок, то есть урок с применением информационных технологий. Так что по умолчанию учителю показано, что по расписанию он должен выбрать определённое занятие.

2. На компьютер учителя и все компьютеры учеников загружается единая компьютерная среда. То есть некоторый виртуальный компьютерный класс — определённая операционная система — определённый набор программ для конкретно этого занятия. Программы для другого занятия не установлены — набор заготовок и примеров — учитель может смотреть, кто чем занят, брать контроль за управлением компьютера ученика, а так же переводить все компьютеры в режим просмотра того, что делает учитель. То есть учитель информируется о том, что происходит в компьютерном классе и в зависимости от этого принимает решение - работа компьютерного класса записывается (то есть с каждого экрана ведётся запись как видео фильм), чтобы потом учитель мог просмотреть, как прошло занятие и улучшить методику преподавания, либо же похвалить/поругать учеников.

3. На следующий урок может быть загружена совершенно другая единая компьютерная среда — другая операционная система, другой набор программ. Работа компьютерного класса не зависит от того, как с ним работали вчера или позавчера. Ученику и учителю при необходимости предоставляется возможность сохранить свои данные, например, на флэшку или на сетевой диск. Все же другие изменения, внесённые в виртуальный компьютерный класс «забываются».

Итого мы хотим достичь следующего:

1. Никто в школе не занимается установкой/удалением программ и настройкой операционных систем. В школе может изучаться и свободное ПО (ОС Линукс), поскольку за него не надо платить, и платное ПО (WinXP + Office 2007), поскольку в некоторых школах родители готовы заплатить деньги на то, чтобы их

дети обучились работе именно в Office 2007, а не в его бесплатных аналогах.

2. Ученики могут изучать администрирование/настройку операционных систем не опасаясь, что приведут компьютеры в неработоспособное состояние. В случае выхода компьютера из строя в компьютерном классе этот компьютер может быть заменён на любой другой компьютер, после чего при минимальном участии специалиста компьютер «подключён» к классу и может быть использован для проведения занятий

3. В районном центре формируется расписание, набор эталонных «виртуальных компьютерных классов». То есть там один раз настраивается компьютерный класс для, скажем, урока рисования в 3-м классе, занятия №1-10. Один раз настраивается компьютерный класс для изучения языка программирования Pascal в 5,6,7 классах. И так для всех занятий. Если в 10 компьютерных классах нужно теперь провести урок по рисованию — то это не требует никаких усилий со стороны администратора.

4. Защита информации (ЗИ) будет вестись централизованно с участием одного специалиста по ЗИ. Если, например, какой-то сайт не рекомендован к просмотру — специалист это настраивает один раз в райцентре, а не в каждой школе.

5. В районном центре формируется маленький штат из высокопрофессиональных специалистов, которые обеспечивают работу всех виртуальных компьютерных классов и при необходимости приезжают в школу и заменяют компьютер в случае выхода из строя его аппаратной части.

Для достижения поставленных целей не требуется написание нового программного обеспечения. По отдельности всё необходимое ПО (и коммерческое, и его бесплатные аналоги) уже существует. В некоторых отраслях эти технологии используются уже больше 10 лет.

«Виртуальный компьютер» — на операционной системе запускается программа, которая внутри себя целиком эмулирует персональный компьютер. Виртуальный компьютер можно сохранить и потом вернуться к сохранённому месту, можно изменить количество оперативной памяти, подключить новый диск. Файл с описанием виртуального компьютера может быть переписан на другой компьютер и запущен там в эмуляторе. То есть копирование компьютера сводится к копированию 1-10 файлов, описывающий компьютер. Раньше виртуальные компьютеры работали

медленно — скорость работы внутри виртуального компьютера в разы (до 10 раз) медленнее, чем запуск программы на обычном компьютере. 2-3 года назад все процессоры стали поддерживать аппаратную виртуализацию — технологию, которая позволяет программам работать внутри виртуального компьютера практически так же быстро, как и на настоящих компьютерах.

«Тонкий клиент» и «Бездисковый компьютер» — некоторые технологии, которые позволяют компьютеру обходиться без жёсткого диска. Компьютер имеет минимальный загрузчик, который по сети грузит очень простую операционную систему. После этого система подключается к другому нормальному компьютеру (в том числе и к виртуальному компьютеру) по сети и фактически становится терминалом. Терминал не выполняет каких либо программ - программы запущены на обычном компьютере, а по сети передаётся содержимое экрана и нажатие клавиш. Для успешной работы требуется быстрая связь (сейчас легко можно получить 100/1000) Мбит внутри школы, мощный терминальный сервер.

Эти технологии можно соединить и создать компьютерную среду. Для обслуживания такой среды потребуется гораздо меньше специалистов, но они должны быть высокой квалификации.

УДК 004.9+ 004.4

ОПЫТ ВНЕДРЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРОФЕССИОНАЛЬНОМ УЧИЛИЩЕ

А.С. Барышева

ГОУ НПО «Профессиональное училище №62»

с. Ванновское, Краснодарский край, Россия

E-mail: Sasha4Barysheva@yandex.ru

Аннотация

Анализируется опыт внедрения свободного (открытого) программного обеспечения в образовательном учреждении НПО. Приводятся примеры применения некоторых прикладных программ. Высказывается мнение о дальнейшем распространении СПО в образовании.

Ключевые слова

Свободное (открытое) программное обеспечение, образование, прикладные программы для профессионального образования.

В настоящее время развитие информационной культуры и, в частности, формирование информационно-коммуникационных (в том числе и компьютерных) компетенций становится определяющим фактором повышения эффективности обучения и развития вообще. Это правило относится как к обучаемому, так и к обучающему. Современные условия диктуют: кто владеет информацией — тот владеет миром. ИКТ позволяют не только получать адекватную информацию и грамотно её использовать, но и распространять, транслировать её оптимальным образом, в том числе в целях образования других людей. Вот почему освоение и распространение опыта применения ИКТ в образовании прописано в моём личном плане самообразования, в плане работы методической комиссии преподавателей общеобразовательных предметов нашего училища, которой я руковожу. В рамках творческой группы «Инновационные технологии» проводятся регулярные семинары и практикумы по обучению педагогов информационным технологиям, а руководство училища уделяет пристальное внимание повышению эффективности учебного процесса, в том числе посредством ИКТ. С 2005 года процесс внедрения новых информационных технологий принимает организованный характер: ежегодно приобретает компьютерная техника, обеспечивается доступ в Интернет. На этом пути одной из главных трудностей у нас становится отсутствие грамотной технической поддержки: вопросы приобретения техники — от системных блоков до интерактивных комплексов — и программного обеспечения решаются целиком нашими собственными силами, на свой страх и риск, к тому же за счёт собственных (внебюджетных) средств.

На фоне общеобразовательных учреждений, получивших пакет Первой Помощи из государственного кармана, регулярно (хотя и очень недостаточно!) обновляющих парк имеющейся компьютерной техники, да к тому же педагоги которых имеют возможность пройти обучение «компьютерной грамотности» в рамках повышения квалификации опять-таки за государственный счёт, — наше бытие сладким не назовёшь.

Вот почему, как наиболее «продвинутый» «компьютерщик» в нашем училище (в 1995 году я закончила Казахский химико-технологический институт по специальности «Автоматизация процессов и производств химтехнологии», где на нашей кафедре АСУТП стояли самые новые по тогдашнему времени GoldStar'ки, а на занятиях английского для практики технического перевода мы

помогали переводить техдокументацию на пакет программ — будущего Office), я пытаюсь отслеживать возможности приобретения недорогих (а лучше бесплатных) и в то же время эффективных программных продуктов.

В 2006-07 учебном году училище приобрело новые компьютеры для администрации и в методический кабинет. Поскольку сэкономить нам необходимо, было предложено установить на них ОС, распространяемые по лицензии GNU. Однако оказалось, что специалистов по установке и настройке, а также дальнейшему сопровождению таких ОС на ближайших к нам территориях не нашлось — только в Краснодаре. Поскольку моя «компьютерная квалификация» такого рода работ не предусматривает, решено было установить ОС Windows и пакет OpenOffice.org. Провели обучение преподавателей и мастеров производственного обучения группами по 6 человек на трёх машинах, на одной из которых был MS Office'95. При этом никаких трудностей с адаптацией, переходом с OO на MS_O и обратно не возникало как у полных новичков, так и у «бывалых».

Поскольку преимущества свободного программного обеспечения, по крайней мере, при его приобретении не вызывают сомнений, к работе по обеспечению училища СПО подключены и наши учащиеся. Объясняется это не только и не столько образовательной необходимостью, но и ещё одной проблемой нашего училища, связанной с ИКТ: отсутствием скоростного доступа в Интернет. Имеется только модемный выход, а о других наши провайдеры отвечают одно: нет технической возможности. Наше село Ванновское расположено в 12 км от районного центра, станицы Тбилисской, в которой есть и ADSL, и «Дизель». К тому же мы на левом берегу Кубани, а они — на правом. Выходили мы даже на главу района с просьбой помочь — на всей левобережной части района те же трудности, имеющиеся точки доступа, например, в почтовых отделениях, терминалы в отделениях Сбербанка работают при скорости не выше 26КБ/с, да ещё и с перебоями. Кстати, после перехода на пластиковые карты мы ездим получать зарплату в станицу — банкоматы работают нормально только там, и всё по той же причине — нет технической возможности, а пластиковой картой можно расплатиться только в 5 магазинах, торгующих бытовой техникой, опять-таки в станице.

Мы не сдаёмся. Прорабатывается возможность установки спутниковой антенны. А до того времени мои «провайдеры» — это

наши учащиеся. Так, благодаря им в интерактивном кабинете математики имеется практически полный набор СПО — спасибо сайту FreeSource. Перейти на «чистый Open Source» не позволяет уже имеющееся оборудование: система интерактивного голосования InterWrite PRS, например, имеет встроенный модуль для Power Point, а на Impress не работает. Это во-первых, а во-вторых, подготовленные разработки для уроков (тесты, интерактивные графики, шаблоны) работают на основе VBA, который OpenOffice «не понимает».

Коротко расскажу об использовании имеющихся прикладных программ FOSS в моей работе. Первое — пакет OpenOffice.org установили во всём училище, поэтому внутри учреждения никаких проблем не имеем. Здесь скорее трудности у тех наших коллег (и вышестоящих организаций), которые требуют предоставления информации в формате .doc и .xls (а для большинства из них приходится всё дублировать и на бумажном носителе — факс освоили быстрее, чем электронную почту). Наиболее опытные пользователи компьютера к стандартному пакету прибавили PDF Creator и WinDjView, а математикам советую ещё и LyX. Пропагандирую FreeMind — составление опорных конспектов, карт памяти, структурирование информации на уроках и методичке помогает эффективно усвоению информации.

Графические редакторы: конечно же, GIMP и Inkscape. А в кабинете психолога — ещё и Tux Paint: всем и каждому, ребёнку и взрослому, — обязательно нужно попробовать «погонять кроликов», как выразился один из моих учеников, или хотя бы лягушек, на что способен только этот, такой простенький альбом для рисования. Кстати, упомяну ещё о бесплатных, хотя и не с открытым кодом, программах ЯндексКраски и Google SketchUp. Первая — тоже больше для разрядки эмоций, которая так необходима нашим детям, в большинстве своём из неполных семей, с ущербным детством. А вторая, не менее эффективная в плане психологической разгрузки, да к тому же позволяющая создавать реалистичные трёхмерные конструкции с любой степенью детализации. Для ребят с хорошей математической подготовкой на уроках информатики пробую использовать POV-Ray, в нашей коллекции уже несколько реалистичных изображений деталей машин и механизмов — такая работа помогает прояснить некоторые моменты геометрических построений в пространстве, связь уравнений и реальности.

Из браузеров, по моим наблюдениям, чаще всего применяются Opera и SeaMonkey. FireFox менее популярен только из-за невозможности сохранять странички в Web-архиве.

Интересен IUP Portfolio как платформа для разработки индивидуальных и групповых проектов учащихся, формирования портфолио. К сожалению, имеющаяся у нас документация только на английском.

Перехожу к самой, пожалуй, главной трудности во внедрении ИКТ на должном уровне. Сложно бывает приучить к эффективному использованию ИКТ. Казалось бы, мелочи — использование стилей, шаблонов документов для повседневной работы, порой нумерованные списки и начало раздела с новой страницы делаем вручную, без помощи главного помощника в таком деле — компьютера, компьютерной программы. А мелочи эти перерастают в нежелание? неумение? боязнь? использовать почтовую программу, к примеру. Это относится, к сожалению, даже к молодым нашим сотрудникам, к счастью, большинство из таких «упорных» — не педагоги. Вот не далее как вчера наши уважаемые историк и русовед — женщины, обременённые бытовыми и семейными проблемами, с педстажем в 25 и 30 лет, узнав, что училище будет приобретать мультимедийную технику, в один голос твёрдо заявили: «И мне мультимедиа!». В кабинет физики будет установлена интерактивная доска — тоже при деятельном участии нашей знаменитой Веры Антоновны Мажарцевой, которая преподаёт в нашем училище уже 29 лет без перерыва. И я уверена, что сложностей с обучением и адаптацией к СПО, о которых любят порассуждать адепты проприетарного ПО, у моих старших коллег будет не больше, чем с пакетом от Microsoft.

Увы, одним энтузиазмом даже такое хорошее дело не вытянешь. Следовало бы организовывать и обучение, и техподдержку ОУ НПО таких, как наше, на краевом уровне. Только в последнее время вроде бы начинают разворачиваться к нам лицом поближе методические службы края. Проводятся семинары по ИКТ, круглые столы под руководством НМЦ НПО, ККИДППО. Больше возможностей трансляции опыта применения ИКТ в образовании, шире распространение различных прикладных программ, в том числе СПО. Но эта деятельность касается именно методической, педагогической стороны работы. Техническая поддержка целиком — дело рук самих училищ.

Вот пример. Комплектуем кабинет информатики. В целях

экономии это будет терминальная система, включающая также интерактивную доску, мультимедиа проектор. Согласно существующей системе госзакупок, приобретать товар в магазине, на складе, в специализированной фирме, мы не имеем права — только после электронного аукциона, куда мы предоставляем необходимые технические характеристики и примерную стоимость заказа (что уже составляет довольно трудоёмкую задачу как для не сведущего в компьютерной технике инженерно-педагогического работника, так и для самого лучшего ИТ-специалиста, не имеющего понятия об особенностях применения ИКТ в образовании). Затем в течение довольно продолжительного времени ждём окончания торгов и получаем, практически, «кота в мешке» — товар прибывает только после оплаты по договору. Имеющиеся несогласования с заявленными на аукцион параметрами приходится решать в рабочем порядке, на что, бывает, уходят месяцы. Так проходит время, когда на рынке появляются новые, ещё более заманчивые предложения — вот почему в заявке всегда закладываем несколько более мощные данные, с возможностью модернизации.

Далее, в процессе эксплуатации, неизбежны неполадки, для устранения которых приходится вызывать сторонних специалистов. И тут начинается песня: фирмы по обслуживанию не пытаются сохранить штат опытных сотрудников, им выгоднее набирать молодёжь, потому текучка там страшная. К нам каждый раз приезжают новые техники. Об уровне квалификации любого из них можно лишь догадываться. И ни разу, повторюсь, на территории нашего Тбилисского района, а также в фирмах-поставщиках я не нашла специалиста по Linux, даже с OpenOffice техники знакомы «шапочно» — офисные программы их не интересуют. Впрочем, это как раз понятно — прибыль от «чистки» компьютеров от вирусов гарантирована именно в Windows, а продажа бесплатного ПО — неблагодарная задача.

Так что решаем свои проблемы собственными силами: при каждом удобном случае обмениваемся номерами телефонов с педагогами других училищ — этакая доморощенная сеть технической поддержки. Организовали свой сайт math-krasnodar.wikia.com, специально для «любителей СПО» (пока, правда, с минимумом информации).

Я думаю, теперь понятно, почему мы со вниманием следим за предложениями, высказываемыми уже с 2007 года, о внедрении СПО в школы, полностью поддерживаем предложения IV кон-

ференции «Свободное программное обеспечение в высшей школе», например, А.А. Якушина: «Необходимо обеспечить поставку оборудования с уже предустановленным ПСПО (как самостоятельно, так и в качестве второй операционной системы)» и далее: «... для успешного развития необходимо постоянное взаимодействие с учителями; энтузиастами СПО на местах, во многих случаях готовых оказывать безвозмездную помощь в продвижении свободных продуктов; с преподавателями и студентами ВУЗов.» А также предложение И.В. Воронина об организации «центров компетенции» на местах. Эту инициативу я готова поддержать личным участием — фактически в нашем училище, а также ближайшей МОУ «Ванновская СОШ №4» подобный центр уже действует, в моём лице :-).

Следует обратить внимание также на то, что обучение педагогов информационно-компьютерным технологиям, организованное как Академией повышения квалификации и профессиональной переподготовки работников образования (АПКиППРО), так и КРИПКиПРО (факультетом информационно-коммуникационных технологий в образовании), реализуется совместно с компаниями Microsoft и Intel — проекты «Партнёрство в образовании» и «Обучение для будущего». Обучение на этих курсах прошли многие из моих коллег, все материалы доступны в Интернете. О подобной поддержке FLOSS можно только мечтать. Ведь даже самый распространённый сейчас учебник информатики фактически прокладывает путь Microsoft, а великолепные идеи уважаемого мной А. Г. Кушниренко, к сожалению, не получают должного внимания из-за менее доступного ученикам языка изложения, как подтверждают многие мои коллеги.

Как итог хочу сказать, что потенциал свободного открытого программного обеспечения неисчерпаем, особенно в образовании, у них одни корни — свобода познания и безграничность реализации личности. Необходимо поддержать и углубить существующее движение по распространению СПО. «Низы» уже готовы. Дело за «верхами».

P.S. Хотелось бы, чтобы в очередной компании по поставке ПО в школы вспомнили бы и о нас, учреждениях начального профессионального образования.

Используемые источники:

1. <http://www.lyx.org>
2. <http://www.freesource.info.wiki>

3. <http://cnews.ru>

4. Материалы IV конференции «СПО в высшей школе», 2009 год.

УДК 004.9+004.4

ПОСТРОЕНИЕ ЕДИНОЙ ОБРАЗОВАТЕЛЬНОЙ СРЕДЫ ТЕХНИКУМА НА ОСНОВЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

С.А. Ильина, Р.Ю. Соловьев

ГОУ СПО «Вятско-Полянский механический техникум»

г. Вятские поляны, Россия

E-mail: vpmt@bk.ru

Аннотация

В статье приводится обоснование создания единой образовательной среды учебного заведения. Описывается программное обеспечение, используемое для создания элементов среды.

Ключевые слова

Единая образовательная среда, свободное ПО, Drupal, электронная библиотека, Greenstone, дистанционное обучение, Moodle.

В настоящее время Россия переживает становление нового типа общественно-экономической формации – постиндустриального информационного общества. Современное информационное общество предъявляет новые требования к качеству профессиональной подготовки, требует развития новых современных базовых знаний, умений, навыков и компетенций. Составной частью компетенции специалиста является грамотное получение и использование информации в профессиональной деятельности. Проблема качества образования и формирования профессионально-информационной компетентности студентов на сегодняшний день наиболее актуальна в сфере образования.

Содержание образования должно постоянно корректироваться с учётом самых последних достижений в соответствующей отрасли. Оперативная корректировка обучающего материала возможна при наличии в учебном заведении единой образовательной информационной среды и использовании современных информационно-коммуникационных технологий.

В настоящее время в ГОУ СПО «Вятско-Полянский механический техникум» идёт создание единой информационной среды техникума на основе свободного программного обеспечения (рисунк 1).

Единая информационная среда

Сайт техникума www.vpmt.ru	Дистанционный портал www.vpmt.ru/ moodle	Электронная библиотека www.moodle.vpmt.ru
--	--	---

Рис. 1. Схема единой информационной среды

Вопрос применения свободного программного обеспечения в нашем учебном заведении встал, когда администрацией техникума была поставлена задача создания сайта и развёртывания Web-сервера.

С начала 2000-х годов свободное ПО (или «ПО с открытым кодом») постепенно завоёвывало внимание представителей крупного бизнеса и государственных учреждений различных стран мира. С каждым годом свободные программы проникают в новые секторы рынка, улучшаются их качественные характеристики, растёт их признание в профессиональном сообществе — все эти факторы не могли не привести к тому, что свободное ПО стало рассматриваться многими правительствами в качестве альтернативы проприетарному ПО.

Свободное ПО сегодня является неотъемлемой составляющей рынка программного обеспечения, и понимание масштабов распространения этого явления, его особенностей, преимуществ и недостатков является залогом успешного принятия решений по информатизации любого уровня — от малого предприятия до государства [8].

Проанализировав большое количество материалов Интернета, было принято решение в качестве операционной системы для Web-сервера использовать ОС Linux. Однако администрацией на реализацию был выделен короткий период времени, а опыта в работе с ОС Linux не было, поэтому была установлена проприетарная операционная система Windows Server 2003 Enterprise Edition. В качестве Web-сервера установлена связка HTTP-сервера Apache, языка сценариев общего назначения PHP и сервера баз данных MySQL.

В качестве системы управления сайтом была выбрана система Drupal, являющейся программным обеспечением с открытыми исходными текстами, защищёнными общественной лицензией GNU. Посредством этой системы был построен сайт техникума. Интерфейс пользователя обеспечивает управление сайтом без знаний HTML, Web-вёрстки и языков программирования, что облегчает заполнение новостного контента сайта [7].

В результате проделанной работы был развернут официальный сайт техникума (рис. 2).



Рис. 2. Главная страница сайта

В 2007 году на основе СДО MOODLE в рамках работы экспериментальной площадки ФИРО по теме «Создание и реализация модели дистанционного обучения в учреждении СПО» был создан Дистанционный портал техникума (рис. 3).

После изучения нескольких систем дистанционного обучения была выбрана СДО Moodle. Основными решающими факторами являлись:

1. СДО Moodle является программным продуктом с открытым кодом.
- Это значит, что:
- возможно бесплатное использование;
 - исправление ошибок происходит быстрее, чем в коммерческих продуктах;
 - возможно создание собственных модулей;
 - существует сообщество пользователей системы

(www.moodle.org).

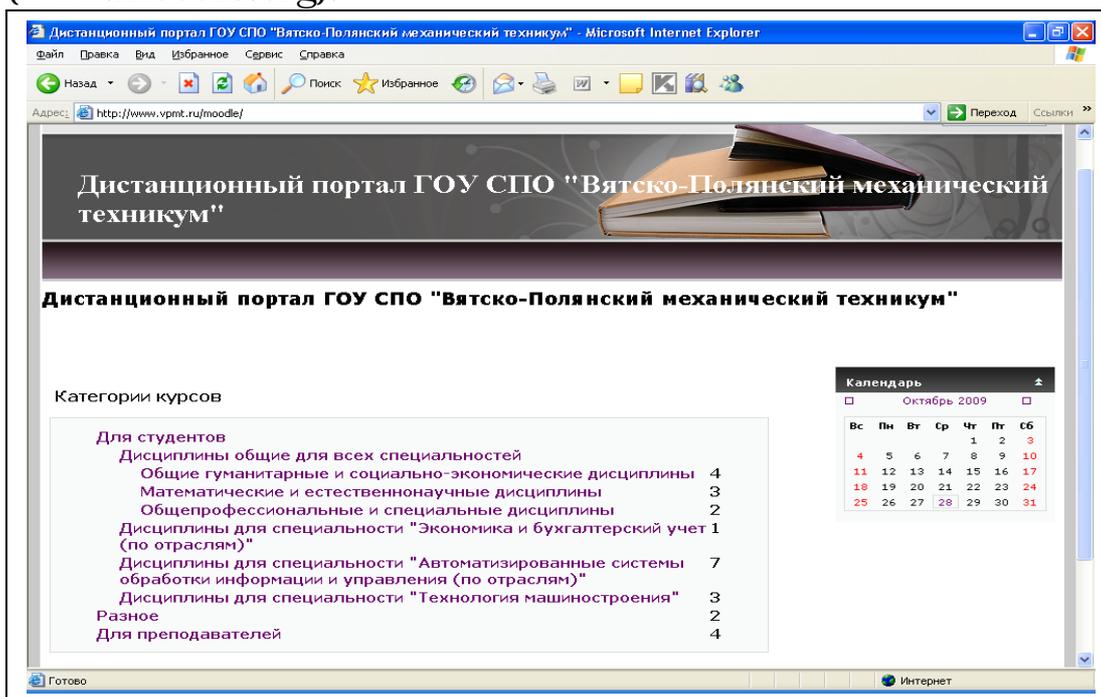


Рис. 3. Главная страница Дистанционного портала

2. Система обладает русскоязычным, интуитивно понятным интерфейсом, что позволяет работать с оболочкой преподавателям, не имеющим специальных знаний в области ИКТ.

СДО Moodle содержит встроенный WYSIWYG HTML редактор, позволяет вводить формулы в формате TeX, создавать глоссарий, использовать различные форматы видео и аудио, предоставляет преподавателю широкий выбор средств обучения (лекции, семинары и т.п.), средств проверки знаний и контроля успеваемости, общения между студентами и преподавателями (электронная почта, обмен файлами, чат), организации групповой работы (форум, чат, Wiki) [3].

Возможности СДО Moodle позволяют студентам создавать новый образовательный ресурс по теме, предложенной преподавателем, используя такие инструменты, как глоссарий или Wiki.

Одновременно с созданием дистанционного портала шла разработка электронной библиотеки техникума.

Организация доступа к источникам информации стала одной из важнейших задач информационного обеспечения науки и образования.

Электронные библиотеки дают возможность преподавателям создавать коллекции специально для учащихся, с которыми они работают, и включать в них разнообразную информацию и позволяют использовать различные средства распростране-

ния (например, на CD-ROM/DVD, локальные сайты учебных заведений).

Электронная библиотека — это не аналог Интернета. В библиотеку отбирается информация по определённым критериям. В каком-то смысле библиотеки представляют собой управляемые тематические коллекции.

В ГОУ СПО «Вятско Полянский механический техникум» для создания электронной библиотеки используется программное обеспечение «Greenstone».

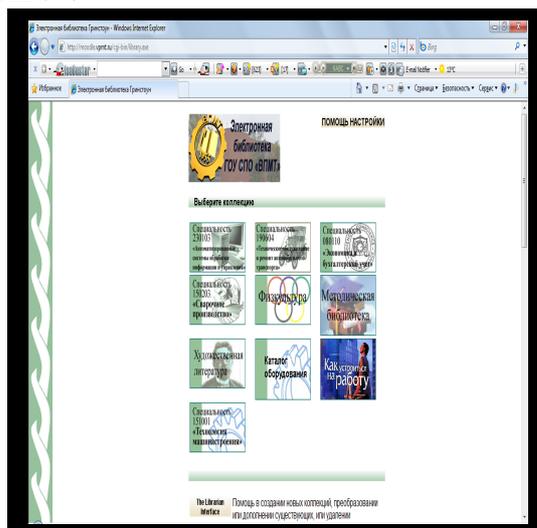


Рис. 4. Главная страница электронной библиотеки

Программа Greenstone — это система для создания и просмотра коллекций, состоящих из многих миллионов документов, включающих тексты, рисунки, аудио- и видеофайлы. Программное обеспечение Greenstone разработано в Новозеландском университете Вайкато, в рамках Проекта создания цифровых библиотек в сотрудничестве с ЮНЕСКО и НПО Human Info. Исходный продукт — это свободно распространяемое программное обеспечение, которое можно получить по адресу <http://greenstone.org>. Использование данного продукта оговаривается в свободном доступе GNU.

Типичная цифровая библиотека, созданная с помощью Greenstone, содержит в себе множество коллекций, организованных по отдельности, хотя они имеют много сходства друг с другом. Легко поддерживаемые, эти коллекции могут быть дополнены и перестроены автоматически. Коллекции могут содержать текст, рисунки, аудио- и видеоматериалы. Не текстовый материал связан с текстовым файлом либо через ссылку, либо имеет текстовое описание (такое как заголовок) для облегчения возможности полнотекстового поиска и просмотра.

Коллекции доступны в Интернете или публикуются на самоустанавливающихся компакт-дисках в таком же виде, как в Интернете. Для уменьшения объёма текста и индексов используется сжатие.

В Greenstone структура каждой коллекции определяется в процессе её создания. Она включает определение формата используемых документов, их вывод на экран, источник метаданных: какие предметные показатели должны быть включены, какие следует предоставить полнотекстовые индексы, как должны отображаться результаты поиска [1].

На сегодняшний день в техникуме планируется постепенный переход на свободное программное обеспечение. Первоочередной задачей является:

1. Перевод на свободное программное обеспечение Web – сервера.
2. Внедрение на вновь приобретаемых персональных компьютерах OpenOffice.

Выполнение данных задач позволит в будущем постепенно перевести большинство компьютерных классов и рабочих мест преподавателей на СПО. Последним структурным подразделением, которое будет переведено на СПО, является бухгалтерия техникума, это связано с очень сложной миграцией части программного обеспечения, используемого в деятельности бухгалтерии.

Используемые источники:

1. Виттен И., Бейнбридж Д., Николс Д. Электронные библиотеки в образовании: специализированный учебный курс/авторизованный перевод с английского. – М: «Обучение-Сервис», 2006.
2. Богданов А.В., Сютюренко О.В., Хохлов Ю.Е. Программа «Российские электронные библиотеки» // Научный электронный журнал «Электронные библиотеки». – 1998. - Том 1. – Вып. 1 [http://www.elbib.ru/content/journal/1998/199801/BHS/bhs.ru.html]
3. www.moodle.org
4. www.infoco.ru
5. http://www.curator.ru/resource.html
6. http://dstudy.ru/articles/
7. www.drupal.ru
8. http://www.ifap.ru/library/book370.pdf. Свободное программное обеспечение в государственном секторе. Сборник материалов. — М.: INFO-FOSS.RU, 2007. - 112 с.

ДИНАМИЧЕСКОЕ WEB-ПРИЛОЖЕНИЕ КАК СРЕДСТВО РАЗВИТИЯ МЫСЛИТЕЛЬНОЙ ДЕЯТЕЛЬНОСТИ ПОДРОСТКОВ ПРИ ОБУЧЕНИИ ГЕОМЕТРИИ

М.В. Подаев

ГОУ ВПО «Елецкий государственный университет
им. И.А. Бунина»

г. Елец, Россия

E-mail: podaev86@rambler.ru

Аннотация

Ситуация в геометрическом образовании сегодня в стране довольно острая, дети не знают и не любят геометрию, у них не развито логическое и пространственное мышление. В то же время сегодня есть возможности применять инновационные средства обучения, к которым относится разработанное нами динамическое Web-приложение, имитирующее построения циркулем и линейкой.

Ключевые слова

Геометрия, мыслительная деятельность, динамическое Web-приложение, мыслительные операции.

В настоящее время уровень геометрического образования школьников катастрофически падает, в основной школе намечилась тенденция замены систематического курса геометрии курсом наглядной геометрии, все чаще звучит мнение о кризисе геометрического образования. В то же время «...именно в геометрии можно собрать воедино все то, что мы хотим дать детям из математики, настолько она богата содержанием. ...Именно геометрия – главная особенность нашего образования, если угодно – наше «тайное оружие», благодаря которому наши специалисты ценятся в мире. И сохранить её необходимо во что бы то ни стало» [2].

В связи с этим возникает насущная потребность в качественной передаче учащимся геометрических знаний, умений и навыков, в развитии у них логического и образного мышления.

Как нельзя лучше для реализации поставленных целей под-

ходят геометрические задачи на построение циркулем и линейкой. Они нестандартны, их невозможно решить по шаблону. В процессе решения приходится прибегать и к аналитико-синтетическим операциям, и к пространственным представлениям. Однако учителя при изучении данной темы сталкиваются с трудностями, прежде всего организационного плана: черчение с помощью циркуля даже ученикам старших классов даётся довольно трудно, а в более ранних классах и вовсе не представляется возможным.

В то же время необходимость наличия данной темы при изучении элементов геометрии в 6 – 7 классе очевидна. Именно этот возраст является сензитивным для развития пространственных представлений, именно здесь происходит переход от наглядно-образного к абстрактно-логическому способу изложения материала. Именно в младшем подростковом возрасте очень важно *начать* развивать теоретическое мышление, обращаясь к аналитико-синтетическим операциям.

Сегодня же, в условиях перехода к информационному обществу, заметным явлением стал компьютер. И здесь особо следует сказать о взаимоотношениях между геометрией и компьютером. По мнению И.Ф. Шарыгина, с одной стороны, геометрический тип рассуждений наименее поддаётся компьютеризации. Геометрия остаётся одной из немногих сфер интеллектуальной деятельности, где человек ещё не проиграл соревнование компьютеру. А с другой — компьютер является очень полезным инструментом в геометрических исследованиях. С его помощью можно экспериментально обнаруживать новые интересные геометрические факты [3].

Использование компьютера как инструмента учебной деятельности даёт возможность переосмыслить организационные подходы к изучению многих вопросов геометрии, приблизить процесс обучения к реальному процессу познания [1].

И здесь на передний план выходит вопрос об использовании интерактивных программных средств. В данном случае речь идет о разработанном автором динамическом Web-приложении, которое обладает необходимым инструментарием для имитации построений циркулем и линейкой (рис. 1).

Приложение написано с использованием языков php, javascript, xml, MySql. Графическая составляющая реализована с применением технологии SVG (от англ. Scalable Vector Graphics — масштабируемая векторная графика). Асинхронные запросы

(AJAX) дают пользователю возможность в реальном времени проводить линии, отрезки, окружности (зная центр и радиус – в качестве расстояния между любыми двумя точками), параллельные прямые, находить точки пересечения прямых и окружностей.

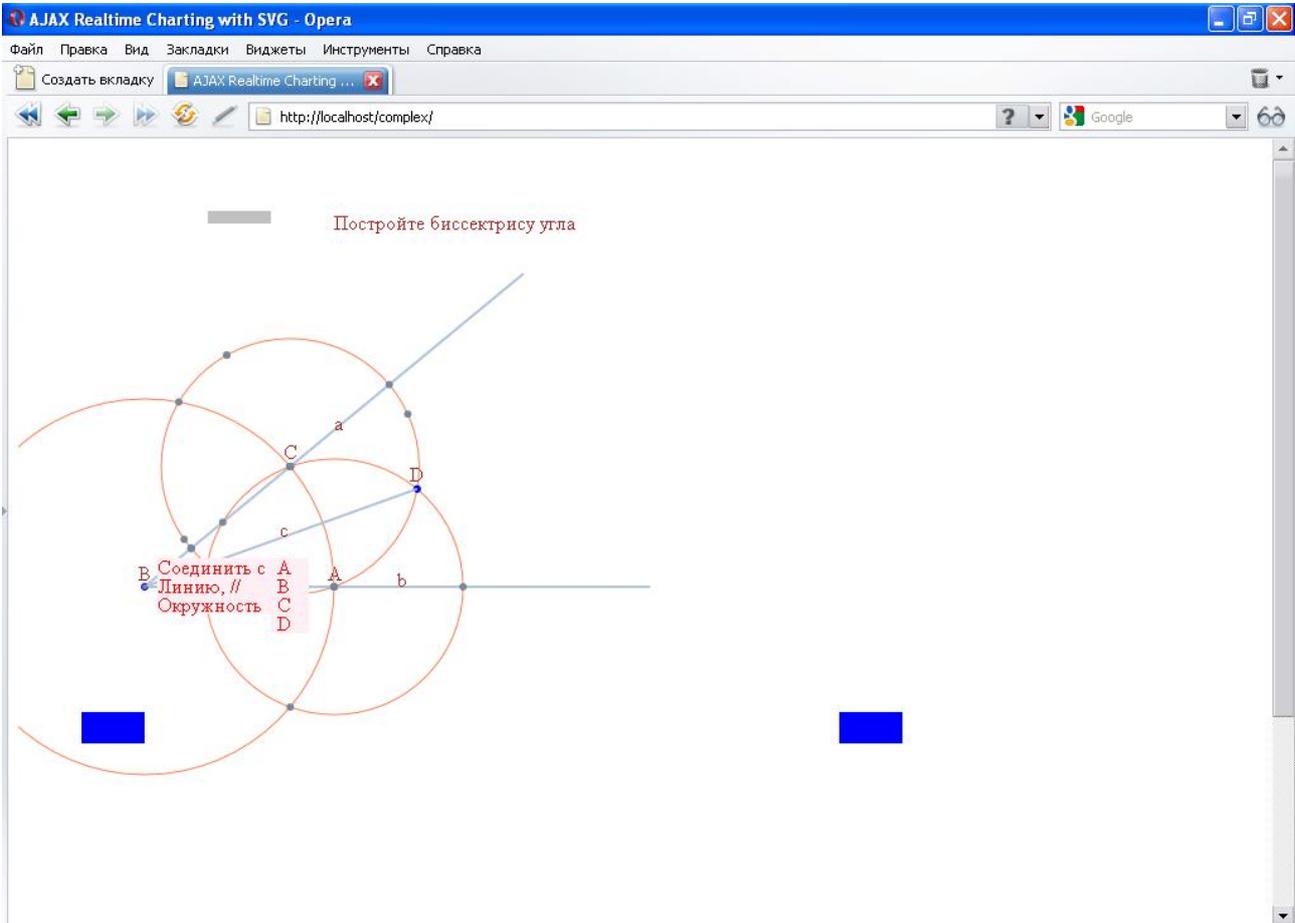


Рис. 1. Динамическое Web-приложение, имитирующее построения циркулем и линейкой

Практика использования данного приложения на протяжении 2008-2009 уч. г. в МОУ СОШ №15 г. Ельца в 7 «а» классе показала положительные результаты. Дружественный интерфейс программы и простота в использовании способствовали тому, что ребята с удовольствием решали различные задачи на построение.

В заключение необходимо сказать, что перспективы применения интерактивных программных средств при обучении геометрии связаны, прежде всего, с эффективной реализацией дидактических принципов наглядности и доступности в обучении, которые, наряду с другими принципами дидактики, являются одними из ведущих факторов обучения и развития. И их использование будет способствовать развитию мыслительной деятельности подростков, а тем самым благоприятно влиять на состояние геометрического образования в стране.

Используемые источники:

1. Красникова Л. Дидактические возможности символьной системы Mathematica в процессе обучения геометрии в вузе // Материалы международной научно-практической конференции “Информатизация образования”. 2005. - С. 116-118.
2. Рыжик В. Время разбрасывать камни? // Математика, приложение к “ПС”, № 11, 2005, с 15.
3. Шарыгин И. Нужна ли школе XXI века геометрия? // “Математика в школе”. 2004. №4. - С. 72-78.

УДК 004.9+004.4

МЕТОДИКА ОРГАНИЗАЦИИ АВТОМАТИЗАЦИИ РАСЧЁТОВ В ЭЛЕКТРОННОЙ ТАБЛИЦЕ OPENOFFICE.ORG CALC

Т.Н. Губина

ГОУ ВПО «Елецкий государственный университет

им. И.А. Бунина»

г. Елец, Россия

E-mail: gubina-tn@yandex.ru

Аннотация

В статье предлагается один из методов знакомства учащихся с процессом автоматизации вычислений в офисном пакете OpenOffice.org на примере входящей в его состав электронной таблицы.

Ключевые слова

Электронная таблица, автоматизация расчётов, поиск решения, макрос, язык программирования, информационно-технологическая культура, обучение.

Модуль OpenOffice.org Calc, входящий в состав офисного пакета OpenOffice.org Calc, относится к классу программ, называемых электронными таблицами.

К настоящему времени имеется большое количество методических разработок, направленных на обучение приемам работы с электронными таблицами.

В силу того, что в последнее десятилетие наибольшую популярность, а следовательно, и распространённость получила электронная таблица Excel, входящая в состав офисного пакета Micro-

soft Office, то в большей степени методические разработки ориентированы именно на эту электронную таблицу.

Но в последние годы в связи с вступлением в силу закона об авторском праве, а также разработкой отечественной операционной системы на базе Linux все чаще используется альтернативный офисный пакет OpenOffice.org.

Мы работаем с пакетом OpenOffice.org уже 3 года и организуем учебный процесс в вузе с его использованием в рамках дисциплин «Программное обеспечение ЭВМ», «Информатика», «Информационные и коммуникационные технологии в образовании», «Практикум на ЭВМ», «Информационные системы» и др. на разных специальностях. Кроме того, организуя работу Федеральной пилотной площадки по внедрению Пакета свободного программного обеспечения на базе МОУГ № 11 г. Ельца, мы проводили занятия с учениками 9-11 классов по формированию информационно-технологической культуры у учеников при работе со свободно распространяемым офисным пакетом.

Опыт внедрения OpenOffice.org в учебный процесс показал, что этот программный продукт является быстро развивающимся офисным приложением, и те его функциональные возможности, которые были недоступны в версии OpenOffice.org 2.2 (дата выхода 29 марта 2007 г.), в текущей версии OpenOffice.org 3.2 (дата выхода 11 февраля 2010 г.) уже доступны.

Быстрые темпы развития офисного пакета OpenOffice.org, отсутствие хорошей методической литературы по его использованию, а также его достаточно быстрое распространение среди пользователей вызвали интерес у учителей, преподавателей, научных работников к этому программному продукту и активизировали желание опубликовать учебные и учебно-методические пособия с рекомендациями по организации использования OpenOffice.org как в сфере образования, так и в сфере коммерции.

В настоящей статье предлагается один из вариантов организации учебного процесса по изучению электронной таблицы OpenOffice.org Calc в общеобразовательных учреждениях.

В методическом письме «О преподавании учебного предмета «Информатика и ИКТ» и информационных технологий в рамках других предметов в условиях введения федерального компонента государственного стандарта общего образования» [1] выделяется как одна из задач преподавания информатики и ИКТ — формиро-

вание методологии использования основных автоматизированных информационных систем в решении конкретных задач, связанных с анализом и представлением основных информационных процессов, в том числе автоматизированных информационных систем (АИС) обработки информации (автоматизированное рабочее место, офисные пакеты). «Информационные технологии, которые изучаются в базовом уровне – это, прежде всего, автоматизированные информационные системы. Это связано с тем, что возможности информационных систем и технологий широко используются в производственной, управленческой и финансовой деятельности» [1].

Как показывает практика преподавания информатики в школе, при изучении электронных таблиц в большей мере уделяется внимание только методам заполнения электронных таблиц данными, выполнению расчётов по формулам, построению различных диаграмм по полученным данным. В лучшем случае учащиеся знакомятся с возможностью хранения баз данных в электронной таблице, осуществлением выборки и сортировки данных, созданием сводных таблиц.

Однако возможности электронных таблиц настолько широки, что, с нашей точки зрения, с ними учеников нужно обязательно знакомить. Что мы и делали на дополнительных занятиях с учениками по предмету «Информационные технологии на базе свободного программного обеспечения».

Первая из таких возможностей — это анализ данных. Мы разбирали с учениками процесс нахождения исходных данных, которые при подстановке в формулы дают необходимое значение в ячейке результата.

Учащимся предлагалась задача: *Вы хотите положить деньги в банк под 8 % и получить ровно 2000 руб. по истечении года. Как определить сумму вклада?*

Учащиеся предлагают решение этой задачи с математической точки зрения. Далее мы говорим о том, что в OpenOffice.org Calc для решения подобных задач предусмотрены две команды:

1. *Подбор параметра* – определяет значение одной входной ячейки, которое требуется для получения желаемого результата в зависимой ячейке (ячейке результата).

2. *Поиск решения* — определяет значения в нескольких входных ячейках, которые требуются для получения желаемого результата. Более того, можно накладывать ограничения на вход-

ные данные, поэтому здесь можно получить решение (если оно существует) многих практических задач.

Реализация решения задачи в OpenOffice.org Calc:

1. Создаём новый документ, первый лист документа переименуем «Задача 1».

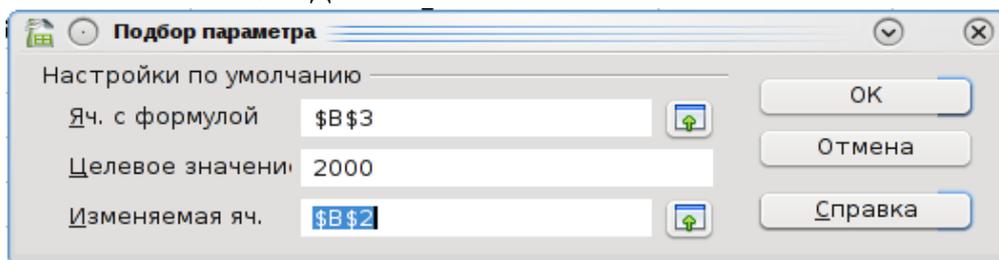
2. Заполняем ячейки таблицы A1, A2, A3 и B1 вручную. При этом для ячейки B3 выбираем формат данных *Процентный*. После чего активизируем ячейку B3, введём формулу: $(1+B1)*B2$ (рис.1).

	A	B
1	Ставка	8,00%
2	Сумма вклада	
3	Конечная сумма	$=(1+B1)*B2$

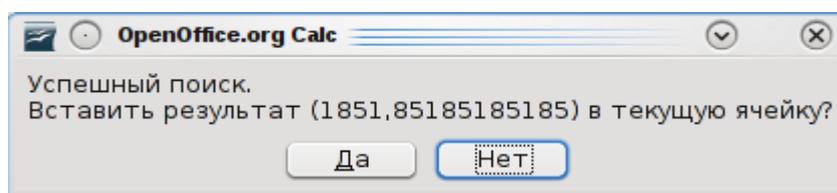
Если требуется, чтобы в ячейки выводились не значения, а формулы, то следует воспользоваться меню *Сервис* → *Параметры*, далее в левом списке диалогового окна выбрать OpenOffice.org Calc, подпункт *Вид*.

Рис.1. Отображение содержимого ячеек в режиме формул

3. При выделенной ячейке B3 выполним команду *Сервис* → *Подбор параметра*. В открывшемся диалоговом окне указываем значение необходимой конечной суммы и ссылку на ячейку с искомым значением вклада:



4. Нажимаем на кнопку *ОК* диалогового окна и откроется окно с найденным решением.



5. Выбираем *Да* и получаем решение нашей задачи.

Команда *Подбор параметра* является удобной для решения задач поиска определённого значения, зависящего от одного неизвестного параметра. Если требуется решить более сложную задачу, то в этом случае лучше воспользоваться командой *Поиск решения*. Также эта команда применяется в том случае, если нам нужен не конкретный результат, а минимально или максимально

возможный. Например, как минимизировать затраты на содержание персонала или максимизировать доходность предприятия?

Для демонстрации возможностей поиска решения учащимся предлагалась задача: Фирма занимается производством нескольких видов конфет "А", "В" и "С". Известно, что реализация 10-килограмм конфет "А" даёт прибыль 900 руб., "В" - 1000 руб. и "С" - 1600 руб. Конфеты можно производить в любых количествах (сбыт обеспечен), но запасы сырья ограничены. Необходимо определить, каких конфет и сколько десятков килограмм необходимо произвести, чтобы общая прибыль от реализации была максимальной [5].

Нормы расхода сырья на производство 10 кг конфет каждого вида приведены ниже.

Сырье	Нормы расхода сырья			Запас сырья
	А	В	С	
Какао	13	16	10	380
Сахар	8	5	10	190
Наполнитель	6	4	4	170
Прибыль	900 руб.	1000 руб.	1600 руб.	

Реализация решения задачи в OpenOffice.org Calc:

1. Создаём новый документ, первый лист документа переименуем «Задача 2».

2. Заполняем ячейки таблицы А1:А4, В1, В5, С1, В7, А8:С8 вручную. В ячейки С2:С4, С5, А9:С9 вводим формулы (см. рис.2).

	А	В	С
1	Наименование	Количество	Прибыль
2	А		=900*В2
3	В		=1000*В3
4	С		=1600*В4
5		Стоимость конфет:	=SUM(С2:С4)
6			
7		Расход сырья	
8	Какао	Сахар	Наполнитель
9	=13*В2+16*В3+10*В4	=8*В2+5*В3+10*В4	=6*В2+4*В3+4*В4

Рис.2. Отображение содержимого ячеек в режиме формул

3. Далее выбираем пункт меню *Сервис* → *Поиск решения*. В открывшемся диалоговом окне задаём целевую ячейку С5 (в неё будет выведено значение максимальной прибыли), диапазон изменяемых ячеек — В2:В4 (количество конфет каждого вида в кг), задаём следующие ограничительные условия (см. рис.3):

по запасу сырья: $\$A\$9 \leq 380$, $\$B\$9 \leq 190$, $\$C\$9 \leq 170$
 по количеству конфет: $\$B\$2 > 0$, $\$B\$3 > 0$, $\$B\$4 > 0$.

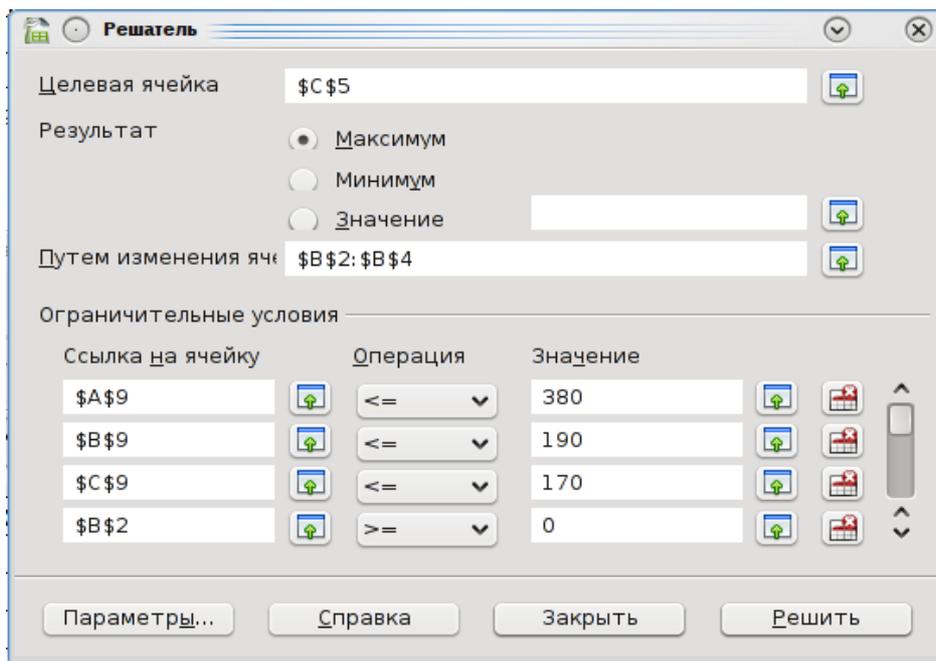
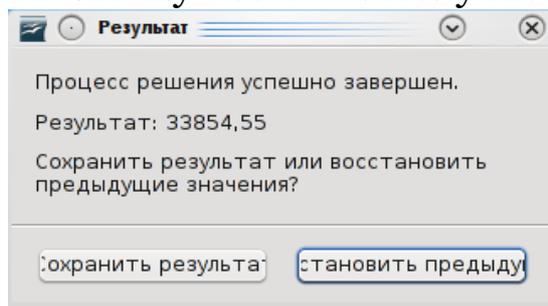


Рис.3. Вид диалогового окна *Решатель* после заполнения

4. Нажимаем на кнопку *Решить*. Получаем решение:



5. Нажимаем на кнопку *Сохранить результат*:

	А	В	С
1	Наименование	Количество	Прибыль
2	А	0	0
3	В	17,27	17272,73
4	С	10,36	16581,82
5		Стоимость конфет:	33854,55
6			
7		Расход сырья	
8	Какао	Сахар	Наполнитель
9	380	190	110,55

Таким образом, из полученного решения видно, что оптимальный план выпуска предусматривает изготовление 17,27 кг конфет "В" и 10,36 кг конфет "С". Конфеты "А" производить не стоит. Полученная при этом прибыль составит 33854,55 руб.

Для автоматизации действий в OpenOffice.org также исполь-

зуются макросы, которые удобны в тех случаях, когда требуется автоматизировать действия. «Макрос (от англ. macros, мн.ч. от macro) — программный объект, при обработке «развёртывающийся» в последовательность действий или команд» [2]. При обработке макроса автоматически выполняется заданная для каждого макроса последовательность действий — нажатия на клавиши, выбор пунктов меню и т. д.

Создание макросов в офисном пакете OpenOffice.org можно выполнять на языках Basic, Python, JavaScript, Shell. Но учитывая, что во многих школах на начальном уровне знакомят с языком Basic, мы при проведении занятий по работе с макросами использовали именно язык OpenOffice.org Basic.

Начали мы рассмотрение макросов с самого простого: автоматическое создание макроса. Действительно, ведь чтобы создать свой макрос в OpenOffice.org, не обязательно быть программистом. В OpenOffice.org имеется встроенная программа регистратор, которая может запоминать все действия пользователя с документом и записывать их последовательно в специальный текстовый модуль, называемый Макрос.

Ученикам предлагалось следующее задание: *пусть нам периодически требуется заполнять ячейки электронной таблицы одними и теми же данными, например, список учеников класса. Конечно, можно выполнить заполнение требуемых ячеек данными и сохранить файл на диске. Но мы с вами создадим для этого макрос и сохраним его.*

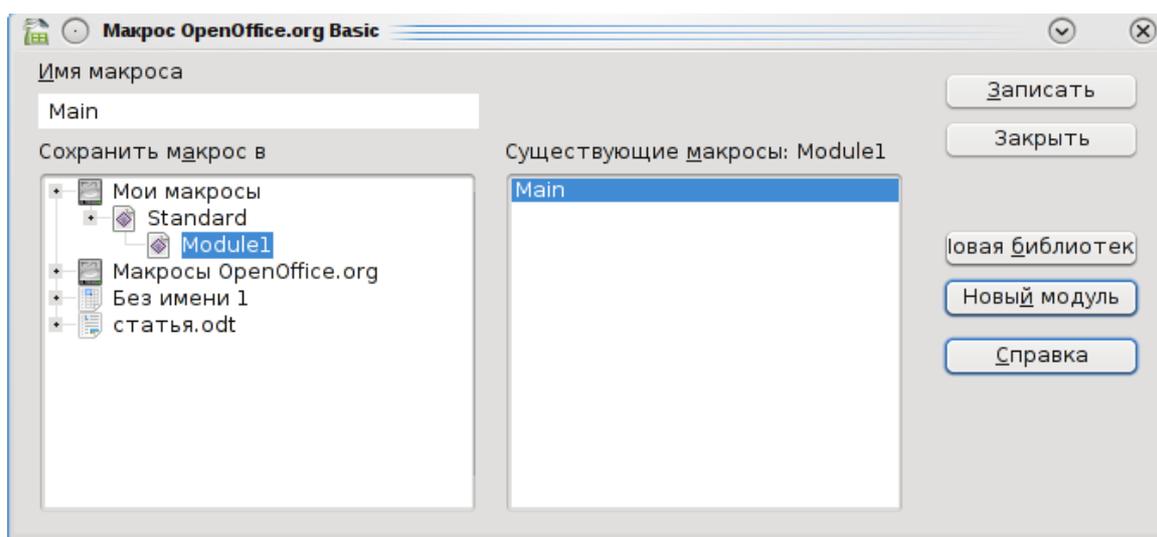
Технология создания макроса «Список»:

1. Открыть документ, для которого нужно записать макрос. Выбрать команду *Сервис → Макросы → Записать макрос*. Откроется диалоговое окно *Запись макроса* с одной только кнопкой «*Завершить запись*».

2. Теперь мы вносим фамилию, имя учеников в столбец А.

	А
1	Иванов Иван
2	<u>Петров Петр</u>
3	Сергеев Сергей
4	Поляков <u>Артем</u>
5	<u>Иванова Елена</u>
6	<u>Белова Анна</u>

3. Нажимаем кнопку «*Завершить запись*». Откроется диалоговое окно *Макрос*, в котором можно сохранить и выполнить макрос.



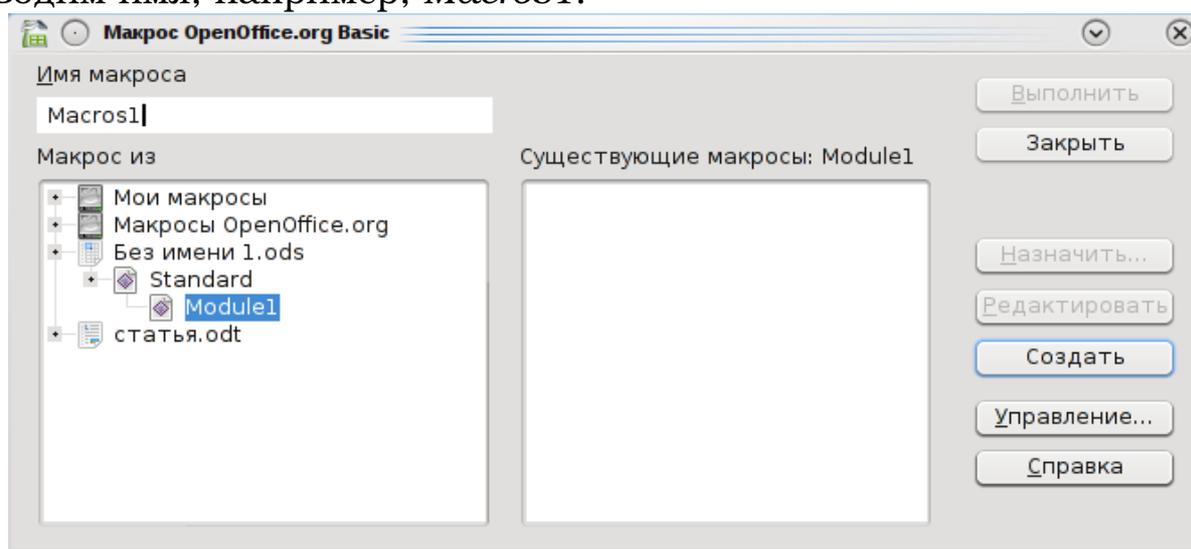
4. Теперь очистим содержимое документа. Установите курсор в ячейку A1 и выполните команду *Сервис → Макросы → Выполнить макрос*. В открывшемся окне *Выбор макроса* найдите ваш макрос и нажмите кнопку *Выполнить*.

Следующим шагом мы переходим к «ручному» созданию макроса. Возможности создания макроса были продемонстрированы учителем на простейшем примере: вывод окна сообщения «Ура! Заработало!».

Последовательность действий:

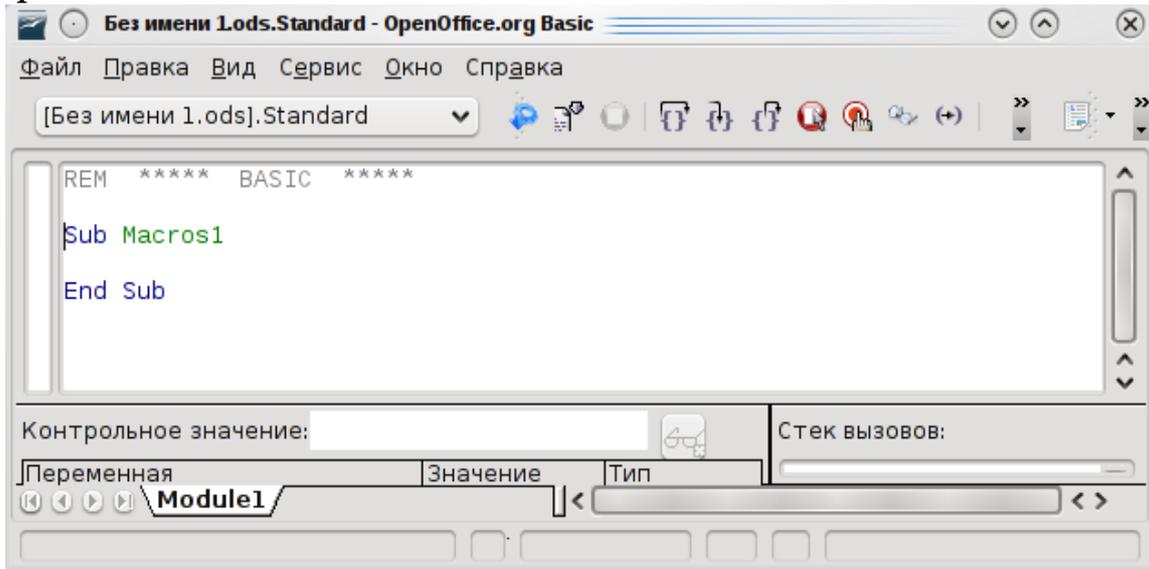
1. Создать новый документ «Без имени 1». Открыть редактор OOo Basic, используя меню *Сервис → Макросы → Управление макросами → OpenOffice.org Basic*.

2. С левой стороны окна диалога надо найти документ «Без имени 1». Выделяем его (см. рисунок ниже). В окне *Имя макроса* вводим имя, например, *Macros1*.



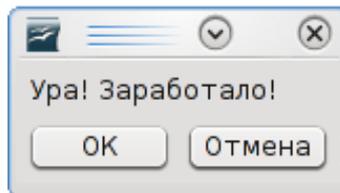
3. Щёлкаем по кнопке *Создать*. Откроется среда разработки

макросов.



Как видим, в окне редактора кода уже появилась заготовка под написание программного кода: подпрограмма-процедура с именем Main по умолчанию. Внутри процедуры вводим команду: Print "Ура! Заработало!"

4. На панели инструментов диалогового окна нажимаем на кнопку «Выполнить Basic». У нас появится окно сообщения:



5. Сохраните документ.

Далее мы переходим на более сложные задания, например: *выполнить построение графика функции*

$$y = \frac{1 + |0,2 + x|}{1 + x + x^2} + \cos(x) + \ln(x + 2) - \arctg(x^3 + 1) + e^{-x} - t g(x^{3,13}) + \sqrt{x}$$

на отрезке [0;0,45]. Вычисление значения функции оформить в виде подпрограммы-функции с именем FУ. Вывести таблицу вычисленных значений с шагом 0,1.

Реализация решения задачи в OpenOffice.org Calc:

1. Открываем новую электронную таблицу. Создадим макрос: функцию с именем FУ. Для этого повторяем последовательность шагов 1-3 из предыдущего примера.

2. В открывшемся редакторе прописываем подпрограмму-функцию (см. рис. 4).

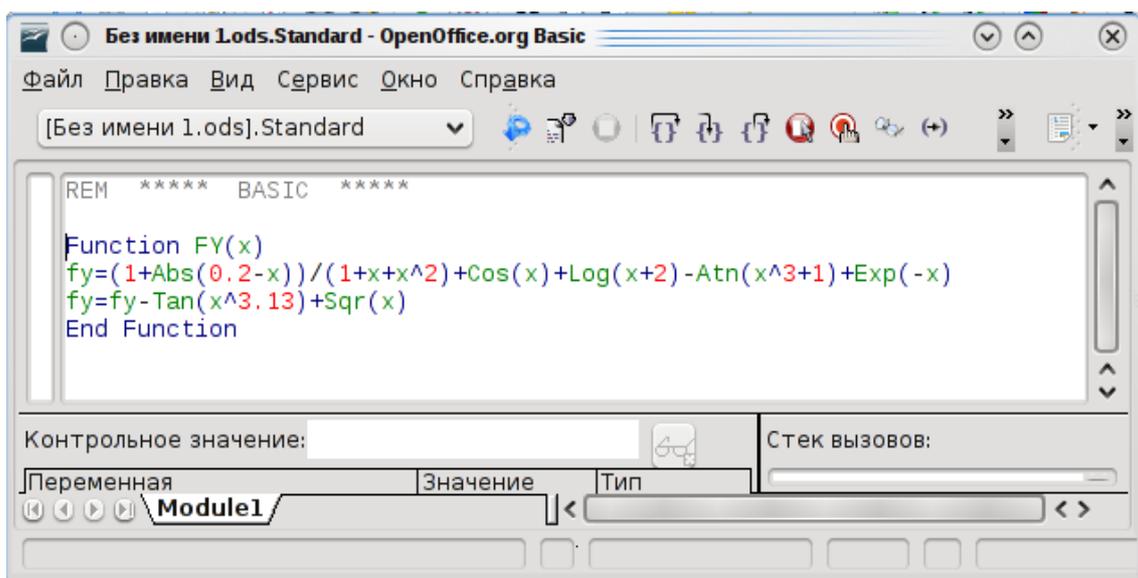


Рис.4. Создание подпрограммы-функции пользователя

3. Сохраняем и закрываем диалоговое окно, изображённое на рис.4.

4. Теперь подготовим таблицу. В первый столбец введём подписи к данным. Далее в первой строке заполним диапазон изменения независимой переменной X. Для этого введём вручную два первых значения переменной X, далее воспользуемся методом автоматического заполнения ячеек числовым рядом, соседние члены которого отличаются на постоянную величину.

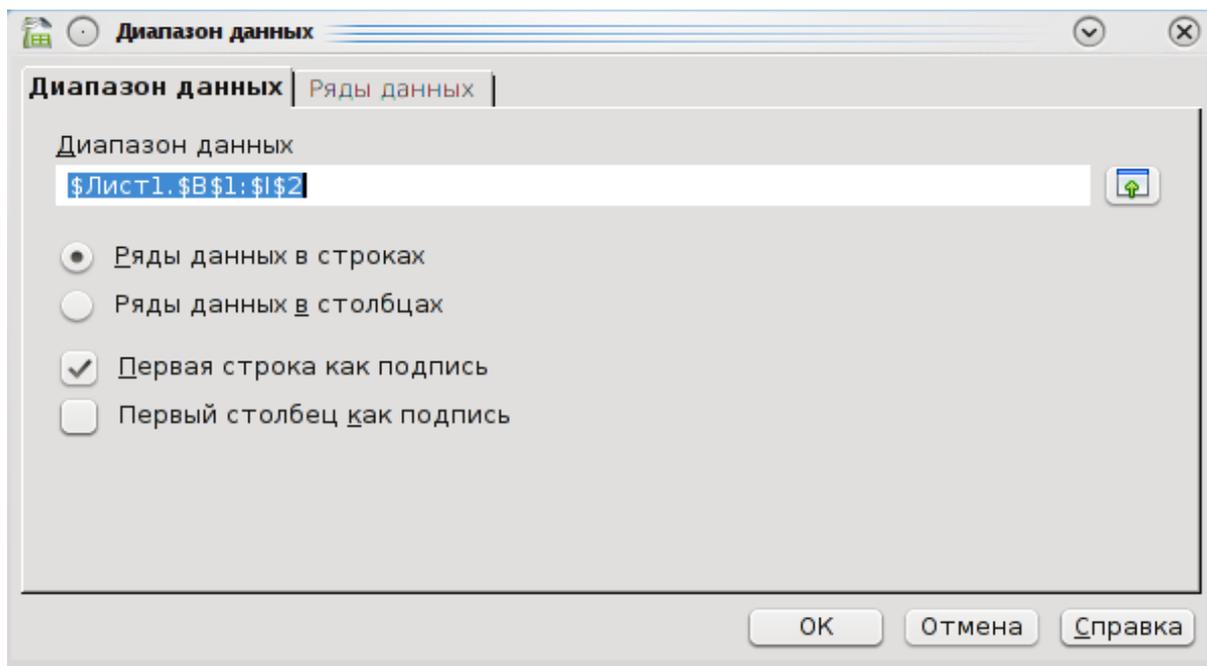
5. Далее введём формулу в ячейку B2: =FY(B1). Получим:

Теперь скопируем формулу в остальные ячейки таблицы:

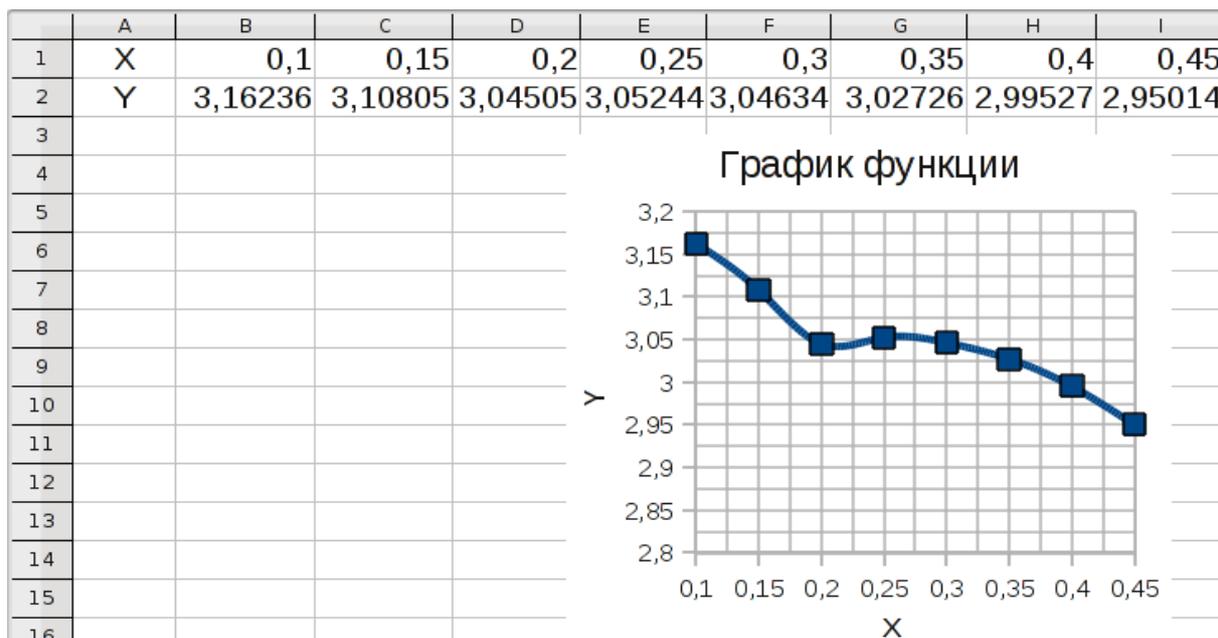
	A	B	C	D	E	F	G	H
1	X	0,1	0,15	0,2	0,25	0,3	0,35	0,4
2	Y	3,16236	3,10805	3,04505	3,05244	3,04634	3,02726	2,9952
3								

Для построения графика зависимости $y=y(x)$ выберем меню *Вставка* → *Диаграмма*. Выбираем тип диаграммы «Линии», «Линии и точки». Нажимаем кнопку «Далее» и в поле ввода «Диапазон данных» вводим диапазон \$B\$1:\$I\$2.

Отметим пункт «Ряды данных в строках» и включим пункты «Первая строка как надпись».



Нажмите кнопку «Далее» два раза. В заголовке ось X введите символ X, а ось Y введите символ Y, зададим заголовок диаграммы. В результате получим:



На следующем шаге знакомства с возможностями электронной таблицы OpenOffice.org Calc мы изучали способы добавления элементов управления в документ. Примеры носили разносторонний характер: использование элементов управления для быстрого ввода информации, для вывода полученного результата вычислений.

Рассмотрим один из примеров: *реализовать поиск решения полного квадратного уравнения с коэффициентами $a \neq 0, b, c$.*

Реализация решения задачи в OpenOffice.org Calc:

1. Создадим новый документ. На одном листе в документе разместим элементы управления: 6 меток, 3 текстовых поля и командную кнопку. Для этого мы воспользуемся панелью инструментов *Элементы управления*. Например, находим на панели инструментов элемент *Метка*, рисуем в рабочем листе прямоугольник методом *drag&drop*. Теперь по выделенной метке щёлкаем правой кнопкой мыши и в открывшемся контекстном меню выбираем подменю *Элемент управления*. У нас откроется окно его свойств. Задаём имя метке, надпись на метке, цвет фона и т.д. (рис. 5). Аналогичным образом поступаем и с другими элементами управления.

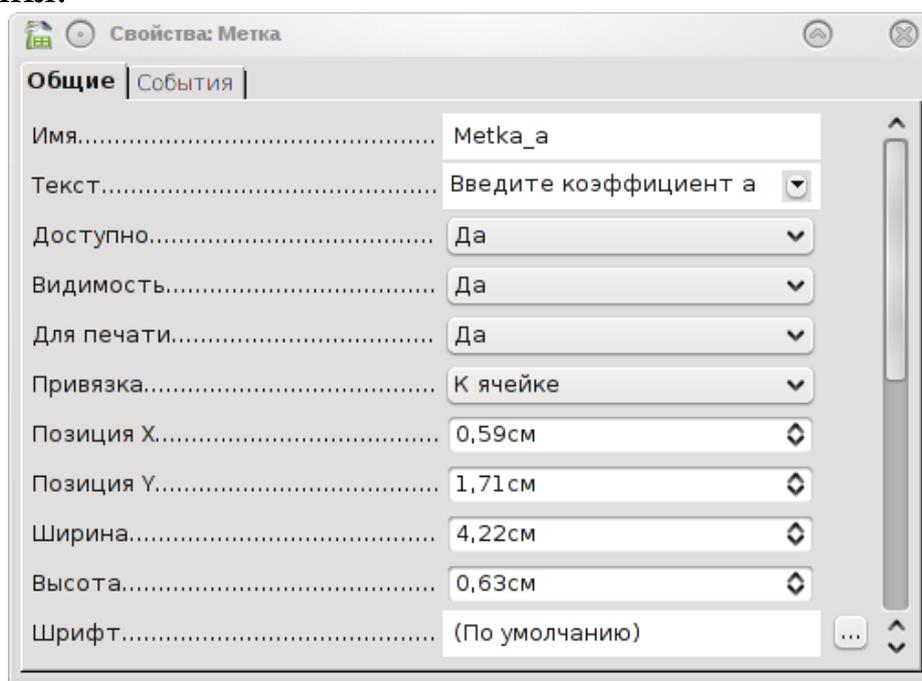


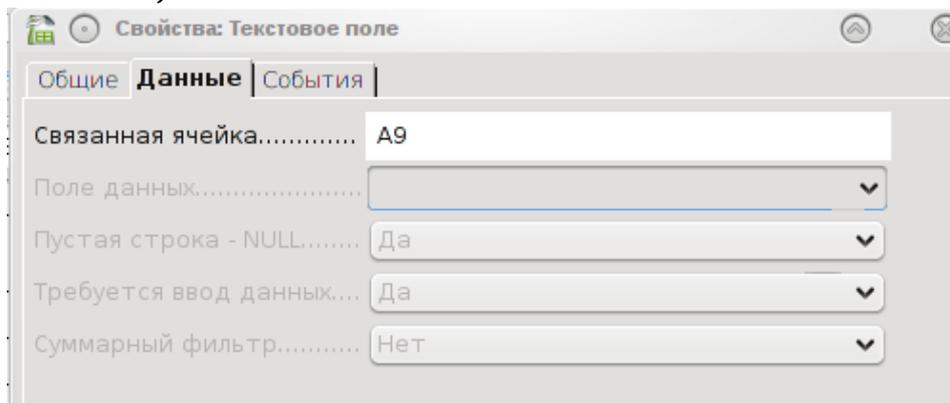
Рис. 5. Окно свойств элемента управления Метка

Ввод коэффициентов уравнения будем производить в текстовые поля, при нажатии на командную кнопку будет осуществляться вывод в одну ячейку значения дискриминанта уравнения, а в другую ячейку — решения уравнения (найденные значения корней или сообщение, что действительных корней нет).

2. После размещения объектов в документе, изменения цвета и размера шрифта надписей объектов получим:

	A	B	C	D	E
1	Решаем квадратное уравнение $ax^2+bx+c=0$				
2	Введите коэффициент a	<input type="text"/>		Дискриминант равен	
3					
4	Введите коэффициент b	<input type="text"/>		Ответ:	
5					
6	Введите коэффициент c	<input type="text"/>			
7					
8	Решить				
9					

3. Теперь для первого текстового поля в контекстном меню выбираем *Элемент управления*. На закладке *Данные* значение поля *Связанная ячейка* установить равным адресу ячейки, в которое будет передаваться значение коэффициента a (в нашем случае ячейка A9). Аналогично для двух других текстовых полей (ячейки B9 и C9).



4. Для командной кнопки мы разработаем макрос, который будет решать наше уравнение и выводить в документ найденное решение. Поэтому создаём макрос. Выбираем пункт меню *Сервис* → *Макросы* → *Управление макросами* → *OpenOffice.org Basic*. После этого в открывшемся диалоговом окне выбираем макрос из созданного нами документа (Без имени 1.ods), задаём имя макроса, например, *Main* и нажимаем на кнопку *Создать*. В окне разработки макроса прописываем код (см. рис.6).

В программном коде мы обращаемся к ячейкам A9, B9, C9 для считывания коэффициентов, введённых с клавиатуры. Это мы делаем с помощью команды `thiscomponent.sheets("sheet1").getcellbyposition(x,y).string`, где

- `thiscomponent` – указатель на абстрактный объект, который содержит в себе конкретный объект `sheets`;
- `sheet2` – имя этого конкретного объекта (в нашем случае

лист в документе);

- `getcellbyposition (x,y)` – это метод, применимый к объекту `sheets` (в нашем случае это ячейка, расположенная в столбце `x` и в строке `y`). Заметим, что нумерация строк и столбцов начинается с нуля;

- `String` — тип данных, расположенных в ячейке. В нашем случае это строковый тип, поскольку в ячейки мы размещаем информацию (число), введенное в текстовое поле.

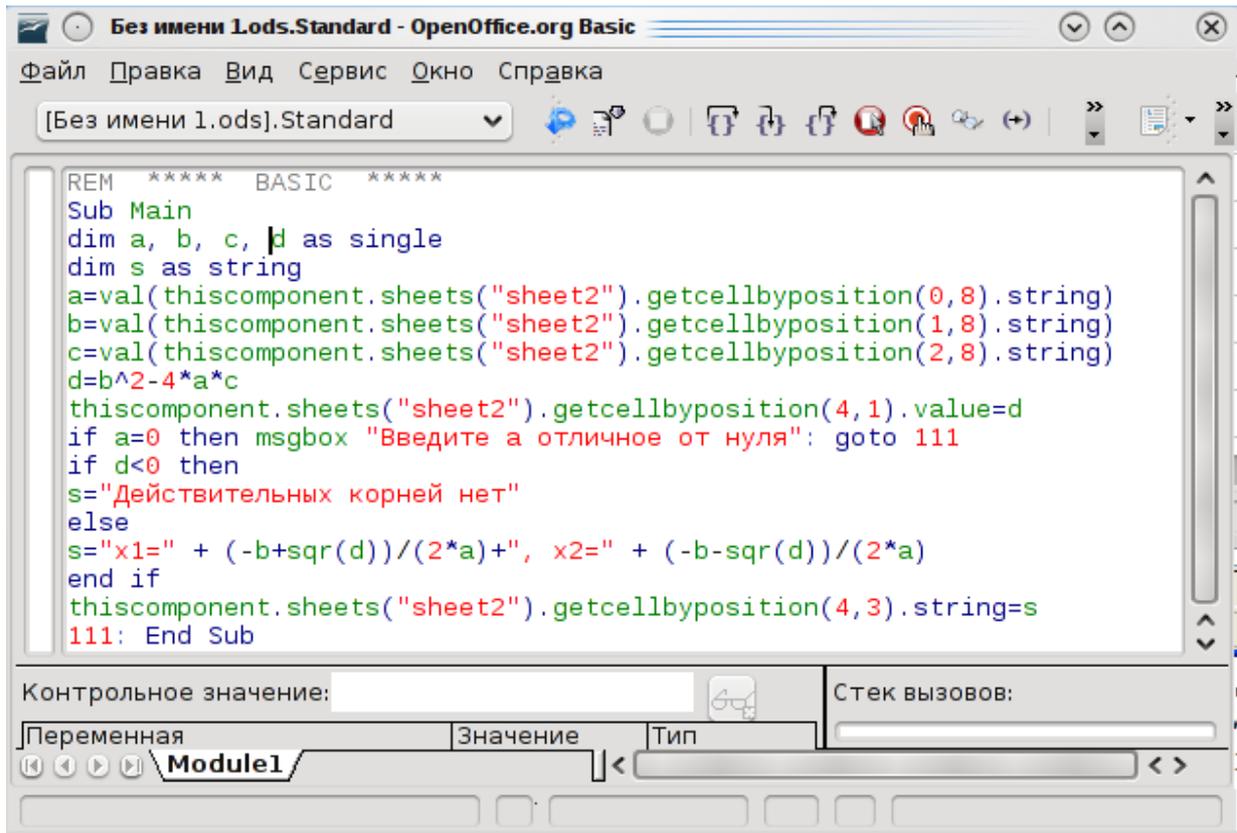
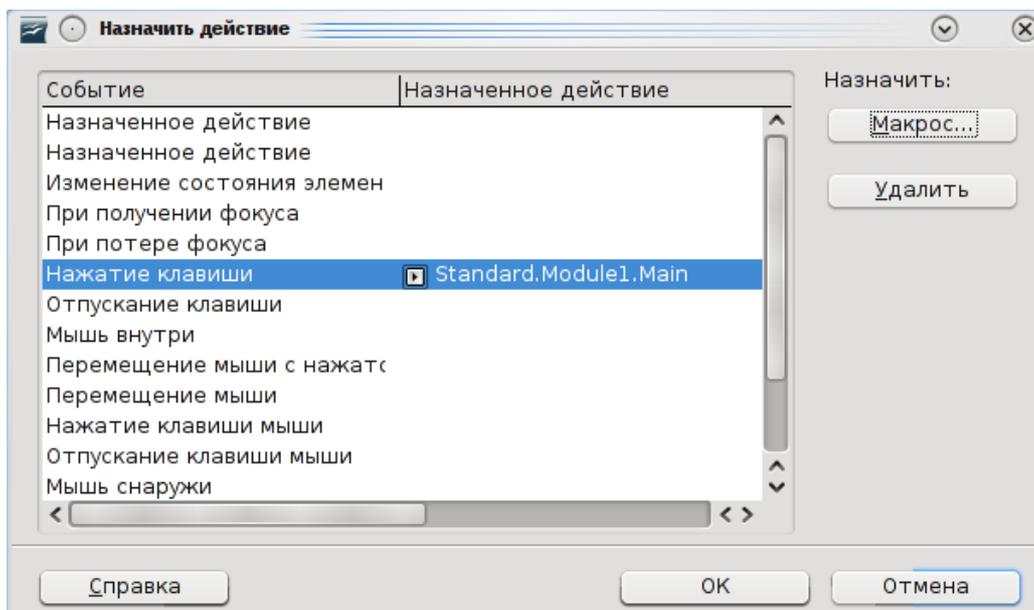


Рис.6. Код программы

Для преобразования данных из строкового типа в числовой воспользуемся встроенной функцией `val(строка)`.

5. После того как макрос написан, надо назначить этот макрос нашей командной кнопке. Для этого мы, находясь в режиме разработки нашего документа, выделяем объект *Кнопка*, открываем окно *Элемент управления* (быстрый способ сделать это — воспользоваться панелью инструментов *Элементы управления*), на вкладке *События* этого окна выбираем событие *Нажатие клавиши мыши* и вызываем диалоговое окно для этого события. В окне *Назначить действие* нажимаем кнопку *Макрос*.



Теперь мы получили готовое приложение. Для проверки его работоспособности на панели инструментов *Элемент управления* выходим из режима разработки, теперь вводим в текстовые поля коэффициенты уравнения, нажимаем на командную кнопку. В результате получаем:

	A	B	C	D	E	F
1	Решаем квадратное уравнение $ax^2+bx+c=0$					
2	Введите коэффициент a	23	Дискриминант равен	-76		
3						
4	Введите коэффициент b	4	Ответ:	Действительных корней нет		
5						
6	Введите коэффициент c	1	Решить			
7						

Таким образом, мы стараемся показать учащимся возможности электронной таблицы с точки зрения программирования и автоматизации вычислений, возможность её использования для решения практически любой достаточно сложной математической задачи, тем самым формируя высокий уровень информационно-технологической культуры учеников.

Используемые источники:

1. <http://www.metod-kopilka.ru/page-1-1-3.html>
2. <http://ru.wikipedia.org/wiki/Макрос>
3. OpenOffice.org: Теория и практика / И. Хахаев, В. Машков, Г. Губкина и др. М. : ALT Linux; БИНОМ. Лаборатория знаний, 2008.
4. Питоньяк Э. OpenOffice.org Объяснение Макросов. — Hentzenwerke Publishing, 2004. (Перевод: Д. Чернов (с) 2007-2008).
5. <http://www.citforum.ru/pp/excel72.shtml>

ДИНАМИКА РАЗВИТИЯ И ВНЕДРЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА РЕГИОНАЛЬНОМ УРОВНЕ

Е.В. Андропова

ГОУ ВПО ЕГУ им. И.А. Бунина

г. Елец, Россия

E-mail: andropovaelena@mail.ru

Аннотация

В статье рассматриваются вопросы использования свободного программного обеспечения в образовательных учреждениях как альтернатива проприетарному. Выявляются психологические проблемы, сдерживающие внедрение свободного программного обеспечения в образовательных учреждениях. Даются практические рекомендации по использованию свободно распространяемой системы компьютерной математики Matha в учебном процессе школы.

Ключевые слова

Программное обеспечение, свободное и проприетарное программное обеспечение, психологический барьер.

Применение современных информационных и коммуникационных технологий (ИКТ) в образовательном процессе стимулирует разработку программного обеспечения и приложений, реализующих методологические идеи, связанные с организацией единого информационного образовательного пространства школы. Приобретение школами новых технических средств обучения: интерактивных досок, интерактивных проекторов, мобильных комплектов (ноутбук, проектор, экран), мобильных классов, а также подключение к сети Интернет — расширяет области применения ИКТ в учебно-воспитательном процессе.

Многие педагоги отмечают, что применение ИКТ изменяет и обогащает содержание образования, создаёт возможность для повышения интенсивности урока, увеличивает педагогическое воздействие на формирование творческого потенциала учащихся, способствует индивидуализации и дифференциации обучения.

Анализируя целесообразность использования ИКТ в учебном процессе, можно сделать вывод о том, что компьютер — это источник учебной информации, наглядное пособие, тренажёр, средство диагностики и контроля уровня знаний, умений, навыков учащихся. «Владение информационными и коммуникационными технологиями учителями современной школы является основой повышения качества образования» [2].

Признавая в целом положительное влияние информационных технологий на образовательный процесс, однако стоит отметить, что на практике мы сталкиваемся с ситуацией, когда многие учителя-предметники не хотят использовать ИКТ в своей профессиональной деятельности. На самом деле это связано с тем, что, во-первых, преподаватель жёстко лимитирован временными рамками и тематическим планированием, что не позволяет в полном объёме и продуктивно использовать непосредственно на уроке педагогические программные средства учебного назначения, а также интерактивные материалы. Во-вторых, использование компьютерных технологий связано и с изменением подхода к уроку и его функциям со стороны самого учителя. Уроки с использованием компьютера – это новый, ещё практически не разработанный вид урока. Компьютер на таком уроке должен стать средством, а не целью [3]. В-третьих, применение компьютерных технологий на уроке требует от учителя необходимости самостоятельного поиска программных средств учебного назначения, адаптированных под конкретный учебный материал урока.

Многие научно-исследовательские центры, вузы уделяют особое внимание процессу информатизации как одному из значимых механизмов становления новой образовательной парадигмы на современном этапе. В их числе и Елецкий государственный университет им. И.А. Бунина. Лабораторией университета «Использование современных информационных технологий в образовательном процессе школы» (научный руководитель — профессор В.П. Кузовлев) систематически организуются курсы повышения квалификации преподавателей Липецкой области, в том числе и учителей-предметников. Кроме того, регулярно проводится оценка мотивационной и содержательной готовности учителей к использованию ИКТ в учебном процессе [1].

Сотрудниками лаборатории с 2006 года в ходе реализации проекта «Intel: обучение для будущего» ежегодно проводится обучение слушателей с целью внедрения передовых педагогических

технологий в учебный процесс, что позволило поднять уровень готовности и способности учителей региона к использованию средств информационных и коммуникационных технологий.

Такая работа с педагогами общеобразовательных школ помогла преодолеть многим учителям трудности в освоении и применении компьютера в профессиональной деятельности. И, как следствие, учителя нашего региона в настоящее время овладели новыми технологиями обучения, осуществили подбор программных средств учебного назначения, сформировали навыки работы с электронными образовательными ресурсами, начали нарабатывать опыт их использования на уроках.

Но теперь учителей ждёт очередная трудность в организации и осуществлении своей профессиональной деятельности с использованием ИКТ — переход с проприетарного на свободное программное обеспечение.

Отправной точкой стала «Концепция развития разработки и использования свободного программного обеспечения в Российской Федерации» [5]. План мероприятий Концепции рассчитан на период до 2010 года и определяет «государственную политику, принципы развития в области разработки и использования свободного программного обеспечения (СПО)». Использование СПО для образовательных целей направлено «на достижение значительной экономии бюджетных средств, расходуемых на разработку и модернизацию программ для ЭВМ как за счёт снижения затрат на приобретение лицензий, так и за счёт повторного использования разработанных программ и их компонентов».

На период с 2008 по 2010 гг. в рамках приоритетного национального проекта «Образование» (ПНПО) все школы были обеспечены лицензионной поддержкой стандартного (базового) коммерческого программного комплекса. Одновременно с мерами по легализации программного обеспечения в рамках ПНПО стартовал масштабный пилотный проект по внедрению в образовательных учреждениях Пакета свободного программного обеспечения, дистрибутив которого был разработан и апробирован в 2007-2008 гг. в рамках работ общеобразовательных учреждений трёх субъектов РФ (Республика Татарстан, Пермский край и Томская область). Одновременно с этим проводились работы по формированию требований к системе поддержки и мониторинга использования Пакета свободного программного обеспечения.

Таким образом, педагогам придётся переучиваться и опять

осваивать новые программные продукты. Ясно, что данная проблема является трудноразрешимой на сегодняшний день, поскольку состоит из целого ряда взаимосвязанных аспектов как субъективного, так и объективного характера.

Начиная с 2008 года, в Елецком государственном университете им. И.А. Бунина Центром СПО ведётся экспериментальная работа по выявлению условий и возможностей перевода образовательного процесса вуза и школ г. Ельца и Елецкого района на использование свободного программного обеспечения.

Обратимся к данным социологического опроса учителей города Ельца, проводимого с целью выявления уровня готовности учителей к использованию операционной системы Linux и свободно распространяемых программных продуктов. Учителям было предложено ответить на ряд опросов:

– Используете ли Вы в своей работе информационные технологии?

– С какой целью Вы используете информационные технологии?

– Знаете ли Вы, что такое свободное программное обеспечение?

– Различаете ли Вы свободное и проприетарное программное обеспечение?

– Знаете ли Вы о возможности перехода на свободное программное обеспечение учебных заведений?

– Знаете ли Вы об апробации ПСПО в рамках приоритетного нацпроекта «Образование» в республике Татарстан, Пермском крае и Томской области?

Опрос показал, что 55% учителей систематически используют средства ИКТ, 33% *иногда* используют средства ИКТ и только 12% заявили, что не уверены, будут ли использовать ИКТ в своей работе. Как выяснилось, 75% респондентов не различают свободное и проприетарное программное обеспечение.

После проведённого опроса сотрудниками Центра СПО было принято решение организовать и провести серию научно-практических семинаров на базе университета и МОУ-гимназии № 11 г. Ельца:

- «Проблемы и перспективы перехода на свободное программное обеспечение» (20 мая 2008 г.) при участии представителей Центра СПО Московского педагогического государственного университета, компании ARMADA, компании ALT Linux;

- «Актуальные проблемы использования программного обеспечения в учебном процессе общеобразовательных учреждений» для учителей и учащихся на базе МОУ-гимназии № 11 (29 октября 2008 г.);

- «Информационные технологии на базе свободного программного обеспечения» (27 апреля 2009 г.).

На семинарах обсуждались актуальные проблемы использования программного обеспечения в учебном процессе общеобразовательных учреждений, в информационной инфраструктуре вузов, предприятий и коммерческих структур; рассматривались вопросы развития программно-технического, нормативно-методического и организационного обеспечения свободных программных продуктов, участники обменивались практическим опытом внедрения свободного программного обеспечения в регионе.

После проделанной работы Центром СПО был проведён второй социологический опрос, цель которого — выяснить уровень готовности учителей к переходу на свободное программное обеспечение и их желания изучать новые программные продукты. Для этого педагогам предлагалось ответить на ряд уже других вопросов:

- Если будет закуплено проприетарное программное обеспечение, то будете ли Вы продолжать использовать средства ИКТ в своей работе?

- Если процесс обучения переведут на ПСПО, готовы ли Вы использовать в своей работе?

- Хотели ли Вы научиться применять ПСПО в своей профессиональной деятельности?

- Что Вам мешает использовать ПСПО в работе?

Опрос показал, что 90% учителей не готовы перейти на использование свободного программного обеспечения в учебно-воспитательном процессе школы. Основными причинами этому были названы: нехватка времени на изучение новых программных продуктов, нежелание изучать новое программное обеспечение, а также отсутствие практических рекомендаций и методической поддержки. Наиболее интересные для целей нашего исследования причины следующие: боязнь того, что приобретённые навыки окажутся ненужными и будет ущемлена профессиональная гордость; нежелание расходовать силы на переобучение; боязнь роста интенсивности труда; страх перед предопределённостью,

обусловленной непониманием сути и последствий перехода на свободное программное обеспечение.

Таким образом, субъективным аспектом рассматриваемой проблемы прежде всего выступает психологический «барьер». Основная масса учителей не готова к переходу на использование СПО в своей работе в силу определённого «страха» перед «незнакомым» программным обеспечением и необходимостью его использования на уроках. Общение с учителями показывает, что их зачастую «пугают» сами названия компьютерных программ, не говоря уже о перспективах освоения работы с ними. В силу этого многие педагоги не представляют себе реальных возможностей бесплатных программных продуктов (бесплатное — значит плохое) и, как следствие, сомневаются в их эффективности.

Пока психологический и когнитивный барьеры не будут преодолены, свободное программное обеспечение вряд ли сможет дать реальную отдачу в образовательном процессе.

Вместе с тем у проблемы перехода на ПСПО имеются и серьёзные объективные аспекты. Во-первых, отсутствует **действенная** система обучения учителей. В октябре 2009 года группа компаний «АйТи» была объявлена победителем конкурса Федерального агентства по образованию на оказание полного комплекса услуг в рамках проекта по внедрению во всех российских общеобразовательных учреждениях ПСПО. Компания «АйТи» организовала обучение учителей по программам «Установка и администрирование ПСПО» и «Применение ПСПО», они выполняли комплекс лабораторных работ в дистанционном режиме в очень сжатые сроки, аудиторные занятия со слушателями курсов не проводились. Во-вторых, отсутствуют действенные стимулы к их освоению. В-третьих, не следует сбрасывать со счетов и материальный аспект: школы деньги на очное обучение не выделяют, учитель со своей стороны позволить себе платное обучение не всегда может.

Поэтому проблема перехода на ПСПО на сегодняшний день остаётся актуальной.

Мы видим несколько путей её решения:

1) создание на базе ЕГУ им. И.А. Бунина Центра СПО с целью поддержки преподавателей, учителей школ города Ельца в эффективном использовании преимуществ свободного программного обеспечения в образовательном процессе и информационной инфраструктуре учреждений;

2) руководство деятельностью экспериментальных площадок на базе школ г. Ельца и Елецкого района с целью отработки процедуры внедрения ПСПО в средних учебных заведениях, а также проведения оценки качества, степени функциональности, целесообразности, успешности применения программного состава дистрибутива ПСПО к использованию в учебном процессе;

3) проведение курсов повышения квалификации и переподготовки педагогов по программе формирования их готовности к работе со свободным программным обеспечением, необходимым для использования в профессиональной деятельности (такие курсы уже проводятся на базе Центра СПО);

3) организация непосредственно в образовательных учреждениях методических обучающих семинаров для педагогов по обмену опытом работы с ПСПО и его применения в рамках классно-урочной системы;

4) регулярное проведение практических семинаров по обмену опытом в данной сфере на уровне города;

5) разработка практических и методических рекомендаций по использованию ПСПО в деятельности учителя-предметника.

К основным задачам по преодолению психологического барьера у учителей мы относим: содействие личностному и профессиональному росту учителей в области использования ПСПО, активизацию способности к самообучению и саморазвитию; выявление основных проблем при работе со свободным программным обеспечением и определение причин их возникновения, путей и средств их разрешения; выяснение затруднений в профессиональной деятельности у отдельных учителей-предметников и оказание им методической помощи. Таким образом, сотрудники Центра СПО ЕГУ им. И.А. Бунина оказывают консультативную и методическую помощь учителям, занимаются разработкой и выпуском учебно-методических комплексов для поддержки учебного процесса, тем самым обеспечивая процесс профессионального сопровождения, который мы видим в организации сотрудничества с учителем, направленного на поиск путей решения актуальных проблем, профессиональную поддержку в преодолении синдрома истощения, создание условий для успешного применения свободного программного обеспечения на уроках.

Деятельность Центра СПО направлена на то, чтобы не позволить произойти падению доли использования средств инфор-

мационных и коммуникационных технологий из-за перехода на Пакет свободного программного обеспечения.

Мы считаем, что проведение уроков с использованием информационных технологий – это мощный стимул в обучении. Посредством таких уроков активизируются психические процессы учащихся: восприятие, внимание, память, мышление; гораздо активнее и быстрее происходит возбуждение познавательного интереса.

В подтверждение вышесказанного приведём методические рекомендации по использованию системы компьютерной математики Maxima (распространяется по лицензии GNU GPL и может эффективно использоваться в учебном процессе) учителем или учениками 10-11 классов для самоконтроля и самокоррекции своей учебной деятельности при решении задач курса алгебры и начал анализа.

Пример 1. Требуется упростить выражение $\left(\frac{1}{27}\right)^{-\frac{1}{3}} - 8^{-\frac{5}{3}} + (12^0)^2 \cdot 7 + 32 \cdot 2^{-4} \cdot 16^{-\frac{3}{2}}$.

Решение.

Очень часто при выполнении операций по упрощению выражений ученики допускают ошибки. В том случае, если у ученика или учителя нет возможности сравнить полученное решение с правильным (например, не даны правильные варианты ответа или не приводится решение задачи), тогда может прийти на помощь система компьютерной математики Maxima.

В этой системе к числовым выражениям автоматически применяется функция автоупрощения числовых выражений. Однако если выражение громоздкое, как в нашем случае, то желательно сначала проконтролировать правильность набранного выражения. С этой целью мы сначала отключим эту функцию и наберём выражение (пробел заменяет знак умножения):

```
(%i1) simp:false$
(%i2) (1/27)^(-1/3)-8^(-5/3)+(12^0)^2*7+32*2^(-4)*16^(-3/2);
(%o2)  $\left(\frac{1}{27}\right)^{-\frac{1}{3}} - 8^{-\frac{5}{3}} + (12^0)^2 \cdot 7 + 32 \cdot 2^{-4} \cdot 16^{-\frac{3}{2}}$ 
```

Теперь, убедившись в том, что выражение набрано правильно, включаем функцию автоупрощения и вычисляем значение выражения.

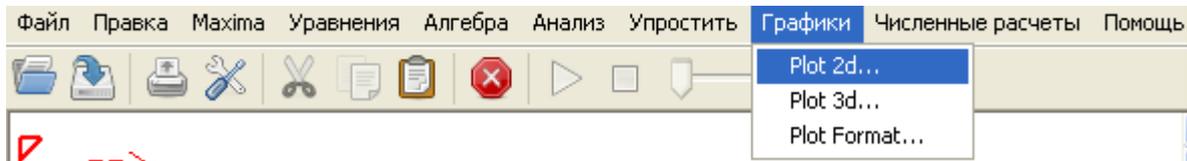
```
(%i3) simp:true$
(%i4) (1/27)^(-1/3)-8^(-5/3)+(12^0)^2*7+32*2^(-4)*16^(-3/2);
(%o4) 10
```

Конечно, *Math*, не выводит ход решения поставленной задачи, однако возможность сверить полученный результат тоже бывает важна.

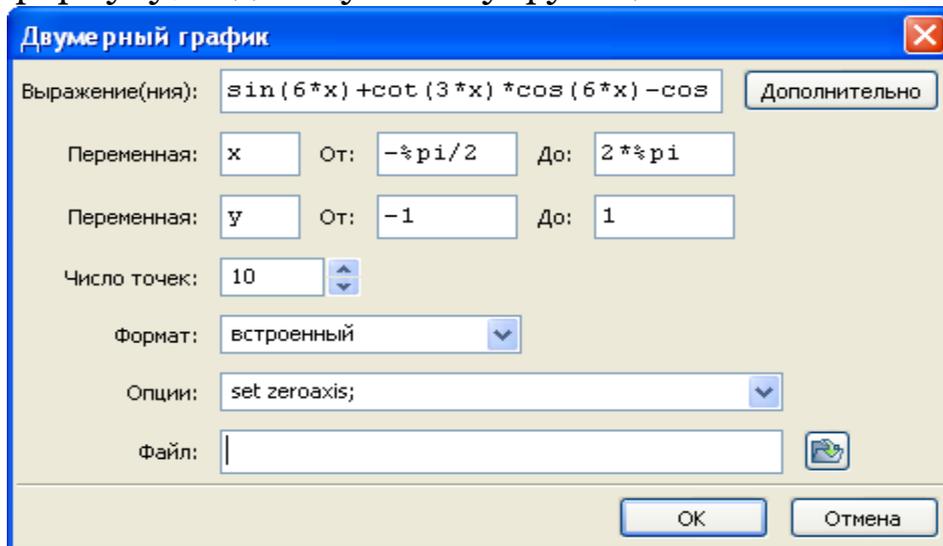
Пример 2. Требуется определить число корней уравнения $\sin 6x + \operatorname{ctg} 3x \cos 6x = \cos 3x$ на промежутке $\left(-\frac{\pi}{2}; 2\pi\right)$.

Решение.

Построим график функции $y = \sin 6x + \operatorname{ctg} 3x \cos 6x - \cos 3x$ на заданном промежутке, диапазон изменения переменной $y - [-1; 1]$. Для этого воспользуемся пунктом меню *Графики* → *Plot2D* (функция построения графика на плоскости).

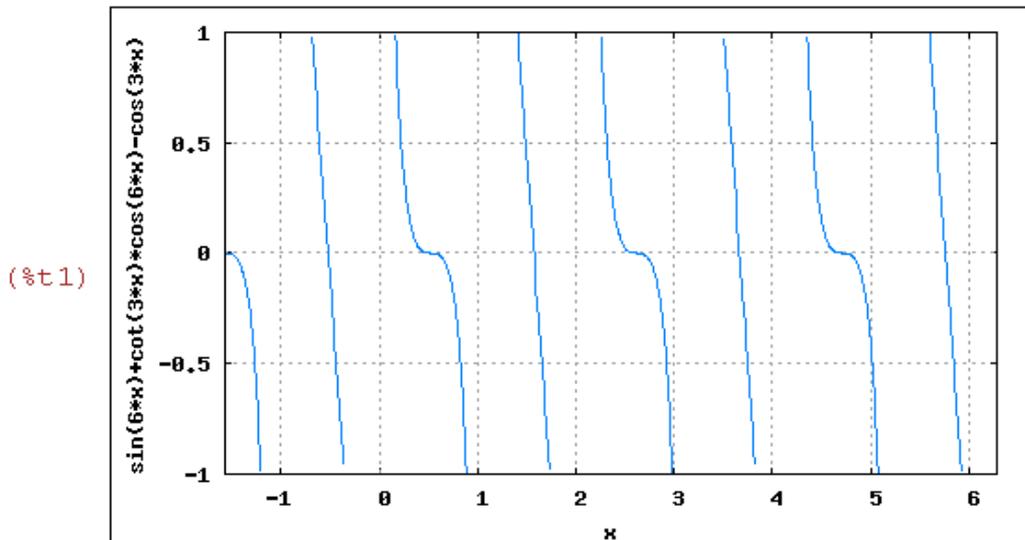


Откроется диалоговое окно, в котором в текстовое поле мы вводим формулу, задающую нашу функцию:



После нажатия на кнопке *ОК* получаем:

```
(%i1) wxplot2d([sin(6*x)+cot(3*x)*cos(6*x)-cos(3*x)],
[x,-%pi/2,2*pi],[y,-1,1],
[gnuplot_preamble, "set grid;"])$
plot2d: some values were clipped.
```



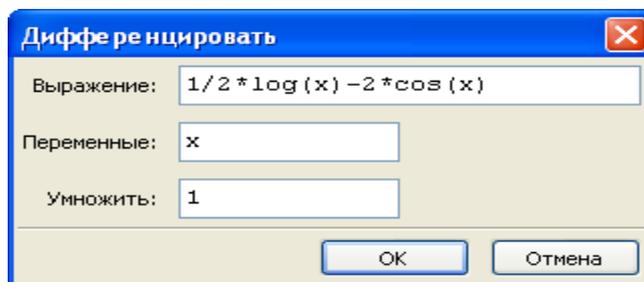
Теперь мы определяем количество точек пересечения графика с осью Ox : таких точек 7, не считая точку $x = 0$, поскольку левый конец указанного в условии задачи отрезка не входит.

На самом деле с математической точки зрения приведённое выше задание решается куда более сложно (надо знать и помнить сразу несколько тригонометрических формул и уметь их применять к упрощению выражения). Система *Maxima* позволяет получить правильный ответ мгновенно.

Пример 3. Требуется вычислить значение производной функции $y = \frac{1}{2 \cdot \ln x} - 2 \cos x$ в точке $x_0 = 1$.

Решение:

Выбираем пункт меню *Анализ* → *Дифференцировать*. В диалоговое окно вводим функцию:



После нажатия на кнопку *ОК* получаем значение производной:

```
(%i1) diff(1/2*log(x)-2*cos(x), x, 1);
```

```
(%o1) 2 sin(x)+ $\frac{1}{2x}$ 
```

Подставим в найденную производную значение $x_0=1$ и получаем требуемый ответ:

```
(%i2) %, x=1;
```

```
(%o2) 2 sin(1)+ $\frac{1}{2}$ 
```

Таким образом, прежде чем свободно распространяемые программные продукты будут использоваться в школе, учителя должны быть в достаточной степени подготовлены для того, чтобы обеспечить высокий уровень преподавания и организации учебно-воспитательного процесса, которые служат условием приобретения обучающимися качественных знаний.

Используемые источники:

1. Кузовлев В.П., Фаустова Н.П. Проблемная лаборатория как путь приобщения учителя к использованию информационных технологий в образовательном процессе // Педагогическая информатика. – № 1. – 2009. – С. 3-12.
2. Мирошниченко Е.Б. Применение информационных технологий в современной школе // Цифровые образовательные ресурсы в учебном процессе педагогического вуза и школы: тезисы докладов III Региональной научно-практической конференции. В 2 ч. Ч. II – Воронеж: ВГПУ, 2009. – С. 15.
3. Обухов Б.Г. Основные проблемы использования ИКТ в условиях классно-урочной системы // Цифровые образовательные ресурсы в учебном процессе педагогического вуза и школы: тезисы докладов III Региональной научно-практической конференции. В 2 ч. Ч. II – Воронеж: ВГПУ, 2009. – С. 36-37.
4. <http://fosscenter.elsu.ru>
5. <http://minkomsvjaz.ru/ministry/documents/1117/3264.shtml> [Электронный ресурс] / «Концепция развития разработки и использования свободного программного обеспечения в Российской Федерации».

ОПЫТ ИСПОЛЬЗОВАНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В СФЕРЕ ВЫСШЕГО ОБРАЗОВАНИЯ

УДК 004.416.6

ДИСТРИБУТИВ LINUX ДЛЯ ЕГУ ИМ. И.А. БУНИНА

А.А. Панюкова

ООО «ALT Linux»

г. Москва, Россия

E-mail: mex3@altlinux.ru

Аннотация

В статье рассматривается один из подходов разработки дистрибутива Linux, соответствующего задачам учебного заведения.

Ключевые слова

Высшее профессиональное образование, дистрибутив, Linux.

Желание получить профильный дистрибутив достаточно распространено в кругах, связанных с образованием, — будь то школы, университеты или даже обычные детские кружки. Причины тому могут быть самыми разными — от отсутствия согласия между преподавательским составом до странного статуса используемого для обучения оборудования — «сегодня оно есть, а завтра работать уже с другим». Даже когда подобных проблем нет, неоднократно встаёт вопрос о наличии профильного дистрибутива, удобного для разворачивания системы «с нуля» с последующими минимальными действиями после установки или с полным их отсутствием.

Была поставлена задача — создать дистрибутив Linux, отвечающий задачам Елецкого государственного университета имени И.А. Бунина.

Исходные условия

Организационные ограничения, свойства оборудования:

- старые компьютеры — ~256 Мб памяти со всеми вытекающими последствиями;
- агрессивная учебная среда — разворачивать систему надо быстро;
- отсутствие канала, подходящего для скачивания обновлений.

Каким должен быть дистрибутив?

1. Полным для данной задачи: все необходимые пакеты ставятся при установке.
2. Компактным: не на всех компьютерах есть CD/DVD привод, на некоторых только CD.
3. Гибким: не все пакеты, которые можно поставить, нужно ставить.

Проектирование: определение задач, приоритетов и ПО

К сожалению, не все пакеты, которые хотелось бы, можно уместить на один диск — причиной этому, в первую очередь, является место. Формат CD — достаточно ограниченный и, даже несмотря на небольшой вес большей части прикладного ПО, сделать универсальный диск, подходящий для всего, — практически невозможно.

Перед подготовкой диска все пакеты-пожелания были разбиты на группы, в зависимости от необходимости решения каких-то задач, наличия оборудования и т.п.

Обязательное

Абсолютный приоритет был выделен следующему ПО/свойствам:

- Lazarus
- MySQL
- FreePascal
- Локализация, близкая к полной

Желательное

ПО, наличие которого крайне желательно, но которое не являются необходимым для непосредственно процесса работы обучающихся:

- Офисное ПО – OpenOffice.org
- Браузер – Firefox
- Работа с графикой – GIMP, как следствие, удобное ПО для просмотра изображений, например, Gphoto2
- Возможность читать книги, документацию – evince, djview

Дополнительное

Помимо основных задач, при обсуждении состава дистрибутива были озвучены дополнительные задачи, такие как простейшая нарезка видео и/или аудио в графическом видео/аудиоредакторе с интерфейсом, позволяющим сделать это достаточно точно без особых неудобств. Как следствие — проигрывание видео и аудио. Кроме того, было озвучено пожелание в наличии ПО для записи CD/DVD дисков для компьютеров, где есть приводы.

Это ПО имеет наименьший приоритет и может быть не добавлено, согласно концепции, в дистрибутив из-за нехватки места.

Таким образом, для обозначенных выше задач было подобрано следующее ПО:

- Редактор видео — Kino
- Редактор аудио — Audacity
- Проигрыватель аудио и видео — Gnome-mpplayer
- ПО для записи CD/DVD — Brasero

Ненужное

Достаточно много места на диске обычно занимает ПО, необходимое для корректной работы системы при разной конфигурации железа, а также ПО для работы с различными внешними устройствами, которое добавляется «про запас». В случае универсального дистрибутива это необходимо, но в данном конкретном случае исходные условия примерно известны и фиксированы, что позволяет сократить занимаемое место не помещением на диск лишних драйверов, а также прочего прикладного ПО для работы, например, с принтерами и сканерами, так как в классе они не предполагаются.

Прототип

По мотивам озвученных пожеланий был подобран прототип, который лёг в основу решения для университета — Simply Linux. Достоинства и недостатки этого прототипа перечислены ниже.

1. Практически полное пересечение по желаемым приложениям — нет необходимости подбирать качественное ПО под все озвученные задачи.

2. Исходный образ занимает CD — меньше проблем с «подгоном» образа под желаемый размер, минимум «лишнего» ПО.

3. XFCE — лёгкая и при этом доступная для неподготовленного пользователя DE — на сегодняшний день является практи-

чески единственным поддерживаемым решением для подобного парка компьютеров.

4. Практически полная локализация. Локализация — одна из ключевых особенностей Simply Linux. Перед выпуском дистрибутива всё вошедшее ПО было дополнительно проверено на наличие и корректность локализации, некоторые приложения были допереведены специально для Simply Linux.

5. <http://slinux.ru> — развивающийся ресурс, на котором появляются статьи для начинающих пользователей. Может быть использован как дополнительная документация по проекту.

6. Недостатки — нет LiveCD.

Реализация: жертвы

Пожелания, даже с небольшим приоритетом, были учтены, но, к сожалению, не все. В итоге из наименее принципиального в дистрибутив не попала программа для записи CD и DVD дисков. Из того, что могло бы быть принципиальным, но оказалось не подходящим для текущей ситуации: дополнительная документация и режим LiveCD. Первое, учитывая характер и приоритеты выбранного ПО, имело слишком вырожденный характер, чтобы технические затраты на добавление были обоснованными. Второе — слишком тяжеловесным, чтобы поместиться на один диск с установщиком системы.

УДК 004.416.6

ОПЫТ ПРИМЕНЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ФИЛИАЛЕ ГУ КУЗГТУ В Г. ПРОКОПЬЕВСКЕ

М.Н. Смирнова

Филиал ГУ КузГТУ в г. Прокопьевске

г. Прокопьевск, Россия

E-mail: smirnovaMN@list.ru

Аннотация

В статье описаны проекты, выполненные студентами филиала ГУ КузГТУ в г. Прокопьевске с использованием CMS Joomla!. Делается вывод об актуальности использования свободного программного обеспечения для целей образования.

Ключевые слова

Свободное программное обеспечение, CMS Joomla!, Web-ресурс, сайт, студенческий проект.

Свободное программное обеспечение (СПО) широко применяется в высших учебных заведениях всего мира как в учебном процессе, так и в научных проектах. Основным преимуществом СПО для целей образования является имеющая правовую основу возможность свободно изучать документированный исходный код свободных программ для ЭВМ и модифицировать его, в том числе создавать на его базе собственные разработки.

В филиале Кузбасского государственного технического университета в г. Прокопьевске наряду с другими свободными программными продуктами используется CMS Joomla!. Joomla! — система управления контентом сайта, написанная на языках PHP и JavaScript, использующая в качестве хранилища базу данных MySQL. Данная система является свободным программным обеспечением, распространяемым под лицензией GNU GPL.

Приёмы работы с CMS Joomla! студенты специальности 080801 «Прикладная информатика в экономике» изучают на факультативном курсе «Web-программирование и Web-дизайн». При изучении этого курса студенты знакомятся с техническими возможностями системы Joomla!, ее модулями и компонентами, создают собственные шаблоны и компоненты, а также закрепляют знание языка программирования PHP и навыки работы с базами данных, полученные при изучении других дисциплин. Очень часто результатом изучения факультативного курса является разработанный «с нуля» студенческий проект — Web-сайт, поддержку которого авторы осуществляют еще долгое время.

Примерами таких студенческих проектов являются Web-сайты «Автоматизированная информационная система поддержки трудоустройства студентов и выпускников филиала ГУ КузГТУ в г. Прокопьевске», «Имитационная модель "Закупка товаров и услуг государственными учреждениями"» и «АИС Электронный учебно-методический комплекс».

Рассмотрим подробнее проект «Автоматизированная информационная система поддержки трудоустройства студентов и выпускников филиала ГУ КузГТУ в г. Прокопьевске».

Разработанный Web-ресурс является площадкой, на которой студенты и выпускники могут рассказать о своих достижениях в учебной и внеучебной деятельности, а потенциальные работодатели найти подходящую кандидатуру на свободное рабочее место.

В рамках проекта студенту предоставляются следующие возможности: ведение личного портфолио достижений в учебной, научной и других видах деятельности, которое отражает профессиональный рост студента; размещение списка работ, которые студент может выполнять во время каникул или без отрыва от учёбы; помощь в поиске места прохождения производственной практики; мгновенное получение уведомления через SMS и электронной почтой о размещении вакансий.

Личная страница студента (его Web-портфолио) содержит следующие разделы: личная информация (фамилия, имя, отчество, специальность, курс, контактная информация, личная информация и т.д.); официальные документы (грамоты, дипломы, сертификаты, результаты тестирования и т.д.); творческие работы (темы выполненных курсовых работ и дипломной работы, участие в научной работе, участие в проектах и т.д.); опыт работы (места прохождения практики, работы во внеучебное время и на каникулах и т.д.); отзывы и рекомендации (рецензии на опубликованные статьи, благодарственные письма, рекомендации от преподавателей, отзывы руководителей практики от предприятия и от вуза и т.д.). Из этих данных студент или выпускник имеет возможность автоматически сформировать резюме для отправки потенциальным работодателям.

Работодателю предоставляются такие возможности, как: размещение информации об организации и об имеющихся вакансиях; оценка профессиональной пригодности кандидатов через просмотр портфолио достижений студента; приглашение на работу наиболее подходящего, по его мнению, кандидата; проведение онлайн – собеседования; поиск работников на сезонные работы (в период каникул), а также на неполный рабочий день.

Форма исполнения проекта выбрана из-за популярности социальных сетей среди молодёжи.

Результатами реализации проекта являются: помощь в трудоустройстве студентов на работу во время каникул, а также в свободное от учёбы время; помощь студентам в составлении резюме; помощь выпускникам в трудоустройстве; помощь работодателям в поиске подходящей кандидатуры; сбор статистиче-

ских данных о трудоустройстве выпускников; увеличение количества молодых специалистов, нашедших работу по специальности.

Проект «Имитационная модель "Закупка товаров и услуг государственными учреждениями"» также является примером успешного внедрения студенческой разработки в учебный процесс филиала.

Целью данного проекта являлась разработка модели, имитирующей механизм государственных закупок (запрос котировок, конкурс и аукцион) для проведения деловой игры на занятиях по дисциплине «Информационные технологии управления» студентов специальности «Государственное и муниципальное управление».

Участники деловой игры разбиваются на команды, представляющие государственные предприятия, размещающие заказы, и предприятия, оказывающие услуги или производящие товары. С помощью функционала сайта имитация государственных закупок производится в виде запроса котировок, конкурса и аукциона. Студенты изучают сам процесс государственных закупок, составляют пакеты необходимых документов, изучают необходимые законодательные и нормативные акты.

Web-сайт «АИС Электронный учебно-методический комплекс» разработан с целью упорядочения учебно-методических материалов по дисциплинам, преподаваемым в филиале. На каждую дисциплину отведена отдельная страница, имеющая разделы: извлечение содержания ГОС по учебной дисциплине; рабочая программа учебной дисциплины; методические указания по практическим занятиям; методические указания по выполнению лабораторных работ; перечень тем рефератов; перечень и примеры содержания контрольных работ; методические указания по выполнению курсовых работ; перечень вопросов на зачет; перечень экзаменационных вопросов; перечень прикладных компьютерных программ, используемых по дисциплине; тестовые материалы; содержание самостоятельной работы и форма контроля, обеспеченность дисциплины учебной литературой.

Для каждой единицы учебно-методических материалов указан год издания, что облегчает поиск и замену устаревших. Также указаны преподаватели, ведущие занятия по указанной дисциплине, кафедра, за которой дисциплина закреплена, что позволяет формировать необходимые заведующему кафедрой отчёты.

Результатами реализации проекта являются: комплексный подход в обеспечении студентов учебно-методическими материалами; удобный поиск материалов, возможность их распечатывания при необходимости; снижение затрат филиала на учебно-методические материалы; унификация учебных материалов филиала и т.д.

При разработке проектов студентам приходится изучать большое количество источников информации как по теме своего проекта, так и по использованию CMS Joomla!, базы данных MySQL, языки программирования PHP и HTTP-сервера Apache. Так как перечисленное ПО является свободным, то всю информацию по его использованию можно найти в свободном доступе в сети Интернет. Кроме того, в сети Интернет можно найти большое количество модулей и шаблонов, созданных другими разработчиками, и применить их в своём проекте.

Таким образом, применение свободного программного обеспечения в образовательном процессе вуза имеет большое значение: формируется мотивация студентов к самообразованию; разрабатывая собственные проекты, студенты учатся применять на практике полученные ранее знания; лучшие проекты могут быть внедрены в учебный процесс или использоваться в административной деятельности учебного заведения, что позволяет вузу экономить средства, устанавливая ПО на необходимое количество компьютеров.

УДК 004.4:061.68

АЛЬТЕРНАТИВНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ВУЗАХ МИНИСТЕРСТВА ОБОРОНЫ

Н.А. Ивличева, П.С. Ивличев

Рязанское высшее воздушно-десантное командное училище
(военный институт)

Рязанский филиал ГОУ «Московский университет МВД Рос-
сии»

г. Рязань, Россия

E-mail: psi940@mail.ru

Аннотация

В статье рассмотрены некоторые методические вопросы, связанные с использованием пакета OpenOffice.org при изучении дисциплины «Информатика» с курсантами инженерных специальностей. Приве-

дены рекомендации по выбору программного обеспечения при переходе на операционную систему МСВС.

Ключевые слова

Свободное программное обеспечение, альтернативные программные продукты, OpenOffice.org, МСВС, Министерство обороны, преподавание, информатика, методика.

Одним из аспектов реформирования Вооружённых Сил Российской Федерации в целом и военного образования в частности является переход ВУЗов Министерства обороны на операционную систему МСВС, основанную на ядре Linux. К тому же с 2006 года в нашей стране резко усилилась борьба с нарушениями авторских прав. Поскольку фактически единственным пакетом офисных прикладных программ, используемым в России, является пакет Microsoft Office, а операционной системой — Windows, встаёт вопрос о выборе программного обеспечения, служащего для поддержки функционирования самого образовательного учреждения и для обучения курсантов работе в сфере информационных технологий. Особо острой в этой связи является проблема преподавания дисциплины «Информатика».

Одним из обязательных навыков, формируемых при обучении информатике, является умение вести электронную документацию. Например, в курсе информатики на специальности «Автомобили и автомобильное хозяйство» офисному программному обеспечению посвящены пять тем. Наиболее известным бесплатным кроссплатформенным офисным продуктом, способным решить поставленную задачу, является проект OpenOffice.org компании Sun Microsystems. Отметим некоторые методические аспекты, связанные с обучением слушателей работе с этим пакетом.

Наиболее часто используемая программа в офисном делопроизводстве – текстовый процессор. С точки зрения методики обучения пользователей работе с текстовым процессором Writer при отказе от традиционного MS Word можно сделать следующие выводы:

1. При рассмотрении приемов форматирования простейших документов методика обучения не меняется.

2. При обучении форматированию таблиц необходимо разработать методику настройки видов границ ячеек для различных типов таблиц.

3. При обучении созданию векторных рисунков необходимо дополнительно учесть работу с панелью свойств изображения, оформленной в стиле Corel Draw.

4. При создании комплексных документов придется формировать новые методики для обучения работе с колонтитулами, стилями, оглавлением и полями.

5. Без дополнительных затрат становится возможным на начальном этапе ознакомить обучаемых с работой с документами в формате PDF.

Табличный процессор стоит на втором месте по частоте использования среди офисных программ. Сравнение Calc с Excel с точки зрения пользователя продукта компании Microsoft позволяет сделать следующие выводы:

1. Организация простейших таблиц с расчётной частью в виде формул осуществляется так же, как и в Excel.

2. Наборы функций в программах отличаются в основном только названиями.

3. Построение диаграмм в Calc гораздо менее интуитивно. Форматирование диаграмм требует хороших навыков работы с Calc и мышью.

Сравнение продуктов с точки зрения надстроек и макросов позволяет сразу разделить их. В стандартной поставке Calc какие-либо надстройки отсутствуют. Таким образом, для использования Calc в линейном программировании, математической статистике или в алгебре необходима самостоятельная разметка рабочих листов.

При преподавании Calc вместо Excel следует учитывать следующие особенности методики:

1. Знакомство со средой и создание простейших электронных таблиц могут осуществляться по прежней методике.

2. При рассмотрении программных функций следует подготовить справочник с адаптированным описанием наиболее широко используемых функций.

3. При построении диаграмм следует предварительно проверять структуру опорных таблиц, типы диаграмм следует подбирать так, чтобы свести к минимуму форматирование, зависящее от точности позиционирования мыши.

4. При изучении математического моделирования Calc может быть использован как «большой калькулятор» при освоении методики расчета моделей. Поскольку новых возможностей программы при этом не открывается, такие занятия целесообразнее проводить вне рамок дисциплины «Информатика».

Система управления базами данных Base, как и другие приложения пакета OpenOffice.org, помимо поддержки своего формата Open Document Base позволяет подключаться и к другим базам данных, в том числе и MS Access. Импорт структур данных при этом неполный: формы и отчеты не импортируются, а запросы импортируются в виде представлений данных. При преподавании Base вместо Access следует учитывать следующее:

1. Необходимо подготовить справочник типов данных и рассмотреть целесообразность применения каждого из них.

2. При работе с запросами следует уделить внимание работе с Мастером запросов в Base.

3. Для обучения работе с Мастерами отчетов и форм необходимо подготовить новые описания шагов построения отчетов и форм.

Однако для обучения студентов работе с системами управления базами данных, по нашему мнению, предпочтительнее использование СУБД MySQL из-за отсутствия каких-либо проблем с её установкой и эксплуатацией. Кроме того, использование MySQL создаёт дополнительную мотивацию для изучения языка программирования PHP, а программирование — традиционно одна из самых «нелюбимых» тем для учащихся в военных ВУЗах.

Рассмотрение программы подготовки презентаций Impress представляет интерес в первую очередь для преподавателя, поскольку с внедрением информационных технологий в учебный процесс все больше занятий проводится с использованием презентаций.

Интерфейс программы Impress практически полностью совпадает с интерфейсом Power Point 2003. В Impress также присутствует область задач, есть несколько режимов работы с презентацией. На наш взгляд, создание стандартных презентаций в Impress у пользователя, имеющего даже минимальные навыки работы в Power Point, затруднений не вызовет. Однако несколько особенностей бесплатной программы хотелось бы отметить:

- отсутствие понятия управляющей кнопки. В качестве аналога кнопки в Impress может выступать любой объект. Однако, изображение на объекте придётся настраивать вручную;

- малое количество стандартных шаблонов оформления и сложность их создания. Это наиболее существенная потеря при переходе на Impress;

- отсутствие возможности создания макросов.

Векторный редактор OpenOffice.org Draw и редактор формул OpenOffice.org Math в силу своей специфики вряд ли могут быть широко используемыми в учебном процессе.

Методическое построение материала для обучения работе в OpenOffice.org должно реализовывать как минимум две развивающие цели:

- развитие кругозора в области прикладного программного обеспечения;

- развитие способностей к самообразованию и анализу материала.

Этих целей можно достичь, используя постоянное сравнение программ пакета OpenOffice.org с MS Office. Поскольку развитие среднего образования в России приводит к постоянному увеличению числа первокурсников, обладающих навыками работы с компьютером, то часть обучаемых уже имеет какие-то знания по работе с офисными программами. Для таких обучаемых целесообразно сокращать количество инструкций, предлагать проведение аналогий.

Наконец, немаловажными являются и воспитательные цели:

- формирование гражданской ответственности. В качестве одной из причин выбора пакета программ OpenOffice.org слушателям должна поясняться правовая сторона не лицензионного использования коммерческих пакетов (в том числе и MS Office);

- формирование рационального подхода при расходовании средств.

Изучение работы с растровой графикой в условиях постреформенного образования уместно проводить с использованием кроссплатформенного проекта GIMP, который позволяет изучить все базовые операции работы с растровой графикой.

Формирование навыков программирования является важной задачей при обучении курсантов инженерной специальности.

В военных вузах курс информатики, включающий в себя программирование, достаточно ограничен, соответственно объем практических часов, отводимых на программирование, также невелик по сравнению со специализированными вузами.

Это делает невозможным качественное изучение нескольких языков программирования, то есть обучаемые лишены возможности сравнить языки и определить области применения каждого из них. Поэтому выбор языка за обучаемых должен осуществить преподаватель, и этот выбор должен учитывать следующие аспекты:

- актуальность языка;
- перспективность языка;
- доступность языковых конструкций;
- применимость к решению широкого класса задач;
- возможность его функционирования с операционной системой MSVC (Linux).

Программы пакета OpenOffice.org используют язык Native Basic для создания макросов. Этот язык по возможностям аналогичен VBA, к тому же он позволяет осветить основные тенденции современного программирования.

Однако выбор был сделан в пользу языка PHP. Он обладает некоторыми неоспоримыми достоинствами:

1. Язык актуален. Он является основой современного Web-программирования. По оценкам специалистов, 40-50% сайтов Интернета написано с использованием этого языка. Благодаря пакету Денвер, язык легко запускается и на локальных машинах.

2. На настоящий момент единственной претензией к языку является уязвимость кода перед хакерскими атаками. Вне глобальной сети это не имеет особого значения.

3. Языковые конструкции достаточно естественны, особенно легко конструкции воспринимаются курсантами, имеющими опыт программирования на Бейсике или Паскале в школе.

4. Несмотря на то, что .exe приложений на языке не создается, язык не сильно специализирован. В форме Web-страницы можно оформить решение любой базовой задачи программирования.

5. Язык является бесплатным. Курсантов следует ориентировать на то, что существуют хранилища готовых фрагментов кода, решающие различные вспомогательные задачи.

6. Код, написанный на PHP, работает как в Windows, так и в операционной системе Linux.

Следует подчеркнуть, что выбор языка осуществлён по комплексной оценке и предназначен для обучения курсантов

специальности, где программирование является вспомогательным навыком.

Для оснащения рабочего места программным обеспечением целесообразно использование пакета Денвер, который включает в себя сервер Apache и СУБД MySQL.

Выбор этого пакета позволяет увеличить число практических занятий по программированию за счёт охвата дополнительных тем:

1. Системы управления базами данных. В пакет входит графическая оболочка phpMyAdmin, которая позволяет естественным образом работать с СУБД. Кроме того, возможно обучение SQL-запросам средствами PHP.

2. Интернет-технологии. PHP-код может быть использован для обработки форм, создания динамических страниц.

К сожалению, нерешёнными проблемами при использовании операционных систем линейки Linux остаются темы «Система математических и инженерно-технических расчётов» и «Основы векторной графики», которые на инженерных специальностях предполагают использование AutoCAD.

Немаловажным моментом при формировании у слушателей правовой культуры использования программного обеспечения является личный пример преподавателя. Во избежание двойного стандарта в этом вопросе преподаватель как пользователь не должен являться нарушителем лицензионных соглашений. Для этого преподаватель должен обладать рядом знаний и навыков:

1. Преподаватель должен знать свои права как легального пользователя программного обеспечения (глава 70, часть IV Гражданского кодекса РФ) и ответственность за бездоговорное использование программного обеспечения, подлежащего правовой охране.

2. Преподаватель должен знать типовые положения лицензионных соглашений и дополнительных документов, регулирующих использование программного обеспечения (например, ограниченной гарантии).

3. Преподаватель должен знать и соблюдать условия лицензионных соглашений на преподаваемые и используемые программные продукты.

4. Преподаватель информатики должен обладать достаточно широким кругозором в области рынка прикладного программного обеспечения.

5. Изучение любого программного продукта должно включать изучение условий распространения, особенностей лицензионного соглашения и стоимости продукта. Желательно в каждом изучаемом классе программного обеспечения рассматривать бесплатные и условно-бесплатные (с ограниченной функциональностью) программные продукты.

6. Во избежание провокационных ситуаций, при необходимости использования мультимедиа на занятиях, рекомендуется использовать бесплатные продукты.

При использовании программного обеспечения на занятиях преподавателю необходимо помнить, что, несмотря на невысокий авторитет своей профессии в современных реалиях, он все равно остаётся для обучаемых публичным человеком и все его действия могут быть обсуждаемыми в среде его учеников. Исходя из этого, в целях повышения гражданской ответственности в области использования программного обеспечения недопустимы как некомпетентность преподавателя в этом вопросе, так и его противозаконные действия на занятии.

УДК 004.416.6

СТРУКТУРА И ПРОГРАММНО-ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ УЧЕБНО-НАУЧНОЙ СРЕДЫ

«ЭЛЕКТРОФИЗИКА»

Г.П. Аверьянов, В.А. Будкин, В.В. Дмитриева

Национальный исследовательский ядерный университет

«МИФИ»

г. Москва, Россия

Аннотация

В докладе рассматриваются основные концепции разрабатываемого на кафедре электрофизических установок (ЭФУ) НИЯУ МИФИ учебно-научного информационно-вычислительного комплекса. Представлены структура и организация основных профильных дисциплин электрофизики и специфика информационного обеспечения. Базовыми элементами в разрабатываемой информационной среде являются созданные кафедрой интерактивные проблемно-ориентированные пакеты по автоматизации проектирования ЭФУ, ими-

тационному моделированию устройств вакуумной техники и мощной импульсной техники, а также средства поддержки автоматизации эксперимента, которые также представлены в докладе. В качестве операционной среды, используемой в системе, является одна из версий UNIX (GNU LINUX), используемых как на сервере, так и на студенческих рабочих станциях уже около 10 лет. Рассматриваются вопросы удалённого доступа проведения занятий с использованием обучающей системы MOODLE.

Ключевые слова

Информационная среда, программное обеспечение среды, интерактивные проблемно-ориентированные пакеты, GNU LINUX, MOODLE.

1. О кафедре «Электрофизические установки»

Кафедра ЭФУ НИЯУ МИФИ была образована в 1948 году. Основные направления её научной и учебной деятельности связаны, прежде всего, с физикой пучков заряженных частиц и ускорительной техникой. Тем не менее, отдельные разделы этого направления электрофизики имеют много общего с устройствами мощной радиотехники (радиолокационные станции, клистроны, магнетроны, лампы бегущей волны), мощной лазерной техникой и т.п.

Электрофизика, в части ускорительной техники, предполагает создание самых разнообразных установок, значительно различающихся как по своим конструктивным особенностям, так и по областям их применения. Это и уникальные ускорители для фундаментальных исследований, связанных с физикой высоких энергий, типа большого адронного коллайдера (БАК) длиной 27 км в Международном центре ядерных исследований (ЦЕРН, Швейцария). Установки менее грандиозные, но также весьма значительных размеров – электронные синхротроны, циклотроны различных типов и другие подобные установки, используемые как для чисто научных ядерно-физических исследований, так и имеющие прикладной характер, применяются в медицине, материаловедении и других областях. Кроме этого, используется и разрабатывается очень большое количество сравнительно небольших ускорителей прямого действия – высоковольтные электростатические, линейные высокочастотные, сильноточные им-

пульсные на базе формирователей сверхмощных импульсов субнаносекундного диапазона. Это одно из наиболее динамично развивающихся и перспективных направлений ускорительной техники. Область применений подобных установок чрезвычайно широка. К ней относятся, как и в предыдущих установках, научные приложения – изучение физики твёрдого тела, радиационная химия, генерация мощных электромагнитных колебаний, лазеры с накачкой электронным пучком. Нашли применение эти установки в промышленности и народном хозяйстве – дефектоскопии (неразрушающий контроль), контроль за нераспространением ядерных материалов, пастеризация пищевых продуктов, стерилизация медицинского оборудования, обработка сточных вод, дезинсекция зерна, очистка газовых выбросов и т.п.

2. Основные задачи и структура информационной среды

На протяжении уже более 30 лет на кафедре ЭФУ проводятся работы по внедрению в практику различных форм занятий студентов современных информационных технологий на базе созданной в 1977 году кафедральной вычислительной лаборатории.

Работы лаборатории направлены на совершенствование технологии проведения лабораторных практикумов, развитие методов контроля эффективности проводимых занятий и повышения управляемости учебным процессом кафедры в целом, особенно в последнее время, в связи с постоянно изменяющимися требованиями нормативных органов образования.

Имеется ряд причин, побудивших руководство физической кафедры самостоятельно взяться за решение не характерных для кафедры задач. Вот некоторые из них:

- в силу специфики профилирующих кафедральных дисциплин, их освоение связано с решением сложных вычислительных процедур (задачи из области моделирования сплошных сред и др.);

- сложность полномасштабного натурального (физического) моделирования в учебных лабораториях по Вакуумной технике, Мощной импульсной технике, Ускорительной технике и в связи с этим необходимость разработки и введения имитационных компьютерных лабораторных практикумов по этим дисциплинам;

- большие потоки студентов в некоторых кафедральных учебных лабораториях (Информатика, Вакуумная техника, Техника СВЧ, Физика электронных приборов);

- большое количество постоянно изменяющегося учебного и отчётно-нормативного материала, которое кафедра обязана предоставлять регулярно в Учебное управление.

Все это потребовало разработки структуры информационной среды кафедры и размещения её в кафедральном вычислительном центре.

Информационное обеспечение всех видов деятельности кафедры может рассматриваться в двух аспектах:

Иерархический, предметно-ориентированный аспект представлен на Рис. 1.



Рис. 1 а. Уровни информационной среды кафедры (Учебные циклы)

1-й уровень – это 6 учебно-научных циклов, определяющих основные направления подготовки инженеров-физиков-ускорительщиков (Рис. 1 а):

- Ускорители заряженных частиц (УЗЧ);
- Техника сверхвысоких частот (Техника СВЧ);
- Физическая электроника (ФЭ);
- Вакуумная техника (ВТ);
- Электронные системы ускорителей (ЭСУ);
- Информационные системы ускорителей (ИСУ).

2-й уровень – профилирующие дисциплины цикла. На Рис. 1 б представлена информационная структура одной из дисциплин цикла ИСУ (всего в цикле 18 дисциплин).

3-й уровень – виды аудиторных занятий по каждой дисциплине цикла (лекции, семинары, лабораторные работы, курсовые проекты, учебно-исследовательские работы и т.п.).

4-й уровень – базовые элементы дисциплин цикла (программа курса, календарный план занятий, учебно-методическая литература, контрольно-измерительные материалы (КИМ) и т.п.).

5-й уровень – формы контроля по различным видам занятий (вопросы к зачёту, билеты к экзамену, вопросы для самоподготовки к лабораторным работам, тесты и т.д.).

Функциональный или технологический аспект информационной среды, имеет единую применительную практику независимо от цикла:

- уровень удалённого доступа и дистанционного обучения – единая структура, независимая от разновидностей дисциплин;
- контрольно-нормативный, или уровень отчётности кафедры, цикла, преподавателя (планы, отчёты, результаты);
- уровень текущего расписания и распределения дисциплин (по семестрам).

3. Структура технических средств и программное обеспечение (ПО) среды.

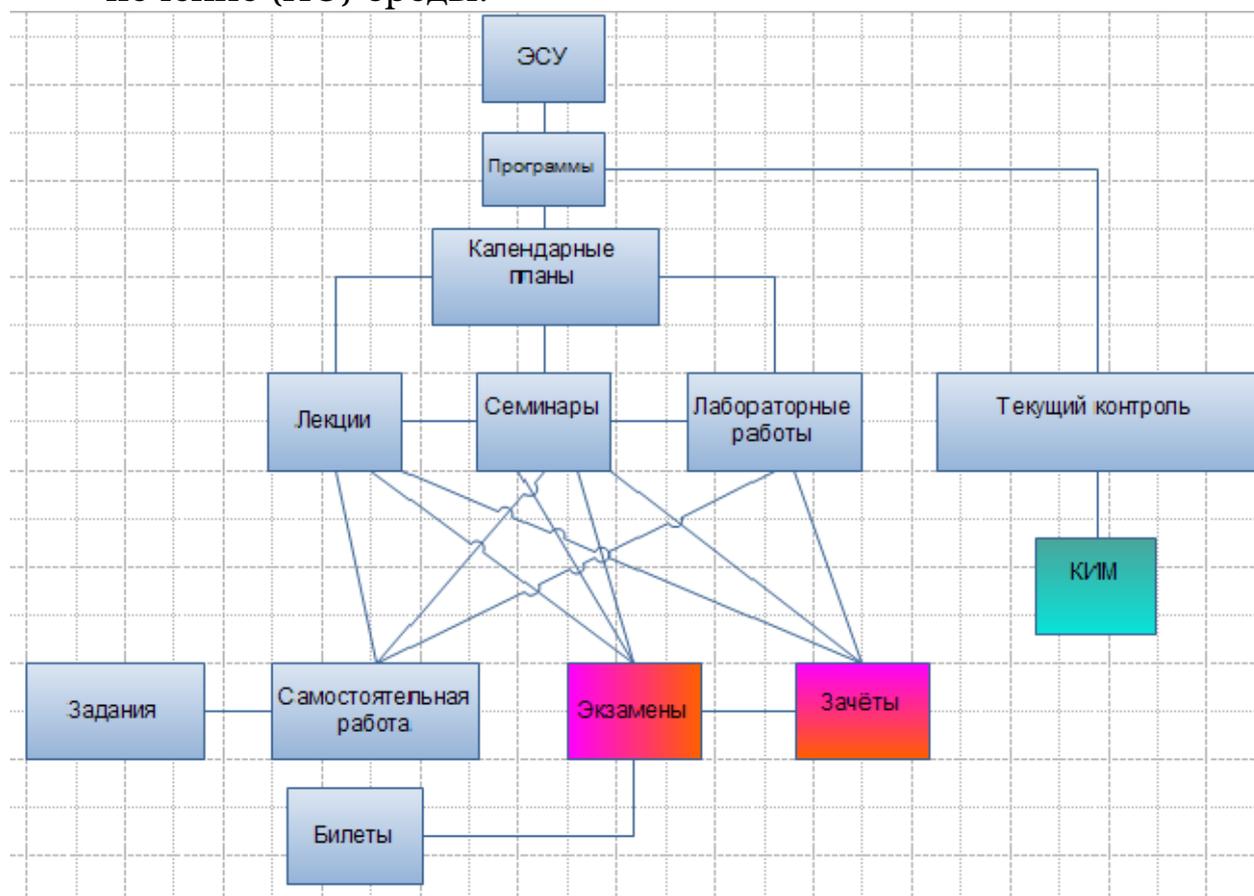


Рис. 16. Информационная структура одной из дисциплин цикла ИСУ

Важнейшим фактором, определяющим выбор структуры технических средств, операционной среды и инструментального ПО для разработки средств дистанционного обучения в создаваемой системе, является обеспечение сохранности и целостности быстро изменяющихся информационных ресурсов, т.е. надёжности работы системы. Актуальность защиты информации доста-

точно понятна, однако основные проблемы и последствия её несоблюдения значительно различаются и зависят от требований, предъявляемых конкретными приложениями. Если вопросы защиты информационных систем органов государственной власти, промышленных предприятий, учреждений кредитно-финансовой сферы, банков и т.п. достаточно очевидны и хорошо проработаны, то особенности информационных систем компьютерных учебных центров и необходимость защиты их информационных ресурсов от случайных и преднамеренных факторов не столь очевидны и нуждаются в проработке, особенно в связи с развитием дистанционного обучения и включения информационных ресурсов в текущие аудиторные занятия. Так, например, в одном из наиболее насыщенных кафедральных лабораторных практикумов в цикле ИСУ проводятся занятия для двадцати групп студентов дневного факультета и примерно для такого же количества групп студентов вечернего факультета. Это означает необходимость оперативной поддержки на центральном сервере около 800 личных архивов студентов. И хотя информация в студенческой базе данных не представляет государственной тайны и не связана с кредитно-денежными операциями, её нарушение или потеря, а также несанкционированный доступ (что часто случается в студенческой среде) могут значительно осложнить проведение учебного процесса.

Важным фактором обеспечения надёжности и устойчивости работы информационной системы является выбор базовой операционной среды. Учитывая имеющийся на кафедре опыт работы вычислительного центра (включая студенческий компьютерный класс) с использованием постоянно изменяющихся ОС фирмы Microsoft (Windows-95, NT, XP) со всеми возникающими в процессе адаптации и дальнейшей эксплуатации трудностями, нами было принято решение по постепенному переходу на ОС UNIX. По-видимому, нет необходимости описывать достоинства этой операционной системы.

На рис.2 представлена схема кафедрального учебно-научного центра со всеми необходимыми внешними связями и аппаратным обеспечением информационной безопасности [1].

Сеть состоит из основного сервера, компьютерного класса с рабочими станциями, сетевого коммутатора и шлюза в сеть Internet. В качестве рабочих ОС применяются Windows XP и Debian GNU/Linux. ОС семейства Windows используются очень широко, поэтому её поддержка была необходима. ОС Debian

GNU/Linux была выбрана как постепенная замена Windows XP. Debian GNU/Linux содержит в своей поставке огромное (более 10000) число программ самой широкой направленности, не требовательна к ресурсам ПК, не требует приобретения лицензий для использования, имеет длительную поддержку обновлений безопасности, хорошую документацию и поддержку со стороны разработчиков и других пользователей. Кроме этого, программа установки Debian GNU/Linux позволяет производить установку и обновление этой ОС с самых различных носителей полностью в автоматическом режиме (режим preseed). Это особенно удобно при обслуживании компьютерного класса.

На данный момент ОС Windows XP используется на 3-х рабочих станциях учебно-научного центра, при этом на этих рабочих станциях также установлена ОС Debian GNU/Linux. На всех остальных компьютерах используется только Debian GNU/Linux.

Центральной задачей основного сервера является предоставление санкционированного доступа ко всем сервисам учебно-научного центра. При этом необходимо было решить задачу создания единого центра авторизации пользователей для двух разнородных типов ОС: GNU/Linux и MS Windows. Для этого была использована база LDAP, хранящая все учётные записи пользователей.

ОС GNU/Linux позволяют использовать различные механизмы авторизации за счёт применения технологий NSS (Name Service Switch) и PAM (Pluggable Authentication Modules). Для авторизации через базу LDAP были использованы модули `libnss-ldap` и `libpam-ldap`. Помимо авторизации, пользователям необходимо иметь доступ к своему домашнему каталогу, хранящемуся на сервере. Для этого используется сетевая файловая система NFS версии 3.

ОС семейства MS Windows используют доменную авторизацию, которая была реализована с использованием программы SAMBA, настроенной на использование базы LDAP. Также SAMBA как файловый сервер для доступа к домашним каталогам пользователей по протоколу CIFS. Таким образом, у пользователя имеется возможность с использованием единой учётной записи получить доступ ко всем рабочим станциям учебно-научного центра и своему домашнему каталогу, независимо от типа установленной или загруженной на данной рабочей станции ОС.

Кроме этого имеется возможность получить безопасный доступ к домашнему каталогу пользователя из сети Internet с при-

менением расширения протокола SSH (Secure Shell) — SFTP (Secure File Transfer Protocol). Во всех случаях для разграничения прав доступа к файлам используется стандартный механизм разграничения прав в системе GNU/Linux, а в особых случаях расширенный метод Unix ACL (Access Control Lists).

Помимо авторизации и файлового хранилища, основной сервер предоставляет доступ к электронной почте, Web-серверу и сетевым принтерам. Для проверки доступа к ним, также, применяется база LDAP.

Электронная почта обеспечивает авторизованную отправку и приём писем по протоколам SMTP, IMAP и POP3. Уязвимым местом этих протоколов является передача учётных данных пользователя в открытом виде по сети. Для предотвращения перехвата этих данных связь с сервером осуществляется с использованием зашифрованного соединения SSL. Для этого на сервере используются программы Postfix и CourierIMAP.

Доступ к принтерам обеспечивает программа CUPS с использованием протокола IPP (Internet Printing Protocol).

Web-сервер обеспечивает работу сайта кафедры ЭФУ (www.accel.ru), на котором размещена учебная информация, программы для самостоятельного выполнения лабораторных работ студентами и портал дистанционного обучения. Работу Web-сервера обеспечивает программа Apache.

Доступ к глобальной сети Internet обеспечивает отдельный шлюз. В качестве шлюза в сеть Internet используется компьютер с двумя сетевыми интерфейсами. Шлюз обеспечивает работу локальных компьютеров в сети Internet с использованием механизма подмены адресов NAT (Network Address Translation), одновременно надежно защищая их от внешних воздействий. Для доступа к локальным ресурсам учебно-научного центра используется механизм Port Forwarding, который открывает доступ из внешней сети к IP портам основного сервера, используемых для работы программ SSH, электронной почты и Web-сервера. Работу NAT и Port Forwarding обеспечивает ядро ОС GNU/Linux.

Для обеспечения безопасности всего учебно-научного центра программы, доступ к которым возможен из сети Internet, выполняются в изолированном окружении (chroot), что предотвращает доступ ко всей системе при взломе одной из них.

Программное обеспечение безопасности хранения информации, кроме традиционных средств ОС, включает средства

инструментальной среды Moodle, на основе которой предполагается построение всех форм взаимодействия кафедральных циклов.

Можно выделить три уровня доступа к основным курсам кафедрального информационно-вычислительного центра:

1 – локальная сеть кафедрального компьютерного класса (20 ПК), в состав которого входит базовый UNIX – сервер (accel.ru);

2 – сеть кампуса МИФИ, включающая компьютеры кафедральных учебно-исследовательских лабораторий, общеинститутские компьютерные классы, ПК студенческих общежитий и т.п.;

3 – домашние компьютеры преподавателей и студентов, обучающихся в рамках учебных циклов кафедры.

Каждый уровень доступа предполагает комплекс программно-аппаратных и административных мероприятий по защите как от преднамеренных, так и от непреднамеренных воздействий.

Что касается непреднамеренных воздействий, которые связаны с ошибками обслуживающего персонала, сбоев в электросети и т.п., то предусмотренная защита от них достаточно традиционна. Это, прежде всего:

1) резервное копирование личных архивов пользователей (под ответственность пользователей), регулярное копирование наиболее важного и часто изменяемого как общеинститутского, так и прикладного ПО на различные внешние носители;

2) избыточность данных обеспечивается как RAID массивами центрального кафедрального сервера, так и отдельным резервным сервером (UNIX) в сети компьютерных классов МИФИ;

3) защита от сбоев в силовой сети осуществляется общепринятыми средствами (UPS, SPS и т.п.)

Основным средством защиты от нежелательных (преднамеренных) внешних воздействий является брандмауэр, представляющий комплекс программно-аппаратных средств, ограничивающих доступ к информационным ресурсам центра (рис. 2. шлюз-Internet). Связь с сервером осуществляется с использованием зашифрованного соединения SSL. Для этого используются программы Postfix и Courier IMAD.

Определённую роль в общем комплексе мер информационной безопасности играет Proxu схемы с аутентификацией пользователей, которая создаёт соединения внешней сети (Internet) с конечным адресатом через промежуточный сервер, и таким об-

разом кафедральная локальная сеть ПК представляет приватную сеть с виртуальными IP-адресами.

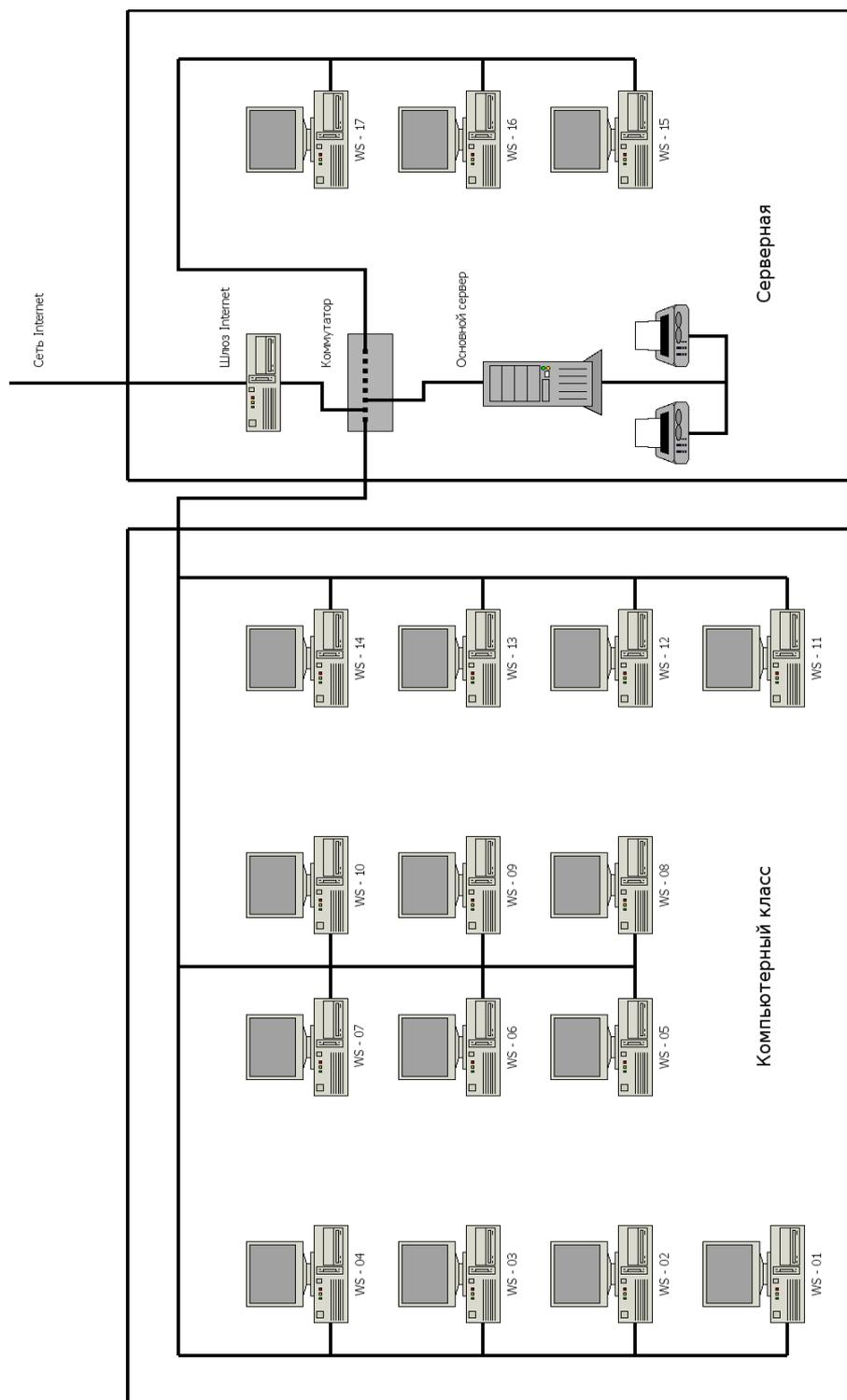


Рис.2. Кафедральная локальная сеть

4. Об удалённом доступе и дистанционном обучении.

Средства удалённого доступа студентов к необходимым ресурсам кафедральной информационной среды являются очень важным дополнением традиционных аудиторных занятий. Про-

ведение регламентированных аудиторных занятий, особенно лабораторных практикумов в «привязке» к конкретной аудитории, очень часто приводит к накоплению «задолженностей» у студентов и «напряжённости» во время зачётной сессии.

Это связано как с разной скоростью выполнения заданий в силу разной подготовленности студентов, так и с пропусками занятий по вполне объективным причинам (болезни, командировки у студентов вечернего факультета и пр.). В связи с этим появление инструментальной среды дистанционного обучения явилось удачным дополнением к традиционным видам занятий, в ряде случаев значительно (ко всему прочему, указанному выше) расширяющим объем изучаемого материала и качество его освоения. Хотя, безусловно, дистанционное обучение может рассматриваться гораздо в более широком аспекте.

На кафедре ЭФУ уже на протяжении около 10 лет для студентов предусмотрен удалённый доступ к необходимым учебным ресурсам, расположенным на Web-сайте ascel.ru. Однако на первом этапе этот доступ имел достаточно примитивную организацию. На сайте располагались необходимые методические материалы, задания по различным видам лабораторных практикумов, некоторые теоретические материалы. В последние годы проводится внедрение многофункциональной обучающей среды с широкими возможностями.

За основу инструментального ПО разработки портала дистанционного обучения принята виртуальная обучающая система Moodle [2], созданная в рамках проекта Collaboration Across Borders, получившего поддержку Европейской образовательной программы Socrates — Minerva.

Moodle — это программный продукт, базирующийся на Internet (позволяющий создавать Web-сайты), распространяется бесплатно в качестве программного обеспечения с открытым кодом (Open Source), работа в этой среде предполагает выполнение двух противоречивых требований. С одной стороны, открытость системы с предоставлением обширных возможностей сети Internet – сотрудничество и дискуссии с другими пользователями, расширенные возможности изучаемого материала и конструирование собственных знаний, возможность загрузки дополнительных компонент по изучаемой дисциплине, т.е. это Web-технология, которая может использоваться как в режиме реального времени (on line), так и в автономном режиме (off line). С другой стороны, система предоставляет права защиты от не-

санкционированного доступа – просмотра, а в ряде случаев и нарушения таких материалов, как тесты, критерии оценки знаний, журналы успеваемости и посещаемости в сети и т.п. Как и любая защита, все эти мероприятия несколько снижают функциональность системы. Работа различных категорий пользователей строго регламентирована, права доступа к различным ресурсам системы устанавливаются авторизацией пользователей (каждый участник этой системы имеет свой login и пароль, которые и определяют их возможности).

Безусловно, удалённый доступ студентов и преподавателей к информационным материалам по различным дисциплинам кафедры электрофизики и выполнение лабораторных практикумов «на дому» не является альтернативой традиционных аудиторных занятий, а служит лишь дополнительным средством расширения возможностей самостоятельной работы с использованием современных информационных и телекоммуникационных средств глобальной сети Internet.

В то же самое время внедрение этой системы позволяет повысить управляемость учебного процесса и степень объективности оценки работы и знаний студентов преподавателями. Текущая информация об «активности» студентов и результативности их работы, степени освоения курса, оценка знания на текущий момент, затраченное время, количество посещений (выход на сайт) сохраняются в базе данных в течение всего времени аудиторных занятий по соответствующему курсу.

Безусловно, каждый разработанный в среде Moodle курс требует индивидуальной проработки, оценки целесообразности выбора различных функций среды (глоссарий, ресурс, задание, форум, урок, тест и т.д.).

В настоящее время на кафедре проводится пробная эксплуатация этой системы в двух лабораторных практикумах – «Автоматизация проектирования ЭФУ» и «Мощная импульсная техника». Уже первые результаты показывают, что система позволяет увеличить объем изучаемого материала примерно на 30 % за счёт сокращения времени выполнения лабораторных практикумов в традиционном режиме.

Используемые источники:

1. Аверьянов Г.П., Будкин В.А., Дмитриева В.В., Коршунов А.М., Фадеев А.А. Обеспечение информационной безопасности сетевого учебно-научного центра по электрофизике. БИТ № 3, 2009.

О СИСТЕМЕ ОЦЕНОК ТВОРЧЕСКИХ ЗАДАНИЙ, ВЫПОЛНЯЕМЫХ С ПОМОЩЬЮ РАСТРОВОГО ГРАФИЧЕСКОГО РЕДАКТОРА GIMP

Т.А. Панюкова

Южно-Уральский государственный университет

г. Челябинск, Россия

E-mail: kwark@mail.ru

Аннотация

В статье приведён перечень заданий для работы с редактором растровой графики GIMP, предлагаемых студентам в качестве курсовых и семестровых работ в рамках дисциплины «Компьютерные издательские системы». Предложены критерии оценок заданий. Разработана шкала оценок.

Ключевые слова

Свободное программное обеспечение, растровая графика, растровый графический редактор GIMP.

В настоящее время программы для обработки и предпечатной подготовки цветных изображений используются повсеместно в профессиональной работе при создании презентационных материалов, документов, содержащих деловую графику, высококачественных печатных изданий. Тем не менее, современные технические и программные средства для работы с графикой привлекательны также и для многочисленных рядовых пользователей, обрабатывающих фотографии, сканированные изображения и решающих другие задачи компьютерной графики.

Наиболее распространёнными среди программных продуктов, позволяющих обрабатывать изображения, являются Adobe Photoshop и его свободный аналог GIMP. Эти программы используются для профессиональной предпечатной подготовки, так как обеспечивают высокое качество графического материала. Дружественный характер интерфейса позволяет использовать эти программы для создания и редактирования графического материала и в любительской среде [1].

Умения грамотно представить некий материал, оформить, подготовить к публикации данные, составить презентацию и т.п. становятся необходимым качеством выпускника вуза, поэтому в настоящее время учебные планы многих, казалось бы, «не творческих» специальностей содержат разделы, посвящённые изучению возможностей пакетов прикладных программ, правилам допечатной подготовки изображений. Например, зачастую необходимо экспортировать графическую информацию в формат *.eps (Encapsulated PostScript) для дальнейшего включения в документы и презентации, создаваемые в LaTeX. Такое преобразование можно сделать достаточно просто в программе GIMP.

Ниже приведён пример составленных заданий для курса «Компьютерные издательские системы» (34 часа), преподаваемого студентам специальностей «Математические методы в экономике» и «Статистика». Курс ориентирован на использование GIMP. Для изучения возможностей графического редактора GIMP материал скомпонован в 4 темы, на которые отводится 14 часов, из них 8 часов лабораторных работ и 6 часов самостоятельной работы [2].

Программа **GNU Image Manipulation Program (GIMP)** – растровый графический редактор, в котором частично поддерживается и векторная графика. **Типичные задачи**, которые можно решать при помощи GIMP, аналогичны тем, что решаются с помощью Adobe Photoshop, и включают в себя создание графических изображений, масштабирование и кадрирование фотографий, раскраску, комбинирование изображений с использованием слоёв, ретуширование и преобразование изображений в различные форматы.

Курс в вышеперечисленном объёме занятий содержит следующие разделы.

1. Основные концепции работы в The GIMP, сравнение возможностей GIMP и Photoshop. Проводится обзор основных возможностей программ Adobe Photoshop и GIMP, коротко рассматривается интерфейс программ, указываются основные приёмы работы с ними. Рассматриваются основные характеристики изображения: разрешение и размер, приводятся задачи по расчёту планируемого размера и разрешения изображения, а также количества места на жёстком диске, требуемого для сохранения файла изображения в различных форматах.

Работа с кистями. Проводится обзор большой группы инструментов, в основе которых лежит кисть, приводятся примеры редактирования имеющихся кистей и создания новых.

2. Выделенные области в GIMP. Основные принципы создания коллажей. Основы цветокоррекции. Редактирование сканированных изображений.

Выделение — основной инструмент при обработке графики. Рассматриваются все возможные инструменты GIMP, их особенности и различия. В качестве примера рассматривается фотоколлаж. Создание подобного коллажа выносится в качестве одного из заданий для **курсовой работы**: подобрать фотографию с пейзажем и портретное изображение; составить из них коллаж таким образом, чтобы портретное изображение находилось хотя бы за одним объектом фотографии с пейзажем.

В этой же лекции рассматриваются основы цветокоррекции: показаны примеры исправления цветовых неточностей в созданном коллаже, создаётся тень для вставленного объекта, приводятся примеры редактирования сканированных и фотографированных изображений, подготовки изображения для размещения его в текстовом документе и последующей печати.

3. Работа с текстом и контурами в GIMP. Создание листовок и приглашений.

Лекция посвящена особенностям редактирования текстовых слоев (форматирование однострочного и многострочного текста, форматирование символов и абзацев) и эффектам, используемым для этих целей. Все рассмотренные особенности проиллюстрированы примерами создания открыток, плакатов и объявлений.

4. Фильтры и GIF-анимация в GIMP.

Основные принципы работы с анимацией. Фильтры для стилизации фотографий, искажения изображений.

Примеры творческих заданий для работы в GIMP

Разработанные задания для самостоятельной работы студентов составлены таким образом, что они охватывают знакомство с основными возможностями программы и принципами работы GIMP в рамках соответствующих конкретных разделов программы изучаемой дисциплины.

Составлено *три творческих задания*. Для повышения качества их выполнения, с одной стороны, целесообразно не ограничивать обучающихся некоторым множеством операций и элементов, выполняемых над тем или иным изображением. С дру-

гой стороны, существует такая категория студентов, как *минималисты*, которые выполняют задание только для того, чтобы получить зачёт или оценку любого уровня: работа выполняется по минимуму, формально соответствует приведённым в постановке задачи требованиям, но при этом имеет очень низкое качество. Причины такого минимализма могут быть различными: нежелание учиться чему-либо (формальное отношение к учёбе вообще), отсутствие творческого подхода, надежда на «авось» и пр.

При недостаточной детализации требований к заданию студент, работающий по минимуму, не осваивает многих основных принципов работы с программой. Чтобы предотвратить формальный подход к выполнению задания, разработаны критерии выставления оценок, которые приведены в данной статье.

В каждое задание введены *усложняющие элементы*. Они, с одной стороны, усложняют задачу студенту, заставляют глубже изучить программу, а с другой – ограничивают его возможности. Эти ограничения несколько сковывают студентов-энтузиастов, увлечённых выполнением задания. Обычно такие люди самостоятельно осваивают программное обеспечение достаточно глубоко и представляют работы высокого качества. Существует и ещё одна группа студентов, которых условно можно назвать *отличниками*. Они выполняют полученное задание досконально с единственной целью — получить высокую оценку. Их работы обычно не так интересны, как работы энтузиастов, но при этом зачастую имеют достаточно высокое качество и формально полностью соответствуют выданному заданию.

Очевидно, что выполненное задание должно удовлетворять общепринятым правилам построения композиции, цветопередачи, работы со световыми источниками. Однако эти правила выходят за рамки оценки умений работы с *техническим инструментарием программы*. Поэтому разработка *шкалы критериев для оценки творческих заданий* по компьютерной графике является актуальной задачей.

Оценку за каждое задание разобьём на *две составляющие*: *техническое исполнение* (степень ознакомления с инструментарием GIMP) и *субъективный критерий* (композиция, качество исполнения).

Задание 1. Рисование кистями. Нарисовать изображение, состоящее из *фона* и как минимум *трёх пересекающихся*

между собой слоев. При создании изображения можно пользоваться инструментами на основе кисти: кистями, ластиком, инструментами размытия и осветления/затемнения. Запрещено использовать заливку и выделенные области. При его выполнении обучающийся должен освоить, каким образом создаются многослойные изображения, научиться раскладывать объекты по слоям, уметь использовать кисти различной формы и разновидности.

Рассмотрим систему оценок для этого задания [2].

Так как в число обязательных элементов следует включить *разбиение композиции на слои*, для этого действия назначим следующую систему оценки:

- 3 – чёткое разбиение, один объект – один слой;
- 2 – на одном слое находятся элементы различных объектов;
- 1 – изображение разбито на слои формальным образом;
- 0 – некорректное разбиение на слои либо отсутствие разбиения.

Всего известно *9 основных видов кистей*: аэрограф, карандаш, кисть, осветление, затемнение, палец, размытие, резкость и ластик. Очевидно, что не всегда есть возможность использовать все эти инструменты при создании одного изображения.

Заметим, что все инструменты можно разбить на *4 группы*: кисти, осветление/затемнение, размытие/резкость и палец. Поэтому для того, чтобы система критериев для оценки охватывала все возможные виды используемых кистей, достаточно в формулировке задания написать, что при его выполнении *необходимо использовать кисти как минимум из трех разных групп*.

Для каждого из использованных инструментов можно составить следующую шкалу:

- 4 – созданы и использованы пользовательские кисти;
- 3 – использование стандартных кистей разной формы и радиуса;
- 2 – использование различных форм кистей стандартного размера;
- 1 – использование одного вида кисти различного размера;
- 0 – использование кисти одного вида стандартного радиуса.

Таким образом, минимальный возможный балл, достаточный для получения отличной оценки, равен $3+4*3=15$. В ре-

зультате, для получения зачёта по заданию на использование кистей необходимо набрать как минимум 15 баллов.

Если за задание предусмотрена оценка, то в случае, когда обучающийся при выполнении задания набирает 8 – 9 баллов, он получает оценку «удовлетворительно», 10 – 12 баллов – оценку «хорошо», 13 баллов и выше – оценку «отлично».

Задание 2. Фотоколлаж. Подобрать две фотографии: пейзаж и портретное изображение; составить из них коллаж таким образом, чтобы портретное изображение находилось хотя бы за одним объектом фотографии с пейзажем. Проанализировать все ошибки тона и цвета и исправить их. Создать тени для всех вставленных фрагментов, если это необходимо.

Для этого задания можно составить следующую группу из трёх шкал.

Во-первых, для определения *уровня сложности* фотоколлажа выделим следующие баллы:

4 – вставленные элементы образуют сложную композицию и находятся за объектами, первоначально расположенными на фоновом изображении;

3 – вставленные фрагменты образуют простую композицию, но находятся за объектами, изначально расположенными на фоне;

2 – вставленные фрагменты являются аппликацией, но образуют довольно сложную целостную композицию;

1 – вставленные фрагменты являются аппликацией и не перекрывают друг друга;

0 – бессмысленное размещение вставленных фрагментов.

Во-вторых, для оценки *качества исполнения* определим следующую шкалу:

4 – вставленные фрагменты вырезаны без дефектов и соответствуют по цвету/тону основной композиции;

3 – вставленные фрагменты вырезаны без дефектов, но цветовая и тоновая коррекция проведены неверно либо не проведены вообще;

2 – вставленные фрагменты вырезаны с видимыми дефектами, цветовая и тоновая коррекция проведены не полностью;

1 – вставленные фрагменты вырезаны с дефектами, коррекция не проведена;

0 – вставленные фрагменты имеют значительные дефекты.

Наконец, введём критерий для оценки *используемых спецэффектов*. Здесь можно добавлять по 2–3 балла за каждый из перечисленных ниже эффектов в зависимости от качества исполнения, обоснованности использования и уровня сложности эффекта:

- создание тени (дубликат слоя, изменение режимов наложения, сложные трансформации) – максимум 9 баллов;
- использование встроенных фильтров – максимум 3 балла;
- изменение режимов наложения для отдельных слоев – максимум 3 балла;
- использование трансформаций (кроме масштаба и поворота) — максимум 3 балла.

Учитывая все вышеперечисленные критерии, максимально возможный балл, полученный за выполнение задания, будет равен $4+4+9+3+3+3 = 27$. Для получения удовлетворительной оценки необходимо набрать 17 баллов, хорошей — 19, отличной — 22. В шкалу можно добавить и бонусные баллы за творческий подход к выполнению задания.

Задание 3. Афиша. Придумать и создать афишу, на которой присутствует как однострочная, так и многострочная текстовая информация. Применить эффекты для акцентирования внимания на текстовой информации.

С одной стороны, задание очень простое, с другой — его оценивать тяжелее всего, потому что формальных критериев для него предельно мало, и одним из основных факторов при выставлении оценки является восприятие изображения (например, не мешают ли созданные эффекты читать текст).

Для каждого из заданий можно составить и другие шкалы, выделить иные критерии, учитывающие особенности обучения на той или иной специальности, использование определённого инструментария и пр. Количество баллов, необходимое для получения оценки, можно вычислить как процент от максимального балла. Например, 50–60% – «удовлетворительно», 60–80% – «хорошо», 81% и выше – «отлично». Ещё один способ — просуммировать баллы, получаемые за обязательный минимум.

Система оценок должна быть открытой, чтобы каждый мог выбрать те элементы, которые он будет реализовывать в своей работе.

Разработанная система критериев позволила значительно повысить качество выполняемых заданий, формализовать вы-

ставление оценки, а в некоторых случаях и уменьшить возможное проявление субъективности со стороны преподавателя.

Используемые источники:

1. Панюкова Т.А. GIMP и Adobe Photoshop: лекции по растровой графике. Учебник. – М.: Либроком, 2010. – 280 с.

2. Панюкова Т.А. Разработка системы оценок и контрольных заданий для работы в графическом редакторе GIMP. – Свободное программное обеспечение в образовании: сборник трудов Всероссийской конференции (Челябинск, 25–26 марта, 2009 г.) / Челябинск: Изд-во ЮУрГУ, 2009. – с. 87–91.

УДК 378.016:004:51(045)

ОПЫТ ИЗУЧЕНИЯ СПЕЦИАЛЬНЫХ ДИСЦИПЛИН В ПЕДАГОГИЧЕСКОМ ВУЗЕ НА БАЗЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Т.В. Кормилицына

ГОУ ВПО «МордГПИ имени М.Е. Евсевьева»

г. Саранск, Россия

E-mail: kortv58@mail.ru

Аннотация

Обсуждается возможность использования свободного программного обеспечения в организации учебного процесса на различных факультетах педагогического вуза.

Ключевые слова

Свободное программное обеспечение, программы, системы символьной математики.

Свободное программное обеспечение развивалось в направлении от инструментальных программ к системным программам и далее, к прикладным, сегодняшнее положение дел, при котором все перечисленные приложения имеют свободную реализацию, сложилось не сразу. Однако уже в течение нескольких лет основная масса свободных прикладных программ находится на уровне, достаточном для их применения в учебном процессе, и в педагогическом сообществе постепенно накапливается опыт их использования.

В нашем вузе свободное программное обеспечение применяется достаточно широко и с разными целями.

Перечислим причины, побудившие обратиться к свободному программному обеспечению. Вначале мы пытались реализовать государственные образовательные стандарты для специальности «Информатика» по основным дисциплинам на основе указанного в них программного обеспечения. Так, по дисциплине «Информационные технологии в математике» составителями стандартов предполагается изучение мощнейших систем символьной математики, таких как Maple, Mathematica, и не менее мощных вычислительных систем MathCad, Matlab.

Не хочется даже приводить суммы, которые должен потратить вуз для обеспечения программного сопровождения только указанной дисциплины, а подобные проблемы возникают практически для каждой дисциплины специальности «Информатика», и не только для этой специальности!

Некоторое время мы пытались работать в бесплатных демо-версиях систем, однако демо-версия, например системы Maple, даёт чрезвычайно слабое представление о возможностях системы и просто неприменима для решения хоть сколько-то серьёзных математических задач. Для других систем демо-версии мы вообще не обнаружили.

Первая система свободного программного обеспечения, с которой мы стали работать, – система GAP. Как мы узнали, изначально система была спроектирована студентами, которые решили автоматизировать процесс решения задач для дипломного проекта. Мы использовали возможности системы GAP для решения задач теории групп в рамках вначале курсового, а затем и дипломного проектов. Результаты исследования получились вполне приличные, сам дипломный проект размещен на образовательном математическом сайте exponenta.ru по адресу [3]. В работе «Алгоритмы решения задач теории групп в системах компьютерной алгебры» всесторонне освещается изучаемая тема, рассматриваются этапы создания систем компьютерной математики, описываются язык, структура, возможности и особенности системы GAP, а также рассматриваются некоторые задачи теории групп и приводятся алгоритмы их решения.

После неудачных попыток поиска демо-версий других систем мы узнали о системе символьной математики Maxima (первоначальное название Macsyma). Надо сказать, что опыт первоначальных установок версий системы был не совсем удачным,

однако вот уже год мы прекрасно пользуемся возможностями системы.

Студентов подкупают удобный графический интерфейс, простота входного языка наряду с мощнейшими возможностями системы. Чаще всего на лабораторных занятиях мы работаем в основной программе без подключения пакетов расширений, так как основная трудность – отсутствие учебной литературы – не даёт возможности активно использовать пакеты расширений для решения прикладных задач. Тем не менее, студенты начали выполнять задачи моделирования с помощью пакета `distrib`. Для методического сопровождения изучения дисциплины составлено учебное пособие [1], доступны методические материалы для проведения лабораторных работ с использованием систем свободного программного обеспечения [2].

Результаты исследований обсуждались на занятиях научно-исследовательской группы «Проблемы использования свободного программного обеспечения в образовании» и докладывались в рамках научно-практической конференции нашего вуза, по итогам которой был опубликован сборник статей студентов с нашей статьёй [5].

Цели выступлений были, в основном, пропагандистские – мы хотели познакомить с системами `Maxima` и `Scilab` как можно больше студентов разных курсов и специальностей. Следует заметить, что после этого интерес к системам проявили не только студенты, но и преподаватели кафедр математики и физики. Особенно привлекли преподавателей графические возможности системы – построения поверхностей и работа с ними: повороты и преобразования, презентацию проводили на интерактивной доске.

В учебном процессе важным оказалось использование систем `Maxima` и `Scilab` при изучении дисциплины «Численные методы». Здесь оказались востребованными как аналитические, так и вычислительные возможности систем на основе встроенных алгоритмов, например, при решении задач аппроксимации функций для построения интерполяционных многочленов. Использовались системы и для вычисления неопределённых интегралов и определённых для проверки результатов, полученных студентами различными численными методами.

На занятиях научно-исследовательской группы студенты проводили вычислительные эксперименты по сравнению вычис-

лительных возможностей систем Gap, Maxima, Scilab. Результаты приведены в [3].

Чтоб окончательно убедить студентов в наличии у рассматриваемых систем мощных вычислительных возможностей, мы провели вычислительный эксперимент по расчёту значения факториала во всех системах. Обычно именно факториалы позволяют обнаружить «слабые» алгоритмические моменты систем. Для вычисления факториала в системе Maxima мы использовали встроенную функцию.

При поиске максимального аргумента, для которого значение функции вычисляется системой, применяли метод дихотомии. Начальный отрезок натурального ряда был выбран в виде $[0, 50000]$. Одновременно оценивали время вычисления системой выбранного значения.

Как было установлено в результате эксперимента, максимальный аргумент, воспринимаемый системой, равен 13512. При этом система затрачивает на вычисления 0,03 секунды (расчёты проводились с процессором типа Pentium-4). Система давала количество цифр в значении факториала – 49889. Отклик системы в случае отказа вычислений занимал от 0,08 секунды (для числа 13513, следующего за максимально вычисляемым аргументом) до 0,19 секунды (для числа 50000) выдавался в виде сообщения об отказе.

Далее эксперимент провели в системе Gap, хотя особых результатов для неё не ожидали, ведь основное назначение системы – решение задач теории групп. Система и материалы по её изучению в основном англоязычные.

Несмотря на направленность системы на конкретный класс математических задач, функция для вычисления факториала $\text{Factorial}(N)$ является встроенной в ядро системы.

Как показал эксперимент, максимальный аргумент для этой функции равен 1712, при этом система выводит результат на 62 строках по 78 позициям, причём последние 428 цифр – нули. При попытке вычислить $\text{Factorial}(1713)$ система выдаёт сообщение «an integer too large to be printed», что мы расценили как предел возможностей вычислений с целыми числами.

Систему SciLab непосредственно к системам символьных вычислений не относят, но среди свободных специальных математических систем она занимает такое же место, как система MatLab среди систем лицензионного класса. Вычислительные возможности системы огромны, она имеет большую библиотеку

функций, но для проведения нашего эксперимента пришлось работать с функцией пользователя, так как среди функций из библиотек для вычисления факториала специальной функции не оказалось. После задания функции в виде

```
function [x]=fact(k)
k=int(k)
if k<1 then k=1,
end
x=1;
for j=1:k, x=x*j;
end
endfunction
```

в результате эксперимента нами было найдено число 170 – максимальное значение аргумента созданной функции. Результат система выдавала в приближенном виде как 7.257+306.

Попытки вычислить fact(171) приводят к сообщению «Inf».

Таким образом, «лидером» в нашем эксперименте стала система Maxima, что ещё раз подтверждает её статус реальной альтернативы признанным лицензионным системам. Результаты других систем вовсе не говорят об их ущербности, а лишь подтверждают идею о том, что системы должны уметь решать различные задачи, но могут делать это по-разному в силу своей специфики.

Исследовательские цели реализовали при выполнении курсовых проектов. Для разработки предлагались темы моделирования решений дифференциальных уравнений, изучения алгоритмов решения задач линейной алгебры и другие.

Для студентов не математических специальностей мы применяли системы свободного программного обеспечения для уточнения у них основных математических понятий и для решения вычислительных задач по дисциплине «Математика и информатика». Заметим также, что практически весь институт работает с OpenOffice.org.

Укажем возможные формы использования свободного программного обеспечения для организации самостоятельной работы студентов физико-математического факультета. Удачным оказалось применение системы Gar для самостоятельной работы студентов, например, по дисциплинам «Теория чисел», «Дискретная математика», «Элементы абстрактной и компьютерной алгебры». Эта система оказалась незаменимой для проверки результатов контрольных и индивидуальных работ. Так что реали-

зованная в системе рациональная арифметика полностью отвечает решению задач по исследованию простоты чисел и других их характеристик.

Таким образом, свободное программное обеспечение мы используем в различных формах учебного процесса: при проведении лекций и лабораторных работ в рамках специальных дисциплин, для выполнения курсовых и дипломных проектов, в работе научно-исследовательской группы, для организации самостоятельной работы студентов по различным дисциплинам.

Системы свободного программного обеспечения представляют исследователям возможности, сравнимые с возможностями лицензированного программного обеспечения. Потенциал применения таких систем как для научно-исследовательской работы, так и при обучении специальным и другим дисциплинам, в том числе математике и физике, значителен. Особенно привлекательны возможности проведения аналитических вычислений и серьезные средства визуализации.

Используемые источники:

1. Кормилицына Т.В. Информационные технологии в математике; учебное пособие. – Саранск, Мордов. гос. пед. ин-т, 2009. – 170 с.
2. Кормилицына Т.В. Лабораторный практикум по информационным технологиям [Электронный ресурс]. – Режим доступа <http://www.exponenta.ru/educat/systemat/kormilicyna/index.asp>
3. Мандров, А.В. Организация вычислительного эксперимента в специальных математических системах / Всероссийские научные Зворыкинские чтения - I. Всероссийская молодёжная научная конференция «Научный потенциал молодёжи - будущее России»: сб. тез. докл. Т. 2. – Муром: Изд.-полиграфический центр МИ ВлГУ, 2009, с 98-99.
4. Миронова С. М. Алгоритмы решения задач теории групп в системах компьютерной алгебры [Электронный ресурс]. – <http://www.exponenta.ru/educat/referat/XVIkonkurs/4/index.asp>
5. Сюмкин С.М, Агафонова О.А. Компьютерное моделирование в системе МАХИМА / Молодёжь и наука: проблемы современного образования : сб. ст. Всерос. науч.-прак. конф.: в 3 ч. Ч.1 Мордов. гос. пед. ин-т. – Саранск, 2009. С. 92-96.

ПРОВЕДЕНИЕ КУРСОВ ПОВЫШЕНИЯ КВАЛИФИКАЦИИ С ИСПОЛЬЗОВАНИЕМ LMS MOODLE

Р.Р. Яфаева, Л.Д. Ситникова

ГОУ ВПО ТГПУ им. Л. Н. Толстого

г. Тула, Россия

E-mail: tgpu@tula.net

Аннотация

В статье представлены основные положения социальной открытой системы профессионального образования, рассмотрены принципы модульного обучения. Описан действующий электронный курс для дистанционного обучения, направленный на проведение курсов повышения квалификации работников образования.

Ключевые слова

Повышение квалификации в образовании, профессиональное обучение, электронное обучение, модульное обучение, дистанционные курсы, свободное программное обеспечение.

В современных условиях развития общества необходим пересмотр образовательных целей и создание образовательной системы, которая соединяла бы в себе лучшие образцы человеческой деятельности, формировала бы мировоззрение общества, основанное на принципах единства существования человека и природы. В основе такой системы должны лежать фундаментальные знания, среди которых особое значение приобретают гуманитарные и междисциплинарные знания, которые формируют целостно-эволюционное мышление человека. Особенностью нового мировоззрения является совокупность общечеловеческих ценностей, составляющих основу открытого гражданского общества, в котором рождается коллективный разум как новое качество в отношениях между людьми. Формирование коллективного разума приводит к идее создания таких образовательных структур, которые бы вовлекали в образовательный процесс всех членов мирового сообщества.

Определяющими становятся коммуникационные факторы, зависящие от средств и технологий обмена информацией. Информационные технологии и их развитие являются ключевой проблемой в стратегическом планировании. Возможности сети Интернет и широкое внедрение компьютерных систем в учебный процесс позволяют сделать его гибким с учётом индивидуальных особенностей обучаемых.

Система профессионального образования является неотъемлемой частью экономики страны. Это отражается на целях и содержании профессионального образования, предполагает способы его модернизации, ориентируемые на потребности экономики и рынка труда, нового общества и его гражданских институтов: заказчики кадров, работодатели, службы занятости, профсоюзы, ассоциации защиты инвалидов и другие.

Социально открытый характер учреждений и всей системы профессионального образования, ориентированных на подготовку кадров по заказам рынка труда, работодателей становится необходимостью и условием их саморазвития, социально-экономической и педагогической эффективности. В этой связи представляется актуальной и требующей развития выдвинутая И.П. Смирновым парадигма профессионального образования, основанная на социальном партнёрстве: «Работодатель определяет чему учить, образовательные учреждения – как учить!».

Социально открытая система профессионального образования – это система, которая активно взаимодействует с социумом через механизм социального партнёрства с ведущей ролью в нем работодателей способна адекватно реагировать на динамику развития рынка труда, структуру востребуемых профессий, а также на диктуемые обществом экономические и социально-культурные приоритеты.

Цель открытой системы – дать возможность каждому наиболее полно реализовать свои образовательные потребности.

Огромная сумма знаний, накопившаяся к настоящему времени, и все возрастающий темп роста научной информации делают невозможным использование традиционных способов усвоения знаний путём увеличения числа дисциплин и их объёма. Аналогичная проблема возникает при решении задачи оперативной подготовки и переподготовки кадров в соответствии с потребностями экономики. Здесь нужны совершенно новые способы систематизации знаний и новые образовательные технологии, способные интегрировать различные науки с целью до-

стижения нового качества в образовании. Интеграция научных и образовательных учреждений позволяет создать систему непрерывного образования для подготовки кадров высшей квалификации.

Для реализации этих задач необходимо использовать современные методики доставки знаний. К ним относится электронное обучение. Сегодня через использование доступных технологических решений, Интернета, мобильных телефонов, а также новых программных средств становится возможным доносить знания до географически отдалённых субъектов, делать это быстро, наглядно и эффективно.

Электронное обучение (E-learning) – это обучение, при котором используются современные информационные и коммуникационные технологии для эффективной доставки знаний и развития человеческих ресурсов.

Методы электронного обучения используются сотрудниками кафедры информационных технологий ТГПУ имени Л.Н. Толстого в преподавании на курсах повышения квалификации: «Технологии электронного обучения», «Электронное обучение в учебном процессе», «Проектирование электронного курса с использованием системы дистанционного обучения MOODLE», «Электронное обучение как эффективное средство доставки знаний» и дополнительной профессиональной образовательной программы для получения квалификации «Преподаватель высшей школы». Слушателями курсов повышения квалификации были педагогические работники учебных заведений, поэтому обучение имело ряд принципиальных особенностей:

- немедленное применение полученных знаний на практике;
- ориентация в обучении на решение проблемы;
- планирование учебного процесса осуществляется совместно с обучающимся;
- определение потребностей обучения совместно с обучающимся;
- формулирование целей обучения совместно с обучающимся;
- учебная деятельность взрослого человека представляет собой технологию поиска новых знаний на основе опыта;
- совместное определение новых учебных потребностей, совместная оценка программ обучения.

Для мобильности представления учебного материала слушателям была предоставлена возможность ознакомиться с лекциями и выполнить ряд заданий дистанционно в любое, удобное для них, время. Для этого преподавателями был разработан ди-

станционный курс «Электронное обучение в учебном процессе» [1], расположенный в среде управления обучением MOODLE. Выход на курс осуществляется через сеть Интернет и сайт Тульского государственного педагогического университета им. Л.Н. Толстого. Данный курс является доступным любому пользователю и отвечает идеологии свободного программного обеспечения.

В ходе изучения данного курса будущие преподаватели знакомятся с тенденциями развития современных информационных и коммуникационных технологий, с устройствами систем мультимедиа, информационными технологиями, ориентированными на использование вычислительной сети, а также применением современного программного обеспечения в научной и образовательной деятельности. Целью данного курса является развитие навыков владения современными информационными технологиями, поиска информации в глобальной сети Интернет, владение навыками работы с электронным учебным материалом, создание собственных сетевых ресурсов.

Одним из достоинств электронных средств обучения является индивидуальный темп обучения, вариативность развёрнутости учебного материала, учёт типа памяти, темперамента и мышления обучаемых, основанных как на психолого-педагогических требованиях, так и на психофизиологических.

Курс построен на основе модульного обучения, сущность которого состоит в том, что содержание обучения структурируется в автономные организационно-методические блоки – модули, содержание и объём которых могут варьировать в зависимости от дидактических целей, профильной и уровневой дифференциации обучающихся, желаний обучающихся по выбору индивидуальной траектории движения по учебному курсу.

В зарубежной и отечественной литературе основное внимание сосредоточено на разработке дидактических элементов модульного обучения, направленных на выработку практико-ориентированных навыков.

Принципы педагогического проектирования (Гурье Л.И.) [2]:

Принцип человеческих приоритетов, ориентирующий на человека – участника подсистем, процессов или ситуаций. Этот приоритет означает гуманистический и природосообразный характер обучения.

Принцип саморазвития проектируемых систем, процессов, ситуаций предполагает создание их динамичными, гибкими,

способными по ходу реализации к изменениям, перестройке, усложнению или упрощению.

Принцип динамизма — проектирование должно вестись «сверху-вниз»: сначала продумывается общая идея, а затем осуществляется разработка отдельных материалов.

Принцип полноты, предполагающий обеспечение реализации в спроектированном объекте системы требований к его функционированию.

Принцип диагностируемости – организация постоянной обратной связи, реализация измерительного инструментария, мониторинг функционирования системы на практике.

Принцип конструктивной целостности – установление прочной взаимосвязи между компонентами методической системы и этапами ее проектирования и реализации на практике.

В настоящее время учебные заведения ориентируются на гуманистический подход в образовании, главной отличительной чертой которого является особое внимание к индивидуальности человека, его личности и сознательное развитие им самостоятельного критического мышления.

Требования, предъявляемые к содержанию учебных материалов:

- соответствие целям обучения;
- практическая направленность;
- интерактивность;
- наличие практикума;
- способность вызвать заинтересованность обучающихся;
- способность активизировать самостоятельную работу и творческую активность обучающихся, обеспечивать выработку определённых умений путём тренировок до уровня осознанного умения.

Формирование профессионально значимых качеств обучающихся в модульном обучении может осуществляться более эффективно благодаря специфическим психодидактическим условиям, которые способствуют следующему [3]:

- развитию профессионально-личностной направленности;
- формированию устойчивой мотивации профессиональных достижений;
- ориентации на самостоятельность в профессионально-личностном становлении;

- монотипности групповой направленности и форм совместной деятельности;
- формированию адекватной самооценки и высоких показателей профессиональной самоактуализации;
- регламентируемой и контролируемой индивидуализации обучения;
- полимодальности в восприятии материала;
- приобретению навыков деятельностного самоконтроля;
- психологической оптимизации профессиональной подготовки, базирующейся на индивидуальном плане обучения.

Модульное обучение предполагает жёсткое структурирование учебной информации, содержания обучения и организацию работы учащихся с полными, логически завершёнными учебными блоками (модулями). Модуль совпадает с темой учебного предмета. Однако, в отличие от темы в модуле, все измеряется, все оценивается: задание, работа, посещение занятий, стартовый, промежуточный и итоговый уровень учащихся. В модуле чётко определены цели обучения, задачи и уровни изучения данного модуля, названы навыки и умения. В модульном обучении все заранее запрограммировано: последовательность изучения учебного материала, уровень и контроль качества усвоения.

Принципы модульного обучения достаточно легко могут быть реализованы при проектировании электронного курса в LMS MOODLE. Размещённый теоретический материал является мобильным – оперативно вносятся изменения и дополнения в курс. В курс можно вставить различные информационные ресурсы от электронных версий текстологических источников до мультимедийных разработок педагогических ресурсов курса.

Элемент курса «Урок» позволяет организовать пошаговое изучение учебного материала. Массив материала можно разбить на дидактические единицы, в конце каждой из них дать контрольные вопросы на усвоение материала. Система, настроенная преподавателем, позаботится о том, чтобы, по результатам контроля, перевести студента на следующий уровень изучения материала или вернуть к предыдущему. Урок позволяет проводить оценивание работы студентов в автоматическом режиме: преподаватель задаёт системе параметры оценивания, система выводит для каждого студента общую оценку за урок, заносит её в ведомость.

Форум является важной частью курса, где студенты могут получить ответы на интересующие их вопросы как от однокурс-

ников, так и от преподавателя. Форумы MOODLE можно структурировать различным образом, можно разрешить добавлять к сообщениям файлы, разрешать студентам оценивать сообщения. Один из существенных форумов, который автоматически создаётся в каждом курсе, — это «Новостной форум». Это хорошее средство для сообщения студентам различной информации. Форум можно настроить таким образом, что при добавлении нового сообщения, оно автоматически отправляется всем студентам по электронной почте. Форум также хорошее средство для организации и проведения групповых работ, особенно когда речь идёт о дистанционном обучении и студенты не могут друг с другом встретиться. В этом случае форум – это место, где они могут общаться, и в то же время преподаватель может контролировать деятельность в группе.

Задания для выполнения электронного курса должны быть построены так, чтобы выполнение заданий носило творческий характер и невозможно было найти готовый ответ ни в книге, ни в Интернете.

Функция преподавателя изменяется: вместо источника знаний он превращается в «центр влияния». Преподаватель должен найти индивидуальный контакт с каждым учеником, адаптируясь под его образовательные потребности, а также направлять дискуссии и совместную деятельность таким образом, чтобы коллективно достичь целей обучения.

Описанные выше принципы модульного обучения реализованы в дистанционном курсе «Электронное обучение в учебном процессе». Так, пошаговое изучение учебного материала происходит в ходе освоения следующих тематических разделов:

- Информатизация общества и образования.
- Аппаратные средства и информационные технологии современного информационного общества.
- Структура и компоненты электронного учебника.
- Современные способы представления информации.
- Методы разработки тестовых материалов учебного назначения.
- Метод проектов.
- Образовательные электронные ресурсы.
- Проектирование и использование в образовательном процессе электронных информационных ресурсов.
- Обучение выполнению учебных проектов в графических и текстовых редакторах.

- Статистическая обработка и представление результатов педагогического исследования.
- Проектирование и разработка информационных ресурсов учебного назначения с использованием развивающей среды ЛогоМиры 3.0.
- Проектирование и разработка информационных ресурсов учебного назначения с использованием программы презентационной графики MS PowerPoint.
- Основы Web-дизайна.
- Технология рисования и создания анимированных объектов средствами Macromedia Flash MX.

Непрерывное взаимодействие преподавателей и студентов в рамках данного курса организовано в форуме «Проблемы электронного обучения». Участие в работе форума позволяет слушателям курсов получить ответы на интересующие их вопросы по данной проблематике, высказать своё собственное мнение. Также в ходе осуществления подобной деятельности у обучающихся формируется понимание возможности различных позиций и точек зрения, умение в разработке стратегий установления положительных отношений с коллегами посредством обмена опытом в рамках межсетевого педагогического взаимодействия, развивается творческая активность обучающихся и т.д.

Особенный интерес представляют предлагаемые в ходе изучения дистанционного курса практические задания. Выполнение задания «Создайте графические образы представления знаний из своей предметной области, используя любые два метода: ментальные карты, кластеры, денотатные графы, схема «рыбий скелет», концептуальные таблицы» для темы «Современные способы представления информации» формирует умения анализировать, синтезировать, структурировать информацию, создавать различные модели по преподаваемым предметам. В ходе выполнения практического задания «Создайте презентацию «психологическое тестирование» по материалам лабораторной работы, представленной в книге Составление интерактивных продуктов раздел Психологическое тестирование» для темы «Проектирование и разработка информационных ресурсов учебного назначения с использованием программы презентационной графики MS PowerPoint» происходит формирование умений по решению профессиональных задач в исследовании и оценке объекта (субъекта) педагогической деятельности, результатов педагогического процесса с применением современных средств

оценивания, в том числе с использованием компьютерных тестов.

Подобные задания разработаны для каждого тематического раздела и способствуют развитию творческой активности обучающихся. В качестве заданий для некоторых тем предлагаются проблемные вопросы, ответ на которые представляет собой решение информационной задачи по поиску, оценке, анализу, обработке полученной информации, и результатом такого решения становится «эксклюзивный продукт» мыслительной деятельности, реализованный средствами ИКТ. В качестве примера такого задания для дистанционного курса «Электронное обучение в учебном процессе» можно представить следующее:

«Составьте эссе в виде текста со следующим содержанием:

1. Тема электронного учебника.
2. Назначение и цели создания учебника.
3. Контингент слушателей, которые будут работать с Вашим учебником.
4. Структура электронного учебника по выбранной теме или разделу преподаваемого курса.
5. Последовательность уроков (см. Особенности проектирования содержания электронного учебного пособия).
6. Список контрольных вопросов.
7. Список основной и дополнительной литературы.
8. Помощь по работе с электронным учебником.
9. Дополнительный материал по работе с электронным учебником.
10. Строка копирайта».

За 2 года существования курса было обучено более 50 человек, среди которых преподаватели средних специальных учебных заведений (ТПК №1, г. Тула), учителя (МОУ Плавская СОШ №2, г. Плавск Тульской области), слушатели дополнительной профессиональной образовательной программы для получения квалификации «Преподаватель высшей школы» в рамках изучения дисциплины «Информационные технологии в науке и образовании».

В заключение отметим, что в учебном процессе образовательных учреждений использование электронных систем управления обучением не только позволяет повысить эффективность обучения, но и знакомит слушателей с современными педагогическими технологиями.

Используемые источники:

1. Богатырева Ю.И., Пихтилькова О.А., Родионова О.В., Ситникова Л.Д., Яфаева Р.Р. «Электронное обучение в учебном процессе»
2. www.tspu.tula.ru/moodle/course/view.php?id=325
3. Гурье Л.И. Проектирование педагогических систем: учебное пособие. –Казань: КГТУ, 2004.
4. Дистанционные образовательные технологии: проектирование и реализация учебных курсов/ Лебедева М.Б., Агапонов С.В., Горюнова М.А., Костиков А.Н., Костикова Н.А., Никитина Л.Н., Соколова И.И., Степаненко Е.Б., Фрадкин В.Е., Шилова О.Н./ Под общ. ред. М.Б. Лебедевой. – СПб.: БХВ-Петербург, 2010. – 336 с.

УДК 004.9+004.4

ВОЗМОЖНОСТИ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧЕНИЯ СТУДЕНТОВ

В.В. Кулямин, А.В. Хорошилов

Институт системного программирования РАН

О.Л. Петренко

Московский институт открытого образования
г. Москва, Россия

E-mail: kuliamin@ispras.ru, o-l-petrenko@yandex.ru,

khoroshilov@ispras.ru

Аннотация

В работе рассматриваются возможности, которые предоставляют проекты по разработке свободного программного обеспечения для организации учебных курсов в высших учебных заведениях, а также представлена рамочная структура практических курсов для обучения студентов по направлению информационных технологий, которая нацелена на реализацию этих возможностей.

Ключевые слова

Свободное программное обеспечение, обучение информационным технологиям.

Достоинства использования проектов по разработке свободного программного обеспечения (СПО) для построения курсов в

высших учебных заведениях и, в особенности, при обучении студентов информационным технологиям активно обсуждаются [1-4]. На первом месте традиционно указывается возможность изучения внутреннего устройства СПО, появляющаяся ввиду доступности его исходного кода. Но это далеко не единственное достоинство СПО. Проекты по разработке СПО часто являются открытыми, то есть придерживаются открытого процесса разработки, в ходе которого все обсуждения по принятию проектных решений, развитию архитектуры ПО и разработке проектной документации проводятся публично. Все это в комплексе создаёт прекрасный материал и готовую инфраструктуру для обучения студентов программной инженерии и информационным технологиям в целом.

Использование открытых проектов по разработке ПО может служить основой для построения открытой образовательной среды в области программной инженерии. В частности, это позволяет разработчикам учебного курса получить практически для любой темы в данной дисциплине следующие инструменты:

- готовый учебный материал: требования к ПО, спецификации его архитектуры, исходный код, работающее ПО, уже решённые и все ещё актуальные проблемы и дефекты, возникшие в ходе его разработки, обсуждения разработчиками различных проектных решений и проблем, часто с развёрнутой аргументацией;

- готовую инфраструктуру для ведения практических занятий: реальные проекты по разработке ПО, описание их проблем и дефектов, которые можно устранять в рамках заданий, доступное общение с реальными командами разработчиков;

- возможность создания сильной мотивации для студентов в их обучении и практической деятельности, связанной с их участием в реальных проектах, имеющих общественную значимость.

Конечно, в качестве учебных материалов можно просто взять документы и код одного или нескольких крупных проектов. Но, помимо этого, существование большого разнообразия проектов по разработке СПО позволяет обеспечить беспрецедентную свободу выбора: студент может самостоятельно, на основе своих предпочтений и собственных представлений о карьере, выбирать предметную область проекта, используемые технологии и инструменты разработки. Такая свобода является осо-

бенно важной в контексте принципов открытого образования [5], которое требует предоставления предельной самостоятельности в обучении, в частности, в выборе собственной образовательной траектории.

Материалы открытых проектов служат отличной доступной базой знаний в различных разделах информационных технологий, своего рода аналогом классической публичной библиотеки, благодаря которой можно повысить активность и самостоятельность студентов в учебном процессе, а также получить больше возможности для индивидуализации образовательного процесса. Кроме того, предоставляя студентам свободу выбора предметной области, языков программирования и применяемых технологий в рамках изучения отдельной темы, можно повысить личную заинтересованность и мотивированность, а также уровень ответственности за самостоятельно сделанный выбор.

Институт системного программирования РАН (ИСП РАН) является базовой организацией для кафедры системного программирования ВМиК МГУ им. М.В. Ломоносова и одноимённой кафедры МФТИ. В ИСП РАН разработана рамочная структура практических курсов для обучения студентов по направлению ИТ, основанная на принципах открытого образования и использующая возможности, предоставляемые открытыми проектами по разработке СПО [6].

В рамках обучения к каждому студенту прикрепляется личный куратор, т.е. преподаватель, основные обязанности которого — поддерживать постоянный контакт со студентом, отвечать на его вопросы самому или указывать необходимые источники информации. К одному куратору прикрепляется не более 4-х студентов. Параллельно с практическим курсом должен проводиться теоретический курс по соответствующей теме, при этом важно, чтобы промежуток времени между лекцией по какой-то теме и использованием материала этой лекции на практике был небольшим. Возможны и опережающие практические задания, в ходе которых студентам нужно использовать материал лекций, которые будут прочитаны в промежутке между выдачей им заданий и контролем их выполнения.

Основные элементы предлагаемой структуры практикума следующие.

- Выбор открытого проекта по разработке ПО. Студенту предлагается самостоятельно выбрать проект для своей работы, удовлетворяющий рекомендациям разработчиков курса. Куратор

помогает студенту в этом выборе, ничего не навязывая, и проверяет соответствие выбираемого проекта предъявляемым требованиям.

- Знакомство с проектом. Студент детально знакомится с выбранным проектом, обращая особое внимание на аспекты, связанные с изучаемым курсом. По завершении этого этапа он проводит презентацию своего проекта преподавателям и другим участникам курса, в ходе которой могут задаваться любые вопросы. Преподаватели особое внимание обращают на умение студента сопоставлять теоретический материал курса с увиденным им на практике, дополнительно проверяя это умение с помощью вопросов. В дополнение к презентации от студента можно потребовать подготовить какой-либо документ о проекте с заранее определённой структурой.

- Практические задачи. В рамках курса готовится набор задач, покрывающих все аспекты изучаемого предмета. Каждая задача состоит в выполнении практического задания с определённой формой отчётности и имеет некоторый весовой коэффициент. Студенту предоставляется возможность самостоятельного выбора задач но так, чтобы сумма их весов была выше обязательного порога. Задания могут зависеть друг от друга, и для выполнения одного может потребоваться сначала выполнить другое.

При подготовке практических заданий важно учитывать дополнительные возможности, предоставляемые открытыми проектами и существенно помогающие процессу обучения.

Для оценки результативности работы студента нужно использовать реакцию команды проекта и сообщества, существующего вокруг него. Это позволяет значительно повысить мотивацию студента к обучению и учит его оценивать свои результаты более объективно.

Необходимо развивать социальные навыки студентов, стимулируя их к активному общению с другими членами проекта, а также к обсуждению деталей проекта с преподавателями и другими студентами. Такое общение постепенно учит студентов эффективно общаться на профессиональные темы и углубляет их понимание предмета.

Свобода выбора заданий в ходе обучения также способствует повышению мотивации и более сознательному отношению студента к своему обучению [2]. Она обеспечивает индивидуаль-

ный выбор глубины и методов погружения в материал курса и предоставляет студенту возможность самостоятельно формировать свои знания и навыки, исходя из собственных предпочтений и потребностей личностного развития.

На основе предложенных принципов нами разработан курс практических занятий по дисциплине «Технологии программирования», который проходит апробацию на кафедре системного программирования ВМиК МГУ и в рамках научно-исследовательской работы на кафедре системного программирования МФТИ. Параллельно первым двум этапам практикума читается теоретический курс «Технологии программирования. Компонентный подход» [7], кроме того, практические занятия предваряет дополнительная лекция, посвящённая принципам открытого программного обеспечения.

Первоначально мы не ставили никаких ограничений при выборе открытого проекта. Единственным обязательным требованием является открытость не только исходного кода, но и всего процесса разработки, проектных документов и обсуждений. Получив первый опыт работы студентов с материалами открытых проектов, мы осознали, что для обучения программной инженерии нужно использовать достаточно активные (хотя бы несколько участников и несколько изменений в исходном коде в месяц), зрелые (результат проекта уже можно запустить и «пощупать») и достаточно большие проекты (более 25000-30000 строк кода на любом из языков высокого уровня). Это необходимо, чтобы в проекте проявлялись проблемы, специфичные для разработки сложных программных систем, а также для возможности изучать принимаемые в проекте решения в их развитии.

На выбор и знакомство с проектом отводится два месяца, по завершении которых студенты проводят презентации, представляя выбранный проект и отвечая на вопросы из заранее известного им списка. Ключевая идея в организации списка вопросов заключается в том, чтобы вынудить студентов найти в выбранном проекте отражение идей и подходов, рассматриваемых в теоретическом курсе. В качестве дополнительного задания при знакомстве с проектом перед студентами ставится задача подготовить один из проектных документов по заранее заданному шаблону (описание концепции проекта, техническое задание к некоторой подсистеме, описание архитектуры, план тестирования и пр.).

Практические задачи включают в себя около 30-ти зада-

ний, покрывающих различные области программной инженерии:

- моделирование предметной области;
- работа с требованиями;
- архитектура ПО;
- обеспечение и контроль качества ПО;
- отладка и исправление ошибок;
- планирование и внесение изменений;
- разработка пользовательской документации;
- поддержка пользователей;
- комплексные задачи.

Первый опыт работы привёл к следующим выводам: студенты на третьем-четвёртом курсе обучения плохо знакомы с принципами свободного программного обеспечения, и это направление не является популярным. При самостоятельном выборе проектов студенты склоняются к проектам, находящимся на начальной стадии, так как считают, что в такие проекты им будет проще вписаться. Возможность выбора предметной области проекта, действительно, существенно повышает заинтересованность студентов в выполнении последующих заданий. Знакомство с wybranymi проектами по разработке СПО показывает, что эти проекты могут быть использованы как для демонстрации грамотных подходов к разработке ПО, так и для демонстрации часто встречающихся ошибок, решений и организационных элементов, которых необходимо избегать в проектах по разработке ПО.

Используемые источники:

1. Allen, E.; Cartwright, R.; Reis, C. Production programming in the classroom. Proc. of the 34-th SIGCSE technical symposium on Computer science education, Reno, Nevada, USA, pp. 89-93, 2003.
2. D. Carrington, S.-K. Kim. Teaching software design with open source software. Proc. of 33-rd ASEE/IEEE Frontiers in Education Conf., pp. 9-14, November 2003.
3. C. P. Fuhrman. Appreciation of software design concerns via open-source tools and projects. Proc. of 10-th Workshop on Pedagogies and Tools for the Teaching and Learning of Object Oriented Concepts, at ECOOP 2006, Nantes, France, July 2006.
4. Meiszner, A, Glott, R. & Sowe, S. K. (2008c), «Preparing the Ne(x)t Generation: Lessons learnt from Free / Libre Open Source Software». GUNI – Global University Network for Innovation,

Newsletter issue September 09.

5. Wedemeyer. Characteristics of open learning systems. In Open Learning Systems, Washington, National Association of Educational Broadcasters, 1974.

6. Петренко А.К., Кулямин В. В., Петренко О.Л., Рубанов В.В., Хорошилов А.В. Подход к преподаванию ИТ на основе проектов по разработке свободного программного обеспечения / Сборник докладов 6-ой всероссийской конференции «Преподавание информационных технологий в России», Нижний Новгород, 12-13 мая 2008 г.

7. Кулямин В.В. Технологии программирования. Компонентный подход. М., ИНТУИТ-Бином, 2007.

УДК 004.9+004.4

ФОРМИРОВАНИЕ ИНФОРМАЦИОННО-ТЕХНОЛОГИЧЕСКОЙ КУЛЬТУРЫ БУДУЩИХ СПЕЦИАЛИСТОВ С ИСПОЛЬЗОВАНИЕМ ВЕКТОРНОЙ ГРАФИКИ

В.А. Дякина

ГОУ ВПО «Елецкий государственный университет
им. И.А. Бунина»
г. Елец, Россия

Аннотация

Доказывается необходимость формирования информационно-технологической культуры будущих специалистов и рассматриваются пути решения этой проблемы с помощью интеграции компьютерной графики в классическом университете.

Ключевые слова

Компьютерная графика, информационно-технологическая культура, свободное программное обеспечение.

Векторная графика в настоящее время является частью повседневной жизни, основным выразительным средством коммуникативных и электронных технологий. Под её влиянием во многом сегодня формируются вкусы и эстетические предпочте-

ния студентов.

К сожалению, уровень современной электронной визуализации далёк от идеального. Как показал опыт работы в классическом университете, в частности, на физико-математическом факультете, даже очень способные студенты при выполнении заданий не могут самостоятельно проявить собственный вкус, стиль, а чаще всего используют готовые шаблоны, образцы. Поэтому формирование информационно-технологической культуры будущих специалистов – проблема, требующая теоретического осмысления и практического решения. Одним из способов её решения, по нашему мнению, является детальное знакомство с возможностями векторной графики.

Изучение векторной графики в университете студенты специальности 010200 «Прикладная математика и информатика» начинают со знакомства с OpenOffice.org Draw. Это осуществляется в рамках дисциплины «Работа в Open Office» в 3 семестре. Практический опыт показал, что из всего многообразия компьютерных технологий наиболее подходящей для первоначального изучения является именно векторная графика, которая проста для изучения.

Целью курса векторной графики является формирование информационно-технологической культуры студентов, достаточной для реализации творческого потенциала личности и востребованной в будущей жизнедеятельности. Под информационно-технологической культурой будем понимать такую характеристику будущего специалиста, которая включает когнитивную, технологическую, личностную, коммуникативную и эмоционально-ценностную компоненты.

Задачами курса являются:

- формирование знаний о фундаментальных понятиях векторной графики (пиксель, воксель, растр, тексель, графический примитив, дублирование, морфинг) и их взаимосвязь;
- выработка опорных знаний и умений в области построения блок-схем, структурных и функциональных схем процессов, схемы сетей, создания листа формата А1с рамкой;
- формирование знаний о принципах, правилах и приемах создания объектов и работы с ними;
- развитие навыков работы студентов в новой технологической среде (рисование графических примитивов с помощью панели «Рисунок»);

- развитие эстетического опыта студентов, включающего восприятие и критическое оценивание созданных объектов;
- привлечение в активную творческую деятельность с помощью новых выразительных средств.

При изучении этого курса использовались пособия А. Ерёмко, А.С. Казанцева, Ю.П. Немчинова [1], [2], [3].

Изучаемый материал был систематизирован для логичного усвоения, описаны рекомендуемые действия для решения профессиональных задач. Каждая тема включала знаниевую и творческую составляющие и опиралась на уже известные студентам из школьного курса средства и правила. Задания подобраны были с постепенно возрастающей сложностью и сопровождалась пошаговой инструкцией, которая технически обеспечивала выполнение законченной работы. По мере выполнения предложенных заданий требования постепенно смещались в область индивидуального творческого решения. Всё это, на наш взгляд, способствует развитию индивидуального вкуса и почерка, как следствие, повышается уровень информационно-технологической культуры будущих специалистов.

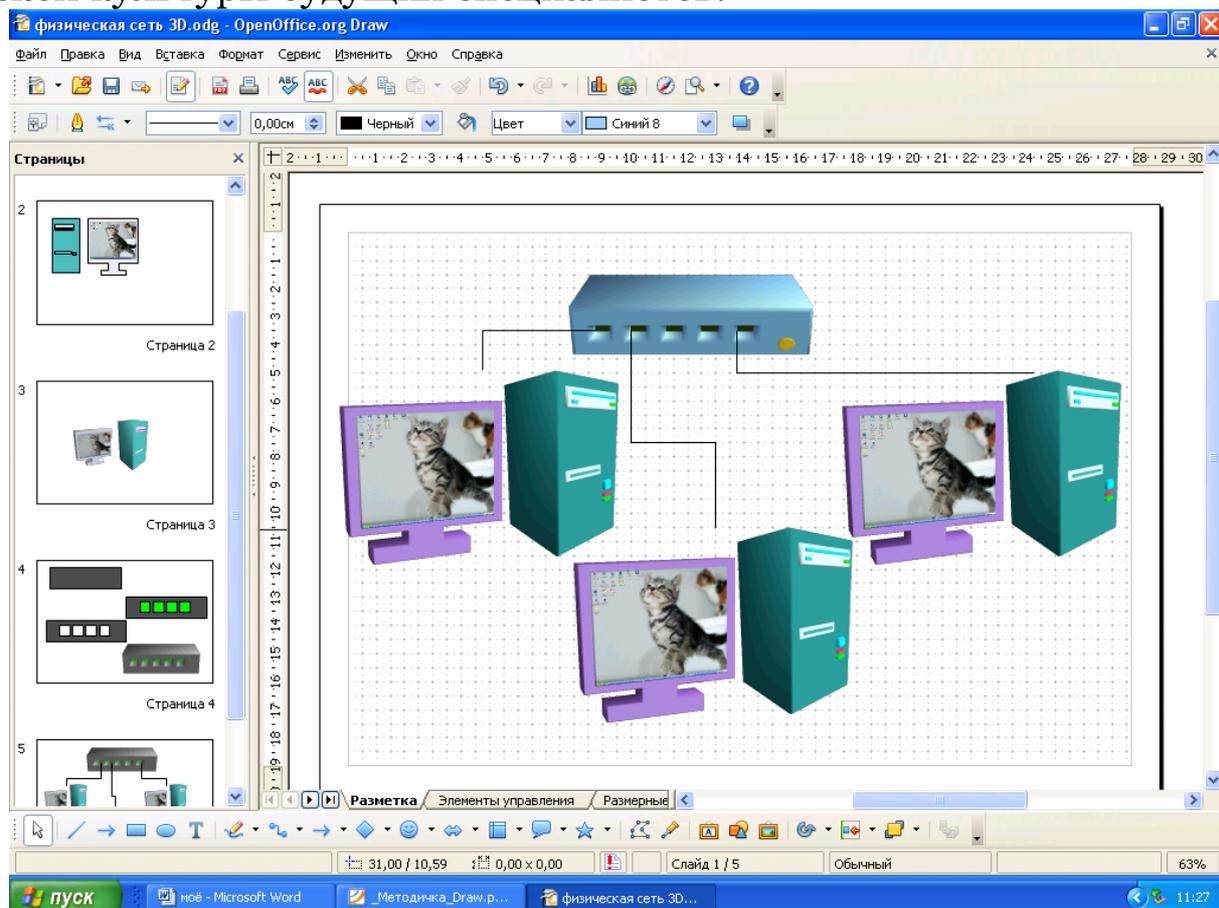


Рисунок 1. Схема сети Fast Ethernet с тремя машинами на базе пяти порто-

вого коммутатора на витой паре.

Рассмотрим реализацию практического задания, предложенного в методическом пособии А.С. Казанцева [2].

Задание: построить схему сети Fast Ethernet с пятью машинами на базе пятипортового коммутатора на витой паре.

Согласно пошаговой инструкции, студенты строят сначала физическую сеть 2D, а затем физическую сеть 3D. В результате получаем итоговую схему. Пошаговая реализация этого алгоритма отражена на рис. 1.

Используемые источники:

1. Ерёменко А. Руководство по Draw [Электронный ресурс] - (<http://authors.i-rs.ru/04%20Draw%20Guide/%d0%f3%ea%ee%e2%ee%e4%f1%f2%e2%ee%20%ef%ee%20Draw.pdf>).
2. Казанцев А.С. Рисование схем в OpenOffice.org Draw [Электронный ресурс] / А.С. Казанцев – (<http://www.i-rs.ru/content/view/full/1108>).
3. Немчанинова Ю.П. Создание и редактирование графических элементов и блок-схем в среде OpenOffice.org: учебное пособие. - М., 2008.

УДК 004.9+004.4

ПРАКТИКА ИСПОЛЬЗОВАНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ПРИ ПОДГОТОВКЕ БУДУЩИХ ЭКОНОМИСТОВ

С.В. Воробьёв

ГОУ ВПО «Елецкий государственный университет им. И.А. Бунина»

г. Елец, Россия

E-mail: vorobsu@mail.ru

Аннотация

Использование свободного программного обеспечения в процессе профессиональной подготовки будущих специалистов экономического профиля позволяет повысить уровень их информационно-технологической культуры. В данной статье рассмотрен один из примеров решения конкретной экономической задачи в электронной таблице OpenOffice.org Calc.

Ключевые слова

Электронная таблица, свободное программное обеспечение, специалист экономического профиля, информационно-технологическая культура.

В настоящее время широкое распространение во всех сферах и направлениях деятельности человека получили программные продукты, относящиеся к свободному программному обеспечению. Такие программные продукты отличаются от привычных коммерческих разработок лишь отсутствием платы за их использование и свободным доступом к программному коду. Сегодня существуют различные типы свободного программного обеспечения, начиная от операционных систем и заканчивая обычным текстовым редактором. Несмотря на доступность данных программ, они обладают большими функциональными возможностями, зависящими от прикладного назначения. Перечисленные достоинства свободного программного обеспечения позволяют стремительно внедряться во все отрасли народного хозяйства, в том числе в экономику и образование.

Профессиональная подготовка современного конкурентоспособного специалиста экономического профиля требует высокого уровня владения новыми информационными технологиями и программным обеспечением, которые реально используются в экономическом секторе любой организации. Поэтому следует большое внимание уделять совершенствованию информационно-технологической подготовки будущих экономистов в общей структуре их профессиональной подготовки. Одним из путей совершенствования может выступать тщательное изучение технологий работы с программными продуктами, относящимися к свободному программному обеспечению.

Профессиональная деятельность любого специалиста экономического профиля не может обойтись без применения так называемой электронной таблицы. С помощью данной программы можно строить различного рода таблицы, проводить сложные математические вычисления экономических операций над данными, строить всевозможные наглядно-графические образы толкования экономической информации, осуществлять анализ данных с использованием встроенных функций. Электронная таблица имеет место и в свободном программном обеспечении, например, в составе пакета OpenOffice.org, где она называется

OpenOffice.org Calc. Покажем технологию применения программного продукта OpenOffice.org Calc в процессе информационно-технологической подготовки будущих специалистов экономического профиля, а именно, рассмотрим решение конкретной экономической задачи.

Предположим, что требуется создать электронную таблицу для учёта заказов в оптовом центре «Роспродторг» с помощью пакета OpenOffice.org. В условии задачи также прописаны поля основной таблицы и таблиц-справочников. А именно, основная таблица Учёт заказов будет содержать следующие поля: *Дата, Номер заказа, Код товара, Наименование товара, Количество, Отпускная цена за единицу, Код клиента, Название фирмы, Сумма заказа, Сумма заказа клиента нарастающим итогом, Скидка, Уплачено*. Первая вспомогательная таблица *Справочник клиентов* должна содержать информацию о постоянных клиентах оптового центра и их основных данных. Справочник клиентов должен иметь следующие поля: *Код клиента, Название фирмы, Контактная персона, Электронная почта*. Вторая вспомогательная таблица *Справочник товаров* будет содержать информацию об ассортименте торгового центра. Справочник товаров должен иметь следующие поля: *Код товара, Наименование товара, Единица измерения, Отпускная цена*. При этом по условию задачи клиентам делается следующая скидка: в размере 5%, если сумма заказа клиента нарастающим итогом больше 30000 р., и 3%, если сумма заказа клиента нарастающим итогом больше 10000 р. При решении задачи следует учесть, что все поля в основной таблице *Учёт заказов* заполняются при помощи встроенных в электронную таблицу функций, кроме следующих полей: *Дата, Код товара, Количество, Код клиента*. Данные в справочниках заполняются вручную.

Рассмотрим один из вариантов решения поставленной задачи в электронной таблице OpenOffice.org Calc. В первую очередь следует создать формы таблиц справочников и заполнить их соответствующими данными. Эти таблицы лучше располагать на разных рабочих листах, которые можно назвать *Справочник клиентов* и *Справочник товаров*. В поле *Отпускная цена* значения должны иметь денежный формат, который устанавливается через меню *Формат* пункт *Ячейки* и на вкладке *Числа* выбирается категория *Денежный*. Фрагмент окна листа *Справочник товаров* с заполненной таблицей показан на рис. 1.

Затем следует сформировать основную таблицу на новом рабочем листе, который назовём *Учёт заказов*. После создания заголовка полей следует приступить к заполнению таблицы данными. Значения в поле *Дата* по условию задачи вносятся в таблицу вручную. При этом важно учесть, что формат ячеек данного поля будет иметь категорию *Дата* с кодом формата DD.MM.YY. Заполним произвольным образом поле *Дата*.

Далее необходимо автоматизировать процесс заполнения данными в поле *Номер заказа*. Сущность автоматизации этого поля состоит в том, чтобы при вводе нового значения в поле *Дата* автоматически заполнялось соответствующее значение в поле *Номер заказа*, причём это значение должно быть следующим по порядку и иметь формат 00-01, 00-02, 00-03 и т.д. Для этого следует проделать следующие шаги.

Задача.ods - OpenOffice.org Calc				
Файл Правка Вид Вставка Формат Сервис Данные Окно Справка				
Times New Roman 12 Ж К Ч				
D4 = 20				
	A	B	C	D
1	Справочник товаров оптового центра "Роспродтор"			
2				
3	Код товара	Наименование товара	Единица измерения	Опускная цена
4	10000	Молоко 1л.	Упаковка	20,00 руб.
5	10001	Молоко топленое 0,5л.	Упаковка	17,50 руб.
6	10002	Сметана 0,5л.	Упаковка	25,76 руб.
7	10003	Кефир 1л.	Упаковка	14,20 руб.
8	10004	Ряженка 0,5л.	Упаковка	11,90 руб.
9	20000	Апельсины	Килограмм	80,00 руб.
10	20001	Мандарины	Килограмм	67,00 руб.
11	20002	Бананы	Килограмм	31,50 руб.
12	20003	Киви	Штука	5,50 руб.

Рис. 1. Фрагмент листа *Справочник товаров*

В ячейку первой строки поля *Номер заказа* заносится числовое значение 1, что характеризует первый номер заказа.

1. В следующую ячейку поля *Номер заказа* добавляется функция IF (ЕСЛИ) через *Мастер функций*. Данная функция указывает логический тест, который необходимо выполнить. Функция IF имеет три поля для ввода: во-первых, Тест, куда вводится любое значение или выражение, которое при вычислении даёт значение TRUE (ИСТИНА) или FALSE (ЛОЖЬ);

во-вторых, *Тогда значение*, куда вводится значение, которое возвращается, если *Тест* имеет значение TRUE; в-третьих, *Иначе значение*, куда вводится значение, которое возвращается, если *Тест* имеет значение FALSE. Логика данной функции в рамках заполнения поля *Номер заказа* имеет следующий смысл: до тех пор, пока не будет введено значение в поле *Дата*, ячейка с соответствующим номером заказа будет оставаться пустой, а как только заполнится ячейка с датой, автоматически в соответствующей ячейке поля *Номер заказа* отобразится следующий по порядку номер. В результате формула в поле *Номер заказа*, начиная со второй строки будет иметь примерно следующий вид: =IF(A5="";"";B4+1). Здесь последнее выражение означает прибавление единицы к предыдущему значению ячейки номера заказа.

2. Перед тем как копировать полученную формулу на много ячеек вниз, необходимо задать формат значений этого поля. Для этого можно воспользоваться созданием нового кода формата ячеек, а именно, "00-"00, который устанавливается через меню *Формат* пункт *Ячейки* и на вкладке *Числа* в поле *Код формата*. Затем можно скопировать формулу, например, на 1000 ячеек вниз, в результате чего все номера будут иметь одинаковый вид, заданный условием задачи.

Следующим полем, заполнение ячеек которого необходимо автоматизировать по условию задачи, является *Наименование товара*. Логика данного поля такова, что в нем должны появляться значения наименования товара в соответствии со значением находящегося левее поля *Код товара*, которое вводится вручную. Зависимость кода товара и наименования товара установлена на листе *Справочник товаров*. Здесь также следует учитывать часть логической структуры, показанной при заполнении поля *Номер заказа*, а именно, значение в поле *Наименование товара* не должно появляться до тех пор, пока не будет заполнена значением соответствующая ячейка поля *Код товара*. Таким образом, вводимая формула должна функционировать по следующему алгоритму: при вводе значения в ячейку поля *Код товара* в соответствующей (пока ещё пустой) ячейке поля *Наименование товара* автоматически появляется значение связанного наименования товара.

Для реализации обозначенного алгоритма воспользуемся комбинацией уже известной функции IF и функции LOOKUP.

Функция LOOKUP вычисляет содержимое ячейки или однострочного или одностолбцового массива. Данная функция имеет три поля для ввода параметров: *Критерий поиска* (значение, с которым следует сравнивать); *Вектор просмотра* (вектор, строка или столбец, в котором следует искать); *Вектор результата* (вектор, строка или диапазон, из которого определяется значение). После применения указанных функций в совместной комбинации окно мастера функций будет иметь вид, показанный на рис. 2.

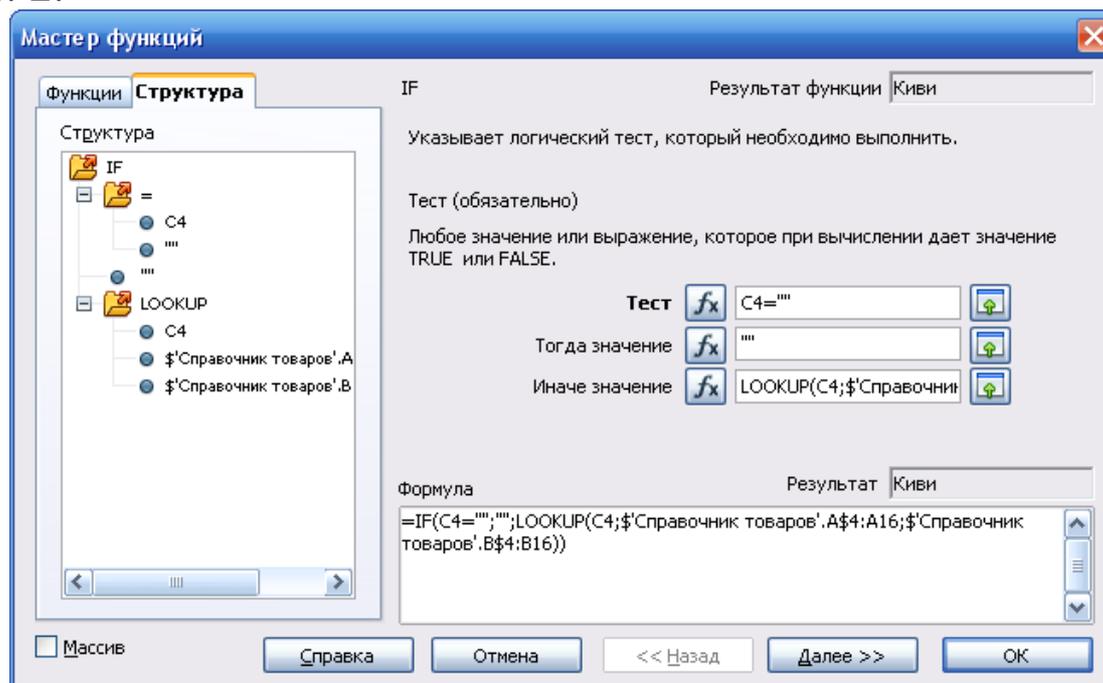


Рис. 2. Окно мастера функций при заполнении поля

При этом параметры функции LOOKUP заполняются следующим образом: критерием для поиска является ячейка с кодом товара на текущем рабочем листе Учёт заказов; Вектор просмотра будет содержать интервал со всеми имеющимися кодами товаров из листа Справочник товаров, причём как минимум верхняя граница этого интервала должна быть зафиксирована абсолютной ссылкой; Вектор результата также берётся из листа Справочник товаров из поля Наименование товара и также начало интервала фиксируется абсолютной ссылкой. Параметры функции IF заполняются аналогично её предыдущему применению, а именно, в поле Тест вводится ячейка с соответствующим кодом товара и приравнивается к пустому множеству; в поле Тогда значение просто вводится пустое множество (обычные кавычки без пробела); в поле Иначе значение вставляется описанная выше функция LOOKUP. Таким образом, значения поля На-

именование товара будут вычисляться автоматизированно, что и требуется по условию поставленной задачи. Аналогичным способом заполняются поля *Отпускная цена за единицу* и *Название фирмы*.

Наименование товара

Поле *Сумма заказа* вычисляется обычным перемножением ячеек полей *Количество* и *Отпускная цена за единицу* с учетом применения функции IF. В этом случае формула будет иметь следующий примерный вид: =IF(G4="";"";E4*F4).

Формула для подсчёта значений в поле *Сумма заказа клиента* нарастающим итогом состоит из комбинации функций IF и SUMIF. Функция SUMIF суммирует аргументы, удовлетворяющие условию. Она состоит из трёх полей заполнения: *Диапазон* (указывает диапазон ячеек, к которому применяются критерии); *Критерий* (область ячейки, в которой заданы критерии поиска); *Диапазон суммирования* (диапазон ячеек, значения которых суммируются). Логика данной функции такова, что она позволяет нам получать сумму заказа клиента нарастающим итогом, т.е. как только один и тот же клиент оформляет покупку более чем один раз, его последняя сумма заказа будет прибавляться к накопленной ранее сумме его заказа. Это позволит нам определить условия для получения клиентом скидки, которые установлены в соответствии с условием задачи. Таким образом, окончательная формула для подсчёта значений в поле *Сумма заказа клиента* нарастающим итогом будет иметь следующий вид: =IF(G4="";"";SUMIF(G\$4:G4;G4;I\$4:I4)).

Теперь перейдём к заполнению поля *Скидка*. В этом поле формула будет состоять из комбинации трёх функций IF. Первая из них должна реализовать процедуру проверки ячейки на пустоту заполнения, что аналогично предыдущим формулам, а вторые две будут участвовать в процессе определения величины скидки в соответствии с условием задачи. Другими словами, если сумма заказа клиента нарастающим итогом больше 30000 р., то ему предоставляется скидка в размере 5%, а если сумма заказа клиента нарастающим итогом больше 10000 р., то скидка будет в размере 3%. В любом другом случае скидка клиенту не даётся. При таких условиях окончательная формула вычисления скидки будет иметь вид:

=IF(G4="";"";IF(J4>=30000;0,05;IF(J4>=10000;0,03;0))).

Для окончательного решения поставленной задачи остаётся автоматизировать поле *Уплачено*, которое вычисляется в зависимости от суммы заказа с учётом полученной клиентом скидки. Здесь также проверяется условие на пустоту ввода последней ячейки, введённой вручную. В результате формула для подсчёта *Уплачено* будет следующей: =IF(G4="";"";I4*(1-K4)).

Таким образом, поставленная задача по автоматизации заказов в оптовом центре «Роспродторг» была успешно решена средствами электронной таблицы OpenOffice.org Calc. На этом примере мы показали лишь малую часть функционала электронной таблицы OpenOffice.org Calc. Данный программный продукт содержит большое количество встроенных функций, что позволяет легко автоматизировать любой сложности финансовые, математические и другие виды расчётов. Изучение электронной таблицы OpenOffice.org Calc и её применение на практике в процессе обучения способствует повышению уровня информационно-технологической подготовки будущих экономистов.

СВЕДЕНИЯ ОБ АВТОРАХ

- Аверьянов Герман Петрович* — доцент, Федеральное государственное бюджетное учреждение высшего профессионального образования «Национальный исследовательский ядерный университет» «МИФИ», Москва
- Андропова Елена Васильевна* — кандидат педагогических наук, доцент, руководитель Центра СПО, ГОУ ВПО «Елецкий государственный университет им. И.А. Бунина», г. Елец
- Баранов Алексей Владимирович* — аспирант, Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИ РАН), г. Санкт-Петербург
- Барышева Александра Станиславовна* — преподаватель математики, ГОУ НПО «Профессиональное училище № 62 Краснодарского края», с. Ванновское
- Бессонов Сергей Геннадьевич* — кандидат технических наук, директор ООО «ЭЛСИС», г. Нижний Новгород
- Будкин Валерий Андреевич* — старший преподаватель, Федеральное государственное бюджетное учреждение высшего профессионального образования «Национальный исследовательский ядерный университет» «МИФИ», Москва
- Вейсов Евгений Алексеевич* — проректор по информатизации, кандидат технических наук, профессор, ГОУ ВПО «Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнёва», г. Красноярск
- Воробьев Сергей Владимирович* — кандидат педагогических наук, доцент, ГОУ ВПО «Елецкий государственный университет им. И.А. Бунина», г. Елец
- Голубев Роман Александрович* — системный администратор, ООО «Сова-софт», г. Липецк
- Горошкин Антон Николаевич* — кандидат технических наук, ст. преподаватель, ГОУ ВПО «Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнёва», г. Красноярск
- Губин Михаил Алексеевич* — ст. преподаватель кафедры автоматизированных систем управления и математического обеспечения, ГОУ ВПО «Елецкий государственный университет им. И.А. Бунина», г. Елец
- Губина Татьяна Николаевна* — ст. преподаватель кафедры вычислительной математики и информатики, ГОУ ВПО «Елец-

кий государственный университет им. И.А. Бунина», г. Елец
Демурин Владимир Борисович — старший преподаватель, АФ
ГОУ ВПО «Московский государственный гуманитарный уни-
верситет им. М.А. Шолохова», г. Анапа
Дербень Александр Михайлович — начальник отдела внедрения
ИТ ЦИТ, ГОУ ВПО «Сибирский аэрокосмический университет
им. ак. М.Ф. Решетнёва», г. Красноярск
Дмитриева Валентина Викторовна — ассистент, Федеральное
государственное бюджетное учреждение высшего профессио-
нального образования «Национальный исследовательский
ядерный университет» «МИФИ», Москва
Дякина Валентина Алексеевна — ассистент кафедры вычисли-
тельной математики и информатики, ГОУ ВПО «Елецкий го-
сударственный университет им. И.А. Бунина», г. Елец
Еркимбаев Адильбек Омирбекович — кандидат технических
наук, начальник информационно-компьютерного Центра, Рос-
сийская академия наук, Объединённый институт высоких
температур, Москва
Зицерман Владимир Юрьевич — кандидат физико-математиче-
ских наук, старший научный сотрудник, заведующий лабора-
торией баз данных по свойствам веществ и материалов, Рос-
сийская академия наук, Объединённый институт высоких
температур, Москва
Ивличев Павел Сергеевич — кандидат физико-математических
наук, старший преподаватель кафедры бухгалтерского учёта
и аудита, Рязанский филиал ГОУ «Московский университет
МВД России», г. Рязань
Ивличева Наталья Александровна — кандидат физико-матема-
тических наук, доцент кафедры математических и естествен-
нонаучных дисциплин, Рязанское высшее воздушно-десантное
командное училище (Военный институт), г. Рязань
Ильина Светлана Анатольевна — преподаватель, ГОУ СПО
«Вятско-Полянский механический техникум», Кировская об-
ласть
Канарёв Денис Игоревич — участковый уполномоченный мили-
ции, старший лейтенант милиции, Управление внутренних
дел по городу-курорту Анапа МВД РФ, г. Анапа
Кобзев Георгий Анатольевич — доктор физико-математических
наук, профессор, заведующий отделом «Теплофизический
центр данных по свойствам важных для энергетики веществ и

- материалов», Российская академия наук, Объединённый институт высоких температур, Москва
- Козлов Андрей Олегович* — аспирант, Институт информатизации образования Российской академии образования, Москва
- Кормилицына Татьяна Владимировна* — доцент кафедры информатики и вычислительной техники, ГОУ ВПО «Мордовский государственный педагогический институт имени М.Е. Евсевьева», г. Саранск
- Кулямин Виктор Вячеславович* — старший научный сотрудник, Институт системного программирования РАН, Москва
- Лиликович Сергей Александрович* — слушатель Малой академии наук Украины при Ровеньковском факультете Луганского национального университета имени Тараса Шевченко, г. Ровеньки, Луганская область, Украина
- Логинов Анатолий Владимирович* — ассистент кафедры информационных технологий и систем, Луганский национальный университет имени Тараса Шевченко, г. Луганск, Украина
- Мироненко Евгений Владимирович* — ученик 9-го класса МОУ гимназии № 11, г. Елец
- Насташук Наталья Александровна* — кандидат педагогических наук, ст. преподаватель кафедры информатики и вычислительной техники ГОУ ВПО «Омский государственный педагогический университет», г. Омск
- Панюкова Татьяна Анатольевна* — доцент, Южно-Уральский государственный университет, г. Челябинск
- Петренко Ольга Леонидовна* — доцент, Московский институт открытого образования, Москва
- Подаев Михаил Валерьевич* — ассистент кафедры математического анализа и элементарной математики, ГОУ ВПО «Елецкий государственный университет им. И.А. Бунина», г. Елец
- Саенко Игорь Борисович* — ведущий научный сотрудник, Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИ РАН), г. Санкт-Петербург
- Сафонов Владимир Иванович* — доцент кафедры информатики и вычислительной техники, ГОУ ВПО «Мордовский государственный педагогический институт им. М.Е. Евсевьева», г. Саранск
- Сафонова Людмила Анатольевна* — доцент кафедры информатики и вычислительной техники, ГОУ ВПО «Мордовский государственный педагогический институт им. М.Е. Евсевьева»,

г. Саранск

Ситникова Людмила Дмитриевна — ассистент кафедры информационных технологий, ГОУ ВПО «Тульский государственный педагогический университет им. Л.Н. Толстого», г. Тула

Смирнова Мария Николаевна — ст. преподаватель, зав.кафедрой информатики, филиал ГОУ ВПО «Кузбасский государственный технический университет», г. Прокопьевск

Соловьев Роман Юрьевич — преподаватель, инженер-электроник, ГОУ СПО «Вятско-Полянский механический техникум», Кировская область

Спендер Виктор Анатольевич — доктор военных наук, профессор кафедры безопасных информационных технологий, ГОУ ВПО «Санкт-Петербургский государственный университет информационных технологий, механики и оптики», г. Санкт-Петербург

Табакман Алексей Петрович — выпускник кафедры информационной безопасности, ГОУ ВПО «Московский государственный институт радиотехники, электроники и автоматики» МИРЭА, Москва

Трахтенгерц Михаил Самойлович — кандидат технических наук, старший научный сотрудник, ведущий научный сотрудник отдела «Теплофизический центр данных по свойствам важных для энергетики веществ и материалов», Российская академия наук, Объединённый институт высоких температур, Москва

Фисков Михаил Михайлович — начальник отдела администрирования сетей и баз данных ЦИТ, ГОУ ВПО «Сибирский государственный аэрокосмический университет им. академика М.Ф. Решетнёва», г. Красноярск

Хорошилов Алексей Владимирович — научный сотрудник, Институт системного программирования РАН, Москва

Черный Владимир Львович — начальник отдела образовательных проектов компании ALT Linux, Москва

Шалеев Антон Олегович — младший инженер ПО, Siemens IT Solutions and Services, г. Воронеж

Яфаева Роза Равильевна — кандидат технических наук, доцент, зав. кафедрой информационных технологий, ГОУ ВПО «Тульский государственный педагогический университет им. Л.Н. Толстого», г. Тула

Научно-методическое издание

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ НА БАЗЕ
СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Материалы Всероссийской научно-практической конференции
с международным участием

Технический редактор — Н.П. Безногих
Техническое исполнение — В.М. Гришин
Переплёт и обложка выполнены
в МУП «Типография» г. Ельца

Лицензия на издательскую деятельность
ИД № 06146. Дата выдачи 26.10.01
Формат 60x84/16. Гарнитура Times. Печать трафаретная.
Усл.-печ.л. 13,4 Уч.-изд.л. 13,6
Тираж 200 экз. (1-й завод — 50-100 экз). Заказ № 84

Отпечатано с готового оригинала-макета
на участке оперативной типографии
Елецкого государственного университета им. И.А. Бунина

Государственное образовательное учреждение
высшего профессионального образования
«Елецкий государственный университет им. И.А. Бунина»
399770, г. Елец, ул. Коммунаров, д.28