



СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ОБРАЗОВАНИИ

**Сборник трудов Всероссийской конференции
(г. Челябинск, 25--26 марта 2009 г.)**



Министерство образования и науки Российской Федерации
Федеральное агентство по образованию
Южно-Уральский государственный университет
Кафедра экономико-математических методов и статистики
Институт дополнительного образования
Alt Linux

681.3(063)
С25

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ОБРАЗОВАНИИ

Сборник трудов Всероссийской конференции
(г. Челябинск, 25–26 марта 2009 г.)

Под редакцией А.В.Панюкова

Челябинск
Издательство ЮУрГУ
2009

УДК [681.3:Ч30/49](063)
С25

Рецензенты:

д.ф.-м.н., профессор Л.Б. Соколинский, д.ф.-м.н., профессор М.М. Кипнис

С25 Свободное программное обеспечение в образовании: сборник трудов Всероссийской конференции (г. Челябинск, 25–26 марта, 2009 г.) / под ред. А.В. Панюкова. – Челябинск: Изд-во ЮУрГУ, 2009. – 121 с.

Сборник содержит доклады Всероссийской научно-практической конференции «Свободное программное обеспечение в образовании» (Челябинск, 25–26 марта 2009 г.). В представленных работах рассмотрены вопросы разработки и внедрения свободного программного обеспечения (СПО), а также актуальные задачи, связанные с практикой использования СПО в учебном процессе (разработка, внедрение, методика преподавания).

Сборник адресован научным работникам, специалистам в области проектирования программ, разработки и внедрения СПО в образовательные учреждения. Материалы представляют интерес для преподавателей, школьных учителей, а также студентов и аспирантов, обучающихся на специальностях, связанных с разработкой, внедрением и использованием информационных технологий.

УДК [681.3:Ч30/49](063)

© Издательство ЮУрГУ, 2009.

ОГЛАВЛЕНИЕ

| | |
|---|----|
| <i>Предисловие</i> | 4 |
| РАЗРАБОТКА И ВНЕДРЕНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ | 8 |
| <i>Курячий Г.В.</i> Динамика внедрения свободного программного обеспечения в образовательной сфере (по материалам конференций «Свободное программное обеспечение в высшей школе») | 8 |
| <i>Сыромятников Е.Л., Курячий Г. В.</i> Командная разработка свободных учебных материалов по проекту «Документирование пакета свободного программного обеспечения» | 14 |
| <i>Ненахов А. В., Мещерякова А. Н.</i> Опыт внедрения Debian GNU/Linux, Ubuntu в образовательных учреждениях г. Челябинска | 19 |
| <i>Панюкова А.А.</i> Дистрибутив «Школьный сервер» | 23 |
| <i>Панюкова А.А.</i> Дистрибутив ALT Linux Children: опыт и перспективы..... | 27 |
| <i>Аппель И.В.</i> jThinker – среда визуализации мыслительных процессов в теории организаций..... | 31 |
| <i>Горбик В.В., Панюков А.В.</i> Развитие программного обеспечения точных дробно-рациональных вычислений для параллельных и распределенных вычислительных систем | 35 |
| <i>Игнатович А.С.</i> О применении GNU OCTAVE в моделировании системной динамики | 41 |
| <i>Рожков С.А.</i> Терминальные решения на предприятиях | 46 |
| <i>Тычинин С.А.</i> Разработка программы для решения задачи MAX TSP с использованием свободной библиотеки GOBLIN | 50 |
| ПРАКТИКА ИСПОЛЬЗОВАНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УЧЕБНОМ ПРОЦЕССЕ | 53 |
| <i>Беспалова П. В.</i> Использование систем электронного документооборота при подготовке специалистов-документоведов..... | 53 |
| <i>Воронин И. В.</i> Внедрение свободного программного обеспечения в процесс обучения преподавателей | 55 |
| <i>Волченкова К.Н.</i> Свободное программное обеспечение в подготовке будущих переводчиков | 60 |
| <i>Зильберман М.А.</i> Преподавание компьютерной графики в школах с использованием свободного программного обеспечения..... | 64 |

| | |
|--|-----|
| <i>Зюляркина Н.Д., Гаврилова Т.П.</i> Применение системы GАР в научных исследованиях | 69 |
| <i>Карманов М.Л.</i> Использование свободного программного обеспечения для обучения студентов в рамках отдельных курсов | 71 |
| <i>Краев Е.С., Габова О. Ф.</i> Проект внедрения СПО в образовательный процесс Верещагинского МОУ «Гимназия». Второй этап | 73 |
| <i>Нилова С.В.</i> Свободное программное обеспечение в системе образования: Компетентностный подход | 77 |
| <i>Павлова Н. С.</i> Опыт миграции на свободное программное обеспечение в КЮТ «Приборист» | 84 |
| <i>Панюкова Т.А.</i> Разработка системы оценок и контрольных заданий для работы в графическом редакторе GIMP | 88 |
| <i>Первушина А.В.</i> Развитие творческого потенциала учащихся средствами свободного программного обеспечения на примере использования графического редактора Tux Paint на уроках информатики | 93 |
| <i>Прокопов И.И.</i> ОС FREEBSD как платформа для обеспечения безопасности сетевых технологий | 95 |
| <i>Рожков А.В., Гаврилова Т.П.</i> Применение научных пакетов с открытым кодом при работе со школьниками и студентами младших курсов | 100 |
| <i>Ханнанова С.Т.</i> Особенности внедрения свободного программного обеспечения в условиях лицейского образования | 104 |
| <i>Цытович П.Л.</i> Реализация контекстного подхода к обучению специалистов по информационным технологиям средствами портала ITEM74.RU | 112 |
| <i>Якшин М.М.</i> Свободное программное обеспечение для внешкольных занятий с детьми | 115 |
| <i>Мирасов В.Ф., Дударева В.И.</i> Интегрированные занятия по информатике, и теории вероятностей и математической статистике с использованием свободно-распространяемого математического пакета «Scilab» | 115 |

ПРЕДИСЛОВИЕ

Сложившаяся до недавнего времени практика внедрения информационных технологий такова, что все участники процесса, как ни странно, заинтересованы в большей стоимости используемого программного обеспечения. Продавец – потому что его прибыль обычно составляет определённый процент от объёма продаж, покупатель (юридическое лицо) – потому что получает дополнительное финансирование в виде «откатов». Наличие специальных цен на программное обеспечение для государственных структур и учебных заведений можно рассматривать как оплату внедрения «нужного» продукта в тех организациях, от которых часто зависит выбор того или иного производителя за счёт остальных пользователей.

Даже массированная реклама о необходимости использования лицензионного программного обеспечения, создавая отрицательное отношение к пиратским копиям, попутно формирует мнение о «единственности» операционной системы. Перечень подобных аргументов можно продолжить. Любое учреждение, особенно государственное, не свободно в выборе программного обеспечения, в частности, обеспечивающего взаимодействие с государственными и ведомственными органами. Работа с налоговой отчётностью, пенсионными документами, результатами ЕГЭ, протоколами лицензирования и множеством других *обязательных* документов зачастую возможна только с помощью единственного программного продукта.

В то же время практически все типовые задачи можно решить с использованием лицензионного, но абсолютно бесплатного программного обеспечения. Как правило, такое программное обеспечение имеет открытый, т.е. общедоступный программный код.

В подавляющем большинстве офисов работа ограничена узким кругом задач: работа с документами, электронной почтой, поиск информации в Интернете, в период отдыха – игры, прослушивание музыки и т. д. Всё это прекрасно реализовано в различных операционных системах на базе ядра Linux. Поддержка операционных систем Linux осуществляется всеми ведущими производителями аппаратной части. Все пользовательские операции на сегодня выполняются в графическом интерфейсе, что гарантирует достаточно быстрый переход на эти системы. Не случайно, что в средствах массовой информации в последнее время часто появляются сообщения, что на Linux переходят муниципалитеты крупных европейских городов.

В условиях малого и среднего бизнеса одним из сдерживающих факторов полного перехода на Linux долгое время являлось отсутствие бухгалтерских программ. Сейчас ситуация изменилась. Выпущена официальная версия 1С для платформы Linux.

Приоритетный национальный проект «Образование» предписывает внедрение свободного программного обеспечения в образовательные учреждения Российской Федерации.

Кафедра «Экономико-математические методы и статистика» ЮУрГУ уже более трех лет сотрудничает с компанией ALT Linux. Один из результатов сотрудничества – создание методического и программного обеспечения серии обучающих занятий для детей по компьютерной графике. Предистория проста: летом 2006–08 гг. в компьютерном классе, расположенном на территории спортивно-оздоровительного комплекса ЮУрГУ, сотрудники этой кафедры провели серию обучающих занятий для детей, отдыхающих в ДОЛ «Березка» и б/о «Наука». Тематика занятий включала выполнение творческих заданий с помощью графических программ, создание простейшей анимации, коллажей из фотографий и рисунков, работу с программами монтажа, создание видеоклипов. На компьютерах установлена свободно распространяемая ОС Linux, инсталлятор которой специально сконфигурирован в компании ALT Linux под особенности компьютерного класса базы.

Неформальный подход к возложенной на сотрудников кафедры задаче обучения детей во время летнего отдыха привел к тому, что наработанный опыт позволил организаторам обучения сделать доклады об особенностях применения свободно распространяемого программного обеспечения как на февральской научно-методической конференции ЮУрГУ (2008 г.), так и на других международных и региональных конференциях.

В октябре 2008 г. в Челябинске впервые был проведен информационный семинар «Ознакомление и демонстрация возможностей современной учебной техники и компьютерных технологий» для школ и учреждений профобразования с участием представителей компании ALT Linux. В рамках семинара состоялся доклад начальника отдела образовательных программ В.Л.Черного (г. Москва) с презентацией программного обеспечения компании ALT Linux. В семинаре приняли участие более 260 представителей образовательных учреждений (зам. директоров по научной и учебной работе, методистов, администраторов компьютерных классов, учителей информатики и др.). Более 60 человек посетили мастер-класс, который после завершения заседания семинара провели представители компании ALT Linux и преподаватели кафедры «Экономико-математические методы и статистика» ЮУрГУ. Семинар показал наличие в нашем регионе достаточного большого круга заинтересованных во внедрении СПО участников.

В работах, представленных в настоящем сборнике, рассмотрены вопросы разработки свободного программного обеспечения (СПО), а также актуальные вопросы, связанные с практикой использования СПО в учебном процессе: опыт внедрения и методика преподавания.

*Ответственный редактор
А.В. Панюков*

РАЗРАБОТКА И ВНЕДРЕНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ДИНАМИКА ВНЕДРЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ОБРАЗОВАТЕЛЬНОЙ СФЕРЕ (ПО МАТЕРИАЛАМ КОНФЕРЕНЦИЙ «СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ВЫСШЕЙ ШКОЛЕ»¹⁾)

Курячий Г.В.

george@altlinux.org
ALT Linux, г. Москва

Рассмотрен процесс вхождения свободного программного обеспечения в инфраструктуру и проблематику учебных организаций. Используются материалы конференций «СПО в высшей школе» последних четырёх лет. Отмечены тенденции в изменении тематики и их возможные причины.

ВВЕДЕНИЕ

Внедрение свободного программного обеспечения в инфраструктуру, учебный и административный процесс учебного заведения сопровождается множеством разноплановых проблем. Представляется важной необходимостью отслеживать, на какой стадии находится этот процесс «в целом».

Значимую картину даёт систематизация тематики докладов конференции «Свободное программное обеспечение в высшей школе», которая четвертый год традиционно проводится в г. Переславле. Изменения тематики позволяют достаточно определённо выявлять тенденции сближения СПО и образовательной среды.

В частности можно заметить смещение проблематики с собственно СПО на обеспечение взаимодействия с «остальным миром» посредством СПО.

ПРЕДЫСТОРИЯ

Конференция «Свободное программное обеспечение в высшей школе» проводится в Переславле-Залесском с 2006 года в дни зимних студенческих каникул. В числе организаторов конференции – Институт Программных Систем РАН и НОУ «Университет города Переславля им. А. К. Айламазяна», выступающие также в роли принимающих организаций.

¹ Институт Программных Систем РАН и НОУ «Университет города Переславля им. А.К. Айламазяна» выступают в роли принимающих организаций, поэтому считаем своим долгом выразить им горячую благодарность за блестящую техническую и организационную поддержку.

Задача конференции вполне традиционна для подобного рода мероприятий: собрать в одно время и в одной точке как можно больше заинтересованных и активных людей, дать им возможность наладить связи друг с другом, организовать общение, которое не затруднено посторонними делами и/или непривычкой к электронным средствам связи.

Принцип формирования корпуса докладов прост: в каждом докладе должна затрагиваться главная тема, сближение свободного ПО и высшей школы. Поэтому принимаются и рассказы об успешном внедрении СПО в ВУЗах, и исследования частных проблем того или иного свободного программного продукта применительно к образовательным или научным целям, и методические рекомендации по построению учебного или плана на базе СПО. Таким образом, аудитория конференции профессионально неоднородна: на неё приезжают и начальники ИТ-подразделений, и преподаватели, и системные администраторы, и студенты.

Объединяет участников конференции принадлежность к образовательной среде и знакомство с СПО. Пожалуй, три года назад главной целью было запустить процесс создания *сообщества* людей, работающих со свободными программами в образовании. Уже из материалов 2006 года видно, что участники конференции в большинстве своём мало что знают друг о друге и занимаются нелёгким, подчас напрямую революционным, трудом в одиночку, не надеясь на помощь извне. И это несмотря на то, что в нашей стране существует мощное преподавательское сообщество, а свободное ПО в отрыве от сообщества пользователей и разработчиков вообще немыслимо!

2006

В самом деле, если отсеять доклады «по случаю», темы Первой конференции разобьются на четыре части:

Общетеоретические рассуждения и обзоры. К сожалению, большинство таких докладов показывают, как «мысли на тему» бывают не похожи на действительность. Не имея опыта, легко перестраховаться от невзгод вымышленных и проглядеть настоящие. Недостаток непосредственных знаний (например, об устройстве свободного сообщества) может привести напряжённо работающую мысль в края совершенно фантастические. Сказывается ли здесь специфика СПО, или же такова судьба любого отвлечённого построения, утверждать не берёмся.

Заметки об особенностях любимого свободного инструмента. Довольно интересные рассказы о том, как применять некий программный продукт в узкой области, освоенной самим докладчиком. В некоторых случаях эта область довольно узкая (как, например, доклады А.В. Столярова и др.), иногда – достаточно широкая, чтобы быть предметом пристального внимания (особенно порадовал В.В. Житников с докладом о матема-

тическом пакете Maxima). К сожалению, сведений о непосредственном использовании этих пакетов в учебном процессе обычно нет.

Сообщения об использовании СПО в качестве одной из составляющих учебного (исследовательского) процесса. В 2006 году это была самая обнадёживающая тема конференции. Доклады Е.А. Чичкарева и Н.В. Назаренко (о моделях металлургических агрегатов), а также Е.Р. Алексеева и О.В. Чесноковой (о преподавании дисциплин «Информатика» и «Вычислительная математика» с использованием пакета SciLab) превзошли наши ожидания. В обоих случаях речь шла не только о внедрении СПО (в учебном процессе и даже на производстве), но и – что важнее – о внедрении, опирающемся на *преимущества* выбранных программных продуктов. И это – на Украине, где ситуация со свободным ПО более сложная, чем в России.

Однако именно в этих докладах наиболее очевиден общий недостаток того времени: работы ведутся в одиночку (в лучшем случае – с помощью аспирантов), никаких публичных информационных ресурсов не создаётся. На прямой вопрос: «Где добыть ваши материалы?» авторы недоумённо отвечают: «А что, разве это кому-то нужно?», а затем смущаются: «Да у меня там всё сырое, неготовое...». Контраст двух подходов – участника свободного сообщества и энтузиаста-индивидуала – к одному и тому же предмету поражает. Активный пользователь свободного программного продукта *уверен* в том, что результаты его трудов нужны и интересны многим, и именно поэтому он вправе ожидать помощи там, где его собственных сил не хватает. Энтузиаст-преподаватель не привык видеть рядом хоть кого-нибудь, столь же вовлечённого в процесс, как и он сам; в результате он идёт вперёд, пока хватает сил, оставляя за спиной множество материала, который не имеет смысла ни обрабатывать, ни публиковать (самому-то всё и так понятно, а больше на белом свете никого нет).

Рассказ о преподавании и изучении непосредственно свободного ПО. Подробные и толковые рассказы о том, как именно следует изучать, например, некоторый дистрибутив на базе GNU/Linux, или основы разработки ПО для него. Здесь СПО является уже не инструментом, а объектом изучения, и оттого, как ни странно, такие рассказы добавляют не много нового к общей теме. Свободные программы *приспособлены* к тому, чтобы их изучали, а главный вопрос – как добиться от начальства того, чтобы Linux *можно было* изучать – обычно лежит за рамками доклада.

2007

В 2007 году стало заметно, что Первая конференция не прошла бесследно. К примеру, уже знакомый нам коллектив авторов из Донецка во главе с Е. Алексеевым обзавёлся сайтом, на котором начали появляться материалы (совместная работа, к сожалению, не предусмотрена).

Что более важно: появился новый тип докладов – **миграционные истории**. Речь в них идёт о применении СПО, но не в качестве учебного пособия или вычислительного инструмента, а в качестве *платформы*, на которой организуется поддержка и учебного, и административного процесса. Таким образом, СПО выбирается в качестве основной платформы не только за программную специфику, но и за общие эксплуатационные характеристики (стоимость владения, надёжность и т. п.). Среди прочих обращает на себя внимание доклад И.А. Хахаева, посвящённый *уже реализованной* миграции образовательного учреждения на полностью лицензионное ПО, причём свободное ПО принимается как базовое, а несвободное выбирается только вынуждено.

2008

Третья конференция, прошедшая в начале февраля 2008 года, собрала неожиданно внушительное количество *слушателей*, примерно половину всех участников (в прошлые годы аудитория практически полностью состояла из докладчиков). Конференция получилась более солидной, её не обошли вниманием «наверху» (с докладом выступил Н.А. Юренко, референт Департамента государственной политики и нормативно-правового регулирования в сфере образования МОН РФ). Впервые программный комитет конференции вынужден был отклонить довольно много докладов. Примечательно, что большинство из отклонённых докладов не имели отношения к заявленной теме (часто за свободные программы авторы принимали нечто сомнительное с лицензионной точки зрения, вроде т. н. «freeware»).

Изменилось и содержание докладов. Доклады о внедрении того или иного СПО в учебный и исследовательский процесс перестали быть жёстко привязаны к области информационных технологий. Оказались затронутыми греческий язык (М.В. Быков), филология (М.В. Пономарева), астрономия (Т.А. Воронина).

Доклады об особенностях программных продуктов приняли более выраженный *сравнительный* характер (И.А. Хахаев; Е.А. Чичкарев). Разговор о возможностях СПО превращается в разговор о критериях выбора среди многообразия свободных программ.

Об итогах 2008 года: материалы донецкого коллектива превратились в книгу, а Санкт-Петербургский Торгово-экономический институт получил в своё распоряжение специализированный дистрибутив, об отсутствии которого так сокрушался И.М. Хахаев. Кстати, этой характерной для СПО возможности – самостоятельному *составлению* сборника программных продуктов под имеющуюся методическую базу – был посвящён и доклад Е.Р. Алексева. Под свою экзотическую задачу (десятидневный кружок в детском летнем лагере) А.А. Панюкова разработала методику, и собрала специализированный дистрибутив.

2009

Наконец, четвёртая конференция, прошедшая в последних числах января 2009 года, показала устойчивость отмеченных выше тенденций в суровых условиях экономического кризиса. Думается, что именно этим объясняется, почему конференция собрала несколько меньше участников, чем в предыдущий год, и куда меньше слушателей. Нередко участник мог только предоставить свой доклад, но на саму конференцию не попадал.

Общая тематика конференции 2009 года наследовала тематику 2008 г. Тема «Использование СПО в качестве составляющей учебного процесса» стала преобладающей. «Миграционные истории» продолжились в докладах А. Дребеня и (косвенно) Н. Ивановой.

Вместе с тем можно отметить и очевидное расширение области *решаемых* участниками конференции задач. В некоторых случаях использование СПО перешло на «метауровень»: так, в докладе Э. Хайруллова рассматриваются вопросы *восприятия* процесса внедрения СПО, в докладе Л. Яковлева – организационные проблемы такого внедрения, в докладе И. Хахаева – вопросы лицензирования учебного заведения (не путать с лицензированием ПО!), внедрившего СПО. Сильный эффект произвёл доклад А. Дребеня, показавший, что приведение *внешнего* вида графической среды Linux (включая такие тонкости, как градиентная раскраска декораторов) в соответствие шаблонам поведения пользователей снимает *большинство* внедренческих проблем.

Неожиданно тревожной стала совместно сформулированная и горячо обсуждаемая тема круглого стола конференции: «СПО и навязанные программные продукты». Суть проблемы: любое учреждение, особенно государственное, не свободно в выборе ПО, в частности, обеспечивающего взаимодействие с государственными и ведомственными органами. Работа с налоговой отчётностью, пенсионными документами, результатами ЕГЭ, протоколами лицензирования и множеством других *обязательных* документов зачастую возможна только с помощью единственного программного продукта. «Навязанное ПО» обычно отличается невысоким качеством, неработоспособно в свободном программном окружении, а подчас даже несовместимо по форматам данных ни с одним свободным программным продуктам.

«Навязанное ПО», по мере его распространения, становится всё более серьёзным препятствием в процессе внедрения СПО, несмотря на успехи энтузиастов на местах (доклады И. Хахаева и А. Дребеня на круглом столе). Конференция 2009 года приняла резолюцию с предложением ввести контроль совместимости ведомственного ПО.

МАТЕРИАЛЫ КОНФЕРЕНЦИЙ

- Материалы первой конференции:
http://altlinux.ru/company_news/free_software_in_higher_education_conference.html, <http://heap.altlinux.org/pereslavl2006/> .
- Материалы второй конференции:
http://altlinux.ru/company_news/pereyaslavl_2007.html,
<http://heap.altlinux.org/pereslavl2007/> .
- Материалы третьей конференции:
http://altlinux.ru/community_news/pereslavl_winter_2008.html,
<http://heap.altlinux.org/pereslavl2008/> .
- Материалы четвертой конференции:
<http://www.altlinux.ru/media/book-thesis-Pereslavl-2009-4.pdf> .

ЛИТЕРАТУРА

1. Четвёртая конференция «Свободные программы в высшей школе». Тезисы докладов. – М.: ALT Linux, 2009.
2. Третья конференция «Свободные программы в высшей школе». Тезисы докладов. – М.: Институт Логики, 2008.
3. Вторая конференция «Свободные программы в высшей школе». Тезисы докладов. – М: Институт Логики, 2007.
4. Первая конференция «Свободные программы в высшей школе». Тезисы докладов. – М: Институт Логики, 2006.

КОМАНДНАЯ РАЗРАБОТКА СВОБОДНЫХ УЧЕБНЫХ МАТЕРИАЛОВ ПО ПРОЕКТУ «ДОКУМЕНТИРОВАНИЕ ПАКЕТА СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Сыромятников Е.Л., Курячий Г. В.

george@altlinux.org

МГУ им М. В. Ломоносова, факультет ВМиК, г. Москва

Приведена история создания рабочей группы по документированию пакета свободного программного обеспечения (ПСПО), организации процесса создания учебного материала, структура получившегося продукта и поддерживающей его технологии. Делается попытка истолкования положительных и отрицательных результатов работы и обобщения полученного опыта.

ВВЕДЕНИЕ

Разработка и пилотное внедрение «Пакета Свободного Программного Обеспечения для образовательных учреждений РФ» включала в себя задачу подготовки учебных курсов для десяти категорий пользователей.

Задача «в кратчайшие сроки подготовить 10 учебных курсов по новому программному продукту силами разработчиков этого программного продукта» не выглядит разумно поставленной. Для того чтобы её всё-таки решить, пришлось пойти на некоторые технические, организационные и методические уловки,

В результате этого решения были выработаны, как нам представляется, заслуживающие интереса методы как взаимодействия внутри команды, так и собственно формирования контента.

Текст представлен в форме тезисного плана, так как рамки доклада не позволят развернуть его в полном объёме; планируется освещение только тех сторон процесса, которые вызовут наибольший интерес участников конференции.

ПОСТАНОВКА ЧАСТНОЙ ЗАДАЧИ

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

Допущения:

- 1) модульная структура курсов;
- 2) отказ от покрытия «чистыми» лекционными материалами.

ОРГАНИЗАЦИЯ РАБОЧЕГО ПРОЦЕССА

Рабочий процесс:

- подготовка модулей;
- подготовка новых материалов;
- подготовка имеющихся материалов.

Организация:

- формирование дисциплины совместной работы (<http://uneex.ru/PSPO/Policy>);
 - подготовка технической базы (Wiki-платформы MoinMoin с дополнительным программированием и установленной дисциплиной работы);
 - распределение работ по участникам;
 - отслеживание состояния работ.

Наполнение курсов материалами (подготовка модулей)

Модуль:

- **лекционный минимум** – развёрнутый тематический план части курса;
- **итоговый текст** – лекционный минимум, дополненный *обязательными* ссылками на материалы, и, при необходимости, комментариями.

Организация (<http://uneex.ru/PSPO/Policy/Modules>):

1. Архитектор составляет «Лекционный минимум»:
 - персональная ответственность;
 - критерий выбора исполнителя: знакомство с темой модуля.
2. Исполнитель из команды подбирает ссылки на материалы, содержащие информацию по каждой теме:
 - критерий качества: покрытие материалами всего лекционного минимума, желательно, из разнородных источников;
 - все использованные внешние материалы импортируются (см. далее).

Подготовка имеющихся материалов

Материал: любой документ, содержащий информацию в рамках курсов.

- Многообразии форматов:
 - текст в wiki-разметке (обрабатывается wiki-платформой), просматриваемый в браузере.
 - HTML-дерево, которое также просматривается в браузере.
 - файлы различных форматов (например, PDF), просматриваемые сторонними программами.

- Отчуждаемость материала:
 - обязательный импорт сторонних материалов в общий корпус (<http://uneex.ru/PspoMaterials>);
 - паспорт материала:
 - название, авторство и источник;
 - формат материала (wiki, HTML, ODT, PDF, ...)
- (<http://uneex.ru/PspoMaterials/FormatInfo>);
- лицензия, под которой распространяется материал. В рамках проекта использовались материалы только под свободными лицензиями. (<http://uneex.ru/PspoMaterials/LicenseInfo>);
 - аннотация (если есть).

В случае необходимости материалы добавляют (и оформляют паспорт) лица, ответственные за работу над модулем (на последнем этапе для наполнения модулей в основном использовались уже имеющиеся материалы, так как накопленный их корпус покрывал курсы чуть менее, чем полностью).

Подготовка новых материалов (Also Sprach Zarathustra)

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

Организация:

- принцип «один человек – одна задача» (<http://uneex.ru/PspoTasks>);
- принцип персональной ответственности (<http://uneex.ru/PspoClasses/Status>);
- содержательное редактирование сведено к минимуму (иногда с небольшой потерей качества);
- перераспределение «затянутых» работ.

ИСПОЛЬЗОВАНИЕ MOINMOIN

MoinMoin:

- современная Wiki-платформа;

- имеет минимальные требования по установке;
- написана на Python, легко расширяется и модифицируется.

Проделанные модификации:

- простейшие средства *планирования и мониторинга* рабочего процесса;
- хранение *иерархии файлов*, связанной с определённой страницей;
- автоматическое создание и оформление ссылок на объекты.

Как расширять функциональность MoinMoin

- Новые синтаксические конструкции – *Macros*.
- Новые способы обработки контента – *Actions* (действие указывается в URL).
 - Новые форматы входного текста (например, другие Wiki, CSV, DOT, HTML) – *Parsers*.
 - Новые форматы вывода (не только HTML, а, например, PDF, DocBook, ODT) – *Formatters*.
 - Новое оформление – *Themes*.
 - Любое расширение – обычный модуль Python, имеющий простой и прозрачный доступ к внутренним API MoinMoin (рекомендуется изучить в первую очередь возможности модуля `MoinMoin.util`).

Новое *действие* (action) обычно сопровождается новым *макросом* (macro), создающим ссылку, содержащую действие, или исправлением *оформления* (theme).

Планирование и мониторинг рабочего процесса (макро *ExtractModules*)

Свойства модуля:

- зависимость по тематике от других модулей (т. н. «Необходимые знания»);
- готовность (уровень и сроки, сообразно рабочему процессу).

Задача: вычислить готовность всего курса и список требований к знаниям слушателей.

Решение (<http://uneex.ru/PSPO/Policy/Lectures>):

- каждый модуль содержит таблицу, отмечающую готовность и зависимости (обновляется исполнителем);
- курс формируется из списка модулей автоматически.

Хранение иерархии файлов, связанной с определённой страницей

Задача: работа с *материалами*, представленными в виде дерева файлов и каталогов (например, полными HTML-страницами) Решение: пара *действие+макро* (<http://uneex.ru/eSyr/WikiHack/Actions/MaterialFiles> и <http://uneex.ru/eSyr/WikiHack/Macros/MaterialFiles>), в которой использование макро `MaterialFiles` в тексте страницы раскрывается в изображение де-

рева файлов. Соответствующие манипуляции файлами – обращение к действию MaterialFiles и его вариациям.

Пример использования можно пронаблюдать на любой странице паспорта материала (<http://uneex.ru/PspoMaterials>).

Возможно, более правильным было бы модифицировать имеющийся макрос AttachFile, но на момент разработки это было признано нецелесообразным.

Полезные мелочи

- Для удобства формирования паспорта были созданы пара макрос/действие (<http://uneex.ru/eSyr/WikiHack/Macros/CreateMaterialPassport> / <http://uneex.ru/eSyr/WikiHack/Actions/CreateMaterialPassport>), которые позволяли создавать его путём заполнения формы, в результате чего на основании имеющегося шаблона генерировалась страница паспорта. Пример макроса можно пронаблюдать на странице со списком материалов (<http://uneex.ru/PspoMaterials>).

- Для удобства формирования информации о формате и лицензии были написаны небольшие макросы (<http://uneex.ru/eSyr/WikiHack/Macros/FormatInfo>, <http://uneex.ru/eSyr/WikiHack/Macros/LicenseInfo>), которые вставляли текст описания того или иного формата/лицензии по его названию. Описания брались с соответствующих страниц.

- Для удобочитаемости ссылок на материалы был написан макрос PassportLink (<http://uneex.ru/eSyr/WikiHack/Macros/PassportLink>), который использовал информацию со страницы паспорта (название, формат) для формирования текста ссылки (и иконки для неё).

РЕЗУЛЬТАТ

Структура получившегося корпуса материалов.

- **Материал** = Паспорт + Файлы; внутренний и внешние материалы.
- **Модуль** = Лекционный Минимум * Материалы + Комментарии.
- **Курс** = сумма Модулей + методические рекомендации.
- Побочный объект: **Книга** = самодостаточная сумма Материалов (<http://uneex.ru/Books> , «Пользовательские и технические аспекты ПСПО (цикл лекций)»).

ЗАМЕЧАНИЯ И ПРЕДЛОЖЕНИЯ

- Высокая роль внутренней мотивации.
- Необходимость автоматического разделения «готовой» и «разрабатываемой» частей при отчуждении.
- Необходимость отслеживать устаревание.

ОПЫТ ВНЕДРЕНИЯ DEBIAN GNU/LINUX, UBUNTU В ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЯХ Г. ЧЕЛЯБИНСКА

Ненахов А. В., Мещерякова А. Н.

andrew.nenakhov@redsolution.ru

директор ООО «Редсоллюшн», г. Челябинск

anastasia@redsolution.ru

менеджер по проектам ООО «Редсоллюшн», г. Челябинск

Опыт внедрения свободного программного обеспечения (СПО) в образовательных учреждениях (ОУ) г. Челябинска. Методы внедрения СПО. Обучение пользователей. Техническое обслуживание. Сложности при переводе ОУ на СПО. Пути решения.

ВВЕДЕНИЕ

Внедрение свободного программного обеспечения в образовательные учреждения является частью приоритетного национального проекта «Образование». В 2008 г. на базе отечественного дистрибутива AltLinux началось полномасштабное внедрение СПО в трех пилотных регионах: Томской области, Пермском крае и республике Татарстан. Ввиду наметившейся тенденции перехода всей системы образования на СПО, такая работа, хотя и не так масштабно, ведётся и в нашем регионе.

В г. Челябинске ООО «Редсоллюшн» с 2007 года успешно реализует проекты по переводу образовательных учреждений города на СПО на базе дистрибутивов Debian GNU/Linux и Ubuntu. На сегодняшний день уже завершены работы по внедрению СПО в МОУ СОШ № 123, Профессиональном лицее № 10, МОУ ДПО «Учебно-методический центр г. Челябинска», в Челябинском институте развития профессионального образования (ЧИРПО).

Опыт внедрения СПО в ОУ г. Челябинска

ООО «Редсоллюшн» – инновационная компания г. Челябинска, специализирующаяся на предоставлении консалтинговых услуг в области информационных технологий и разработки высококачественного программного обеспечения.

В 2007 году Министерством экономического развития Челябинской области был проведен конкурс инновационных проектов предприятий и организаций Челябинской области. Одним из победителей был признан проект ООО «Редсоллюшн» по внедрению ОС GNU/Linux. Работа с образовательными учреждениями является логичным расширением деятельности предприятия в этом направлении. Основным препятствием на пути внедрения СПО является низкая осведомлённость пользователей о возможных альтернативах ОС Microsoft Windows, поэтому работа с детьми является наиболее эффективным способом упростить внедрение СПО в будущем.

В 2007 году усилиями ООО «Редсолюшн» на свободное программное обеспечение был переведен Профессиональный лицей № 10 г. Челябинска. СПО было установлено на всех административных ПК.

В начале 2008 года ООО «Редсолюшн» приступило к внедрению СПО в Челябинском институте развития профессионального образования (ЧИРПО). Значительная часть парка компьютеров, в т. ч. используемых в образовательном процессе, была переведена на GNU/Linux.

Также в 2008 году ООО «Редсолюшн» выиграло муниципальный грант по внедрению СПО в МОУ СОШ № 123. В данном образовательном учреждении на ОС GNU/Linux были переведены все административные ПК, а также компьютерные классы. С 1 сентября 2008 года обучающий процесс по дисциплине «Информатика и ИКТ» проводится полностью на базе СПО.

В конце 2008 года состоялось внедрение СПО в МОУ ДПО «Учебно-методический центр» Ленинского, Калининского, Советского и Metallургического районов г. Челябинска. Свободное программное обеспечение было установлено как на административных ПК, так и в компьютерных классах.

МЕТОДЫ ВНЕДРЕНИЯ СПО

В применяемых решениях ООО «Редсолюшн» основывается на дистрибутивах Debian GNU/Linux для серверов и Ubuntu для персональных рабочих станций. Первоначально на пользовательские ПК также устанавливался Debian, однако практика показала, что при сравнимых показателях надежности и отказоустойчивости Ubuntu является более дружелюбной для пользователей средой, благодаря чему мотивация пользователей к переходу на новое ПО значительно возрастает. Фактором, определившим выбор Ubuntu в качестве основного дистрибутива для рабочих станций является стабильность цикла разработки. Так, для внедрения изначально используются только т.н. LTS-версии (Long Term Support, с долгосрочной поддержкой) этой ОС. Изначально, значительное количество ошибок в новых версиях Ubuntu внушало опасения, но практика показала, что через достаточно короткое время (1–3 месяца) после релиза они оперативно исправляются, после чего стабильность Ubuntu и Debian практически идентична. Для работы с различными цифровыми образовательными ресурсами (ЦОР), использование которых на сегодняшний день является неотъемлемой частью образовательного процесса, в локальную вычислительную сеть образовательного учреждения включается терминальный сервер с установленной ОС Microsoft Windows Server 2003.

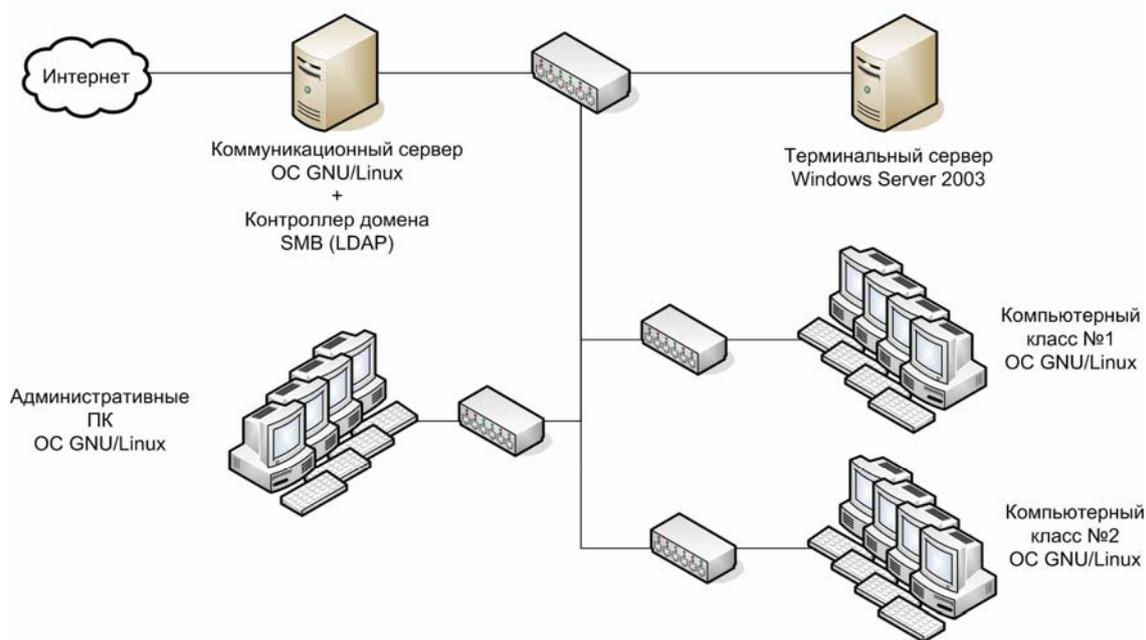
Решение об использовании терминального сервера на базе ОС Microsoft Windows обусловлено стремлением облегчить переход педагогического состава образовательного учреждения на СПО. Возможность работы с ЦОР, по мнению преподавателей ОУ, является безусловным достоинством

создаваемой системы. Однако, было принято решение сократить количество используемых ЦОР – из списка более чем в 100 программных продуктов были выбраны только те, применение которых действительно необходимо при проведении занятий. Такой подход заставил педагогов реально взглянуть на содержание ЦОР и детально проанализировать актуальность и необходимость их использования.

В результате, типовая схема локально-вычислительной сети ОУ выглядит следующим образом (см. рисунок).

ОБУЧЕНИЕ ПОЛЬЗОВАТЕЛЕЙ

Основной проблемой при переходе на СПО является человеческий фактор, а именно, нежелание пользователей изучать новые программные продукты. Стойкую мотивацию педагогических работников к «уже изученным и освоенным» программам возможно побороть только обучением пользователей основам работы с новым ПО.



Типовая схема ЛВС ОУ

Во всех образовательных учреждениях, где ООО «Редсолюшн» осуществляло внедрение СПО, были проведены курсы обучения пользователей. Помимо ознакомления пользователей с новыми программными продуктами из разряда СПО, в программу обучения входит курс лекций по основам работы в единой локальной сети. Кроме того, были разработаны методические рекомендации и руководства для пользователей по данным аспектам.

В Челябинском институте развития профессионального образования (ЧИРПО) были проведены курсы для системных администраторов. После курса лекций и практических занятий, данное образовательное учреждение полностью перешло на самообеспечение в части обслуживания парка компьютерной техники и установленного программного обеспечения.

ТЕХНИЧЕСКОЕ ОБСЛУЖИВАНИЕ

Что касается других образовательных учреждений, переведенных на СПО, то им оказывается техническая поддержка посредством удаленного администрирования. Созданная в компании «Редсолюшн» служба технической поддержки позволяет оперативно решать все возникающие у клиентов вопросы. Практика показывает, что большинство обращений связано не с функционированием локальной сети ОУ, а с использованием установленных программных продуктов. При этом после прохождения курса обучения, количество таких вопросов значительно сокращается.

СЛОЖНОСТИ ПРИ ПЕРЕВОДЕ ОУ НА СПО. ПУТИ РЕШЕНИЯ

Безусловно, при переводе образовательного учреждения на СПО возникает ряд трудностей, например, при переводе бухгалтерии. На сегодняшний день каждое ОУ использует свой набор бухгалтерских программ, большая часть из которых предоставлена им различными инстанциями (клиент-банки, программы обработки статистической отчетности, налоговые и пенсионные программы). Бухгалтерию филиалов МОУ ДПО «Учебно-методический центр г. Челябинска» удалось перевести на ОС GNU/Linux полностью, в остальных же ОУ, которые ООО «Редсолюшн» перевело на СПО, данная проблема решена путем создания гибридной сетевой инфраструктуры ОУ с одновременным использованием ОС GNU/Linux и ОС Microsoft Windows. Ниже представлены основные трудности, с которыми сталкивается ОУ при переходе на СПО, и способы их решения (см. таблицу).

ЗАКЛЮЧЕНИЕ

Все вышеперечисленные трудности, безусловно, будут решены при переходе большинства ОУ на СПО.

В настоящий момент компания «Редсолюшн» при поддержке Министерства экономического развития проводит ряд мероприятий по переводу коммерческих и бюджетных организаций Челябинской области на лицензионное ПО, а также по внедрению в них свободного программного обеспечения.

Таблица

| Задача | Решение |
|--|---|
| Невозможность запуска некоторых коммерческих программных продуктов, в том числе ряда цифровых образовательных ресурсов (ЦОР) | Подбор совместимых с Linux аналогов таких программ, а также использование гибридной сетевой инфраструктуры с одновременным использованием GNU/Linux и Windows |
| Наличие документов в закрытом проприетарном формате | Конвертирование в открытые форматы |
| Отсутствие драйверов для некоторых периферийных устройств, наличие оборудования с драйверами исключительно под Windows | Консультация со специалистами при приобретении оборудования |
| Непривычный для пользователей интерфейс программ | Обучение и консультирование пользователей |
| Отсутствие навыков администрирования ОС у технических специалистов | Обучение технических специалистов и/или аутсорсинг технической поддержки |

Кроме того, совместно с Управлением по делам образования г. Челябинска ООО «Редсоллюшн» разрабатывает поэтапный план массового перевода образовательных учреждений на свободное программное обеспечение.

ДИСТРИБУТИВ «ШКОЛЬНЫЙ СЕРВЕР»

Панюкова А.А.

mex3@altlinux.ru

выпускающий разработчик, ALT Linux, г. Москва

Предпосылки

Мининформсвязи России совместно с Рособразованием предложили в рамках конкурса «Разработка и апробация в пилотных субъектах Российской Федерации пакета свободного программного обеспечения для использования в общеобразовательных учреждениях Российской Федерации в 2007–08 годах» путем отбора лучших свободных программных решений, их адаптации под нужды отечественных ОУ и должной локализации разработать отечественный пакет свободного программного обеспечения (ПСПО) и провести его внедрение в пилотных субъектах Российской Федерации, а в 2009 году обеспечить установку данного решения на всех ПК, используемых в ОУ, в качестве альтернативы зарубежным коммерческим лицензионным программным продуктам.

Однако создание и внедрение ПСПО решает задачи обеспечения ОУ современным программным обеспечением, свободным от лицензионных отчислений лишь частично. Для решения задач полноценной организации сетевого информационного пространства общеобразовательного учреждения, его взаимосвязи с другими образовательными учреждениями необходимо создание принципиально новых программных продуктов, основанных на свободных решениях и открытых стандартах, не имеющих каких бы то ни было ограничений срока использования.

На сегодняшний день задача организации информационного пространства ОУ решается путем внедрения разнородных программных продуктов, работающих под управлением семейства операционных систем MS Windows, что требует значительных затрат на приобретение лицензий как на сами программные продукты, так и на технологическую платформу. Многие применяемые сегодня программные продукты используют принципиально устаревшую файл-серверную технологию для организации сетевой работы, реализация которой в школе приводит не только к сложности обслуживания, но и к нарушению информационной безопасности. Отсутствие доступных современных средств организации обмена образовательной информацией резко снижает значимость использования средств вычислительной техники в образовательном процессе.

Целью проекта является разработка современного комплекса программного обеспечения для создания информационного пространства общеобразовательного учреждения, свободного от лицензионных или иных отчислений, функционирующего на свободной программной платформе и использующего для информационного обмена открытые стандарты.

Для реализации цели проекта необходимо решение следующих задач.

- Разработка технологической платформы для функционирования информационной системы автоматизации деятельности общеобразовательного учреждения.
- Разработка модульной объектно-ориентированной динамической учебной среды, ориентированной на организацию взаимодействия между преподавателем и учениками, для организации дистанционных курсов и поддержки очного обучения;
- Разработка управляемой web-среды для организации взаимодействия участников учебного процесса.

ОПИСАНИЕ ДИСТРИБУТИВА

Школьный Сервер представляет собой современный комплекс программного обеспечения для создания информационного пространства общеобразовательного учреждения. В дистрибутив включены компоненты, автоматизирующие эту деятельность.

Управление всеми основными серверами и службами технологической платформы должно осуществляться с помощью единого веб-интерфейса.

СОСТАВ ДИСТРИБУТИВА

Интерфейс Школьного Сервера позволяет легко управлять:

- межсетевым экраном (с помощью упрощенного или расширенного интерфейса);
- почтовым сервером с поддержкой средств борьбы с вирусами и спамом;
- прокси-сервером (с поддержкой вывода статистики доступа);
- серверами файлов (организация доступа к каталогам, доступным по протоколу Samba и FTP) и печати (CUPS);
- обновлениями системы (включая настройку источников обновлений);
- подсистемой авторизации и персональных данных (LDAP-сервер)
- сетевыми интерфейсами;
- созданием локальных зеркал репозитория;
- резервным копированием;
- синхронизацией времени на сервере;
- выделением IP-адресов для локальной сети (DHCP-сервер);
- веб-ориентированными приложениями (Moodle и MediaWiki).

LDAP

Протокол LDAP (Lightweight Directory Access Protocol) предназначен для организации централизованного каталога пользователей с их личными данными, и авторизации пользователей с использованием данных из этого списка. Существует ряд реализаций этого протокола, в том числе реализация, разработанная под свободной лицензией – OpenLDAP. Подсистема состоит из клиентского модуля, реализующего удалённую авторизацию и доступ к каталогу пользователей, и LDAP-сервера, обеспечивающего хранение каталога, выдачу необходимых данных и авторизацию пользователей. Клиентский модуль для этой подсистемы можно установить на любой дистрибутив Пакета СПО, созданного в 2008 году. Благодаря использованию централизованной авторизации исчезает необходимость поддерживать на сервере отдельные списки пользователей для других сервисов (например, FTP или электронная почта).

MOODLE

В качестве модульной объектно-ориентированной динамической учебной среды предполагается использование среды Moodle.

Moodle (модульная объектно-ориентированная динамическая учебная среда) – это свободная система управления обучением, ориентированная, прежде всего, на организацию взаимодействия между преподавателем и учеником. Она подходит как для организации традиционных дистанционных курсов, так и для поддержки очного обучения.

Используя Moodle преподаватель может создавать курсы, наполняя их содержимым в виде текстов, вспомогательных файлов, презентаций, опросников и т.п. Для использования Moodle достаточно иметь любой web-браузер, что делает использование этой учебной среды удобным как для преподавателя, так и для обучаемых. По результатам выполнения учениками заданий, преподаватель может выставлять оценки и давать комментарии. Таким образом, Moodle является и центром создания учебного материала и обеспечения интерактивного взаимодействия между участниками учебного процесса.

Со Школьным Сервером поставляется несколько курсов для Moodle, которые развёрнуты и доступны для гостевого просмотра сразу после установки дистрибутива.

MEDIAWIKI

Для организации взаимодействия участников учебного процесса в качестве управляемой веб-среды в составе ТПОУ предлагается использовать программный продукт MediaWiki.

MediaWiki – это программный механизм для веб-сайтов, работающих по технологии «вики». Это один из самых мощных и распространённых вики-движков. Он встречается как на совсем маленьких сайтах, так и на огромных порталах, таких как Википедия (для работы которой MediaWiki и создавалась).

При установке дистрибутива Школьный Сервер автоматически устанавливается и Mediawiki. В результате она сразу готова к использованию. В отличие от Moodle, Mediawiki не комплектуется никакими дополнительными материалами.

АППАРАТНЫЕ ТРЕБОВАНИЯ ДИСТРИБУТИВА

В ходе внедрения ПСПО стало ясно, что около трети школ имеют компьютеры с объемом оперативной памяти менее 128 Мб, что примерно соответствует уровню десятилетней давности, поэтому аппаратным требованиям дистрибутива было уделено особое внимание.

Аппаратные требования дистрибутива:

- привод CD (необходим на время установки);
 - процессор совместимой с Pentium III архитектуры от 500 МГц (рекомендуется тактовая частота не ниже 1 ГГц);
 - объём оперативной памяти от 128 Мб (рекомендуется от 512 Мб).
- Если планируется использование Moodle и/или MediaWiki, то не менее 256 Мб;
- свободное место на жёстком диске от 4 Гб (рекомендуется от 10 Гб).

Необходимое место для хранения пользовательских данных, зеркал дистрибутивов и прочих потенциально объёмных данных может сильно

варьироваться, поэтому пользователю необходимо позаботиться о наличии достаточного резерва дополнительно к указанным величинам;

- сетевой адаптер 10/100 Мбит (рекомендуется 2 адаптера по 1 Гбит каждый);
- видеокарта необходима только во время установки.

ДИСТРИБУТИВ ALT LINUX CHILDREN: ОПЫТ И ПЕРСПЕКТИВЫ

Панюкова А.А.

mex3@altlinux.ru, ALT Linux, г. Москва

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

ИСТОРИЧЕСКАЯ СПРАВКА

Летом 2008 года была выпущена бета ALT Linux 4.0 Children в формате Live CD [1]. Live CD («Живой CD») – это возможность полноценной работы в Linux-окружении на любом компьютере без установки операционной системы на жёсткий диск. Единственное требуемое действие – загрузка компьютера с самого Live CD. Большая часть файлов системы находится на диске, значительно меньшая – в оперативной памяти, а сохранять результаты работы можно на флеш-носитель или дискету. Содержимое жёсткого диска в используемом варианте Live CD доступно в качестве отдельных каталогов для того, чтобы можно было задействовать при работе имеющийся банк изображений.

В первую очередь, диск был выпущен в виде раздаточного материала, который в качестве подарка получали дети по окончании занятий в оздоровительном лагере «Берёзка» и базе отдыха «Наука» при Южно-Уральском государственном университете [2].

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

ALT LINUX 4.1 CHILDREN СЕГОДНЯ

Программное наполнение ALT Linux 4.1 Children составлялось с расчётом не только на непосредственное проведение обучающего курса, но и на

последующую самостоятельную работу детей дома. В дополнение к программным продуктам, необходимым для самого курса, в дистрибутив входят инструменты для более требовательных или любознательных пользователей: диспетчер фотографий цифровой фотокамеры, редакторы фрактальной и ASCII графики, мощный аудиопроигрыватель, программы по астрономии и географии. По сравнению с предыдущей версией дистрибутива значительно обновлены графический редактор GIMP и редактор нелинейного видео Kdenlive.

Для дошкольников от 4-х лет и младших школьников предусмотрен развивающий центр GCompris, который содержит множество модулей, начиная с освоения клавиатуры и мыши и заканчивая логическими играми и заданиями, помогающими изучать окружающий мир. Для укрепления навыков набора текста добавлен клавиатурный тренажёр.

Следует отдельно отметить, что в состав дистрибутива не входят офисные программы, почтовые клиенты, средства манипуляции контактной информацией и прочие продукты, не отвечающие поставленным задачам. Практика преподавания показала, что эти программы создают значительный отвлекающий фон и засоряют информационное пространство при проведении занятий. По той же причине подключение к сети Интернет переведено в «ручной» режим и по умолчанию не используется. Для многопрофильной работы на компьютере рекомендуется использовать дистрибутивы ALT Linux 4.1 с другим спектром решаемых задач.

ПЕРСПЕКТИВЫ И ПУТИ РАЗВИТИЯ LIVE CD

По завершению работы создания курса занятий для детей [3], планируется релиз Live CD на бренче 5.0 с выпуском книги.

Рассматривается также вариант выпуска Live DVD с более подробным набором исходных материалов (в том числе, с видео), большим количеством игр и, конечно, возможностью настроить себе интернет. Рассматривается вопрос о наличии «профилей загрузки» на Live CD. В настоящее время по умолчанию загружается система, в которой примонтированы все носители информации на запись. Такая схема идеальна для того, чтобы была возможность сохранять созданные файлы на любые носители, чтобы не потерять их к моменту завершения работы. Так как не все родители уверены в том, что ребёнок не повредит информацию, имея доступ на запись ко всем жёстким дискам, поэтому рассматривается возможность создания нескольких вариантов загрузки:

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

УСТАНОВОЧНЫЙ ВАРИАНТ. ДОМАШНИЙ ДЛЯ ДЕТЕЙ

Есть некий класс детей, которые утверждают, что компьютер в доме используется только ими, что они отдают отчёт в своих действиях и т.п. Есть некий класс родителей, которые были бы не против, чтобы ALT Linux Children стоял на компьютере: не было бы необходимости постоянно грузиться с Live CD. Поскольку такие вопросы возникают довольно часто, логично предположить, что такой вариант дистрибутива тоже имеет место быть. С другой стороны, очевидно, что такое решение не может быть универсальным, если компьютер в доме используется не только и не столько детьми, сколько родителями. Ставить отдельный дистрибутив для ребёнка довольно неудобно: разница между загрузкой с Live CD и установленной второй системой (особенно, если один Linux уже стоит) с точки зрения удобства процесса работы не велика. Устанавливать детям Children, а родителям потом доставлять для себя все недостающие программы – тоже довольно неудобно. Доводить Desktop под пользователем ребёнка до состояния, близкого к Children – тоже можно, но по количеству приложенных усилий эквивалентно предыдущему варианту.

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

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

УСТАНОВОЧНЫЙ ВАРИАНТ. БОЕКОМПЛЕКТ ПРЕПОДАВАТЕЛЯ

Другой вариант «Устанавливаемого Children», который весьма полезен – это установочные диски для быстрого разворачивания класса на неизвестной территории. Как показала практика, изначальный вариант с разворачиванием всего с чистого листа может применяться в практически любых условиях. Здесь может возникнуть проблема, очень похожая на ту, что имела место в школьном проекте: железо везде разное, для разного железа оптимально подходят разные решения. Тем не менее, одно обстоятельство везде примерно одинаково: есть некий класс с примерно равными по характеристикам компьютерами; возможно, есть какая-то сеть (или её нет, но её можно организовать). В этом случае довольно универсальный вариант – установка системы на все компьютеры, при этом при наличии сети (или

возможности её сделать) один компьютер выделяется под «сервер». Под «сервером» в данном случае понимается некий компьютер, не отличимый от других. Это может быть один из класса (возможно, самый мощный), за которым будет работать ученик (или только преподаватель), и который будет отличаться от всех остальных только тем, что на нем будет работать некоторое количество служб, а также будут храниться все файлы, созданные учениками (для того, чтобы обеспечить доступ к этим файлам с разных компьютеров).

В качестве комплекта для установки предполагается два CD или один DVD, с которых будет ставиться и «сервер», и клиентские системы.

Например, сначала устанавливается система на «сервер». Далее, когда становится известен IP или хостнейм сервера, системы устанавливаются на остальные компьютеры, и в этот момент прописываются все необходимые параметры с учетом установленного ранее сервера (при этом нет необходимости в дальнейшей настройке). В результате после установки с минимумом действий будет получаться класс, в котором:

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

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

ЗАКЛЮЧЕНИЕ

Представленный вариант дистрибутива в формате Live CD, основанный на ветке ALT Linux 4.1, не является окончательным вариантом. Доработке подлежит и сам курс, и техническая база (переход на ALT Linux 5.0). С каждым годом в процессе проведения обучения появляются новые идеи, новые приемы и задания, которые находят отражения в новых версиях.

ЛИТЕРАТУРА

1. Панюкова, А. А. Создание обучающего курса для детей на базе Linux / А.А. Панюкова // Третья конференция «Свободное программное

обеспечение в высшей школе», Переславль, 2–3 февраля 2008 г. – М., ALT Linux: 2008. – С. 61–63.

2. Panyukova, A. Organization and methodics for realization of computer graphics studying using free software / A. Panyukova, M. Yakshin, T. Panyukova // Proceedings of the 10th International Workshop on Computer Science and Information Technologies, Antalya, Turkey, September 15–17, 2008: Сб. науч. тр.– Уфа, Редакционно-издательский комплекс УГАТУ: 2008. – Т. 1. – С. 238–241.

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

JTINKER – СРЕДА ВИЗУАЛИЗАЦИИ МЫСЛИТЕЛЬНЫХ ПРОЦЕССОВ В ТЕОРИИ ОРГАНИЗАЦИЙ

Аппель И.В.

ivan.appel@gmail.com

аспирант, кафедра ЭММиС, ЮУрГУ, г. Челябинск

В настоящее время в бизнес-среде большую популярность приобретает методология анализа и оптимизации бизнес-процессов, известная как «теория системных ограничений Гольдратта». Ключевым разделом этой методологии является построение логических диаграмм взаимосвязей процессов в организации, «мыслительных процессов» в терминологии теории системных ограничений. Данная работа основана на опыте разработки средства визуализации и автоматизации построения логических диаграмм, выполняемой в рамках проекта jThinker.

ВВЕДЕНИЕ

Теория системных ограничений – это популярная в настоящее время методология анализа управленческих решений [1, 2]. Одним из основных разделов теории ограничений являются «мыслительные процессы» – формализованные подходы к идентификации системных ограничений в деятельности предприятия, установления их причин, выработке подходов к их устранению и планированию мероприятий по реализации системных улучшений [1, 3]. Мыслительные процессы теории ограничений включают в себя построение логических диаграмм взаимосвязей системы, которые представляют собой графы.

Настоящая работа основана на опыте разработки средства визуального построения диаграмм мыслительных процессов, выполняемой автором в

рамках проекта jThinker [4]. Программное средство разрабатывается по модели «open source» и доступно для свободного скачивания с веб-сайта проекта. На момент написания статьи последней версией является jThinker 0.3.3, загружено более четырех сотен копий программы, автором поддерживается постоянная связь с рядом практиков теории системных ограничений по всему миру, активно предоставляющих рекомендации по дальнейшему развитию программного средства.

ПОСТАНОВКА ЗАДАЧИ

Диаграммы мыслительных процессов, такие как «дерево текущей реальности», «диаграмма разрешения конфликтов», «дерево плана перехода» и все прочие являются по сути графовыми моделями, базовыми объектами которых являются вершина, ребро «вершина-вершина» и ребро «вершина-ребро» (рис. 1).

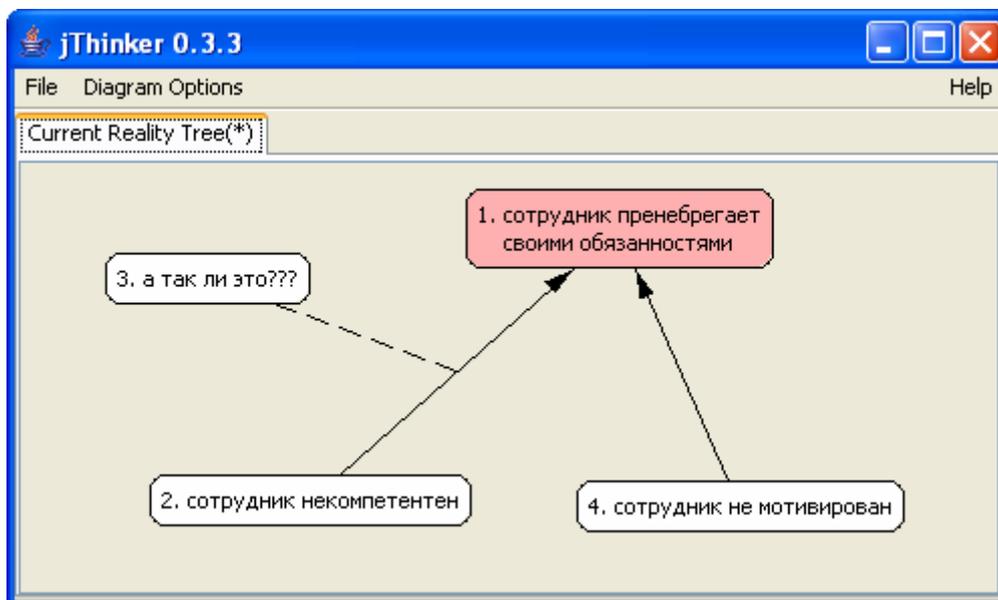


Рис. 1. Пример изображения дерева текущей реальности

В процессе разработки средства и последующих консультаций с пользователями был поставлен целый круг разнообразных подзадач, которые можно разнести по следующим категориям.

1. Базовый функционал для отрисовки и диалогового ввода диаграмм, а также для сохранения/загрузки файлов с диаграммами.

2. Разнообразные дополнения, такие как раскраска вершин, их группировка и нумерация.

3. Экспорт построенных диаграмм в графические файлы JPEG и PNG;

4. «Мастера подсказок» для построения диаграмм.

5. Функционал анализа и оптимизации. В частности, для улучшения визуального представления диаграммы интегрируется имеющаяся у автора кодовая база решения задачи Штейнера [5].

ОБЩИЙ ОБЗОР АРХИТЕКТУРЫ РЕАЛИЗАЦИИ

Для реализации такого широкого круга задач необходима программная платформа работы с графовыми моделями, которая должна обеспечивать:

- максимально простую для пользователя работу с диаграммой;
- точное графическое соответствие компонентов стандартам оформления, общепринятым в теории ограничений;
- возможность подключения «мастеров»;
- двустороннее отображение модели диаграммы в модель графа;
- возможность ассоциации произвольных данных с вершинами и ребрами и их графическое отображение.

В силу достаточно жестких требований к удобству использования, соответствия стандартам и расширяемости, было принято решение не использовать в качестве основы продукта сторонние решения, а разработать такую платформу самостоятельно.

В основу положена классическая для такого рода разработок архитектура «модель-представление-контроллер». Разработка программного средства ведется на Java с использованием Swing [6], что оказало определенное влияние на архитектуру.

Приложение состоит из трех нескольких компонент:

- **графическая.** Осуществляет отрисовку компонентов и взаимодействие с пользователем;
- **структурная.** Организует группировку и нумерацию вершин в диаграмме, выполняет преобразования данных для оптимизационной компоненты;
- **модельная.** Хранит всю информацию о диаграмме, сохраняет ее в XML-файл и восстанавливает из файла;
- **оптимизационная.** Реализует некоторые алгоритмы графовой оптимизации для улучшения визуального представления диаграммы.

Графическая компонента построена из следующих основных классов:

- базовые графические компоненты «вершина с текстом», «ребро между вершинами», «ребро между вершиной и ребром», «группа вершин»;
- фабрика обрамлений для вершин (рис. 2);
- редактор содержимого вершины (диалоговое окно).

Модельная компонента представляет собой группу классов, выполняющих роли моделей вершины, ребра, группы вершин и диаграммы в целом, реализующих отображение данных в XML документ.

Оптимизационная компонента предоставляет универсальный интерфейс представления данных графа моделей (реализованный в структурной компоненте) и реализует алгоритмы преобразования графа. В настоящий момент реализованы алгоритмы простого подавления пересечения вершин и, основанный на алгоритме для задачи Штейнера, оптимизатор диаграммы.

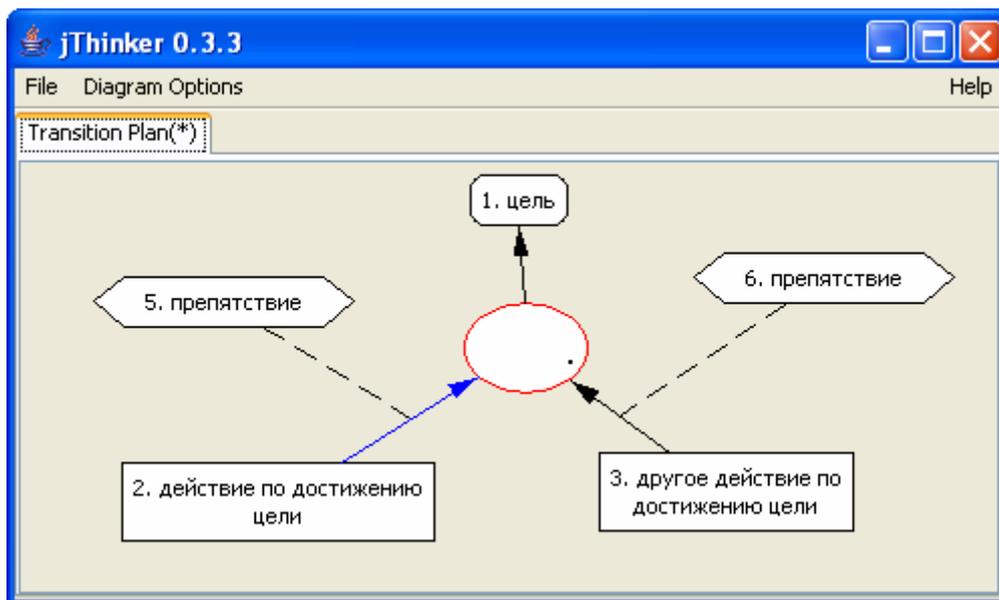


Рис. 2. Пример различных форм вершин в плане перехода

Структурная компонента состоит из набора вспомогательных классов:

- индекса вершин в диаграмме;
- хранилище данных о группировке вершин
- отображение структуры диаграммы в пригодный для оптимизации

вид.

ВНЕДРЕНИЕ. РЕЗУЛЬТАТЫ И ПЕРСПЕКТИВЫ

На настоящий момент основная часть просьб и рекомендаций, поступающих автору от пользователей, относится не к добавлению новой функциональности, а к упрощению использования уже имеющейся. Поэтому можно сказать, что приложение в своём развитии уже преодолело стадию тестового прототипа и пригодно для повседневного использования практиками теории ограничений.

Программное решение может использоваться не только опытными практиками, но и в процессе обучения методам теории ограничений на бизнес-тренингах, семинарах и спецкурсах, а также для самообразования.

Тем не менее, пользовательский интерфейс приложения на данный момент требует улучшений, что и планируется сделать в нескольких ближайших версиях.

Также планируется добавление «мастеров подсказок», которые будут упрощать построение диаграмм посредством использования формальных подходов, рассмотренных в [1, 3]. Прежде всего, эти «мастера» будут предназначены для тех, кто использует приложение в образовательных целях.

ЛИТЕРАТУРА

1. Goldratt, E. M. Theory of Constraints / Eliyahu M. Goldratt. – New York, North River Press, 1999.
2. Голдратт, Э. Цель: процесс непрерывного улучшения / Э. Голдратт, Дж. Кокс. – М.: Логос, 2005.
3. Детмер, У. Теория ограничений Гольдратта: Системный подход к непрерывному совершенствованию / У. Детмер. – М.: Альпина Бизнес Букс, 2008.
4. Веб-сайт проекта: <http://code.google.com/p/jthinker>.
5. Аппель, И. В. Разработка функционального элемента оптимального синтеза для САПР сенсорных сетей / И. В. Аппель // Материалы XLV Международной научной студенческой конференции «Студент и научно-технический прогресс»: Труды / Новосибирский госуниверситет, Новосибирск, 2007. – С. 161–167.
6. Портянкин, И. А. Swing: Эффективные пользовательские интерфейсы / И. А. Портянкин. – СПб.: Питер, 2001.

РАЗВИТИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ТОЧНЫХ ДРОБНО-РАЦИОНАЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ПАРАЛЛЕЛЬНЫХ И РАСПРЕДЕЛЕННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Горбик В.В., Панюков А.В.

*thqx@mail.ru, аспирант,
кафедра ЭММиС, ЮУрГУ, г. Челябинск*

*a_panyukov@mail.ru,
д.ф.-м.н., профессор, кафедра ЭММиС, ЮУрГУ, г. Челябинск*

Исследуется проблема реализации точных дробно-рациональных вычислений в параллельных и распределенных вычислительных системах с использованием свободно распространяемой библиотеки точных вычислений GNU. Проблема производительности решается созданием масштабируемых параллельных алгоритмов, ориентированных на многопроцессорную среду с интерфейсом MPI. Предложена адаптация библиотеки к среде с интерфейсом MPI.

ВВЕДЕНИЕ

Ориентация на применение многопроцессорных вычислительных систем в составе персональных компьютеров или рабочих станций (параллельные вычисления) и на применение сетевых технологий (распределенные вычисления) требует разработки новых параллельных методов их решения. Они должны быть лишены недостатков «традиционных» методов таких, например, как последовательный характер вычислений. Анализ способов распараллеливания показывает эффективность распараллеливания

«по информации». Поэтому, весьма перспективной становится SPMD-технология программирования (Single Program – Multiple Data). При этой технологии вычислительный процесс строится на основе единственной программы, запускаемой на всех процессорах вычислительной системы или на многих станциях локальной сети. Копии программы могут выполняться по разным ветвям алгоритма, обрабатывая подмножества данных. Неизбежна синхронизация во времени и при обработке общих данных. Данная идеология используется в стандарте MPI (Message Passing Interface) [1–3]. Такая технология параллельного программирования и обусловила разработку соответствующих методов. В то же время не отрицаются известные традиционные методы, сокращающие общее число операций и исключая перебор. Иной методологический подход открывает дорогу к решению задач большой размерности и эффективной параллельной работе многих процессоров.

Целью исследования является развитие программного обеспечения безошибочных дробно-рациональных вычислений для параллельных и распределенных вычислительных систем. Предполагается использование классов безошибочных дробно-рациональных вычислений `mpq_class` из библиотеки GNU MP [5]. Важным аспектом при этом является возможность и эффективность адаптации данных классов к многопроцессорной среде. Интерфейс MPI уже на протяжении длительного времени является негласным стандартом при построении распределенных вычислительных систем. В данной работе рассмотрен способ интеграции классов точных вычислений к MPI на основе сериализации объектов. Для целей безошибочных вычислений была выбрана библиотека GMP в виду того, что она является открытой разработкой, входит в дистрибутивы GNU/Linux и имеет хорошую производительность.

1. БИБЛИОТЕКА GMP

Пакет `gmp` содержит открытую библиотеку GNU MP для точных арифметических вычислений: операций над целыми числами со знаком, рациональными числами и числами с фиксированной точкой [5]. GNU MP разработана для быстрой работы, как для больших, так и малых операндов. Она работает быстро, поскольку использует целые слова как базовый тип, применяет быстрые алгоритмы, имеет оптимизированный ассемблерный код под многие типы процессоров и совмещает скорость с простотой и элегантностью выполнения операций. С помощью класса `mpq_class` реализованы точные вычисления с дробями, `mpq_class` построен на основе структур и функций библиотеки C, числителем и знаменателем являются структуры `mpz_struct`, которые содержат:

- размер выделенной области памяти;
- размер занятой области памяти;

- указатель на массив представляющий число.

Фрагменты листинга с объявлением `mpq_class` представлены на рис. 1. Как следует из листинга, `mpq_class` представляет собой обертку для структур и функций работы с дробями `gmp`.

2. АДАПТАЦИЯ MPQ_CLASS К СТАНДАРТУ MPI

Основная задача MPI – это производительный обмен данными между узлами. Однако интерфейс MPI адаптирован для передачи данных простых встроенных типов, массивов и пользовательских типов фиксированной длины, пересылка сложных пользовательских типов переменной длины является задачей более сложной.

| | |
|--|--|
| <pre>// FILE: gmp.h // Class definition for <i>rational</i> #ifdef __GMP_SHORT_LIMB typedef unsigned int mp_limb_t; #else #ifdef _LONG_LONG_LIMB typedef unsigned long long int mp_limb_t; #else typedef unsigned long int mp_limb_t; #endif #endif typedef struct { int _mp_alloc; int _mp_size; mp_limb_t *_mp_d; } __mpz_struct; typedef struct { __mpz_struct _mp_num; __mpz_struct _mp_den; } __mpq_struct; typedef __mpq_struct mpq_t[1];</pre> <p>(a)</p> | <pre>// FILE: gmpxx.h //Interface of mpq_class template <> class __gmp_expr<mpq_t, mpq_t> { private: typedef mpq_t value_type; value_type mp; public: __gmp_expr(); __gmp_expr(const __gmp_expr &q); template <class T, class U> __gmp_expr(const __gmp_expr<T, U> &expr); __gmp_expr(signed char c); __gmp_expr(unsigned char c); __gmp_expr(signed int i); __gmp_expr(unsigned int i); __gmp_expr(signed short int s); __gmp_expr(unsigned short int s); __gmp_expr(signed long int l); __gmp_expr(unsigned long int l); __gmp_expr(float f); __gmp_expr(double d);</pre> <p>(b)</p> |
|--|--|

Рис. 1. Фрагменты листинга класса `mpq_class`

Для передачи пользовательских типов фиксированной длины в MPI предусмотрен специальный механизм описания. Объявив структуру или вектор согласно стандарту, программист получает новый тип `MPI_Datatype` и в последствии может передавать данный тип целиком стандартными средствами, не думая о размере и составе сложного типа. Кроме того, данное решение оптимизирует передачу, исключая при пересылке ненужные байты выравнивания.

В случае с `mpq_class` данный подход не сработает, в связи с его внутренней структурой: `mpq_class` содержит две структуры `__mpz_struct`, которые в свою очередь содержат длину массива и указатель на массив содержащий число. Поэтому объявить структуру `mpq_class` по стандарту MPI невозможно по двум причинам:

- 1) необходимо хранить указатель;
- 2) каждый объект `mpq_class` может иметь произвольную длину, которая может измениться при следующей математической операции, т.е. не представляет универсальную структуру.

Из вышесказанного следует, что стандартными средствами передачу типа `mpq_class` осуществить не удастся. Решить данную проблему можно следующими способами:

- 1) передача `mpq_class` в виде составных типов данных;
- 2) сериализация `mpq_class` в буфер с целью дальнейшей передачи.

Первый способ решения исключается сразу из-за его неоптимальности, т.к. четыре сетевых транзакции (две передачи типов `int` и две передачи массивов из `int` произвольной длины) обойдутся слишком дорого (пересылка является самой дорогой операцией). Второй вариант позволяет обойтись одной транзакцией и, поэтому, взят за основу модификации.

Фрагменты листинга `mpq_class` нуждающиеся в упаковке приведены на рис. 2. Структура представления числа содержат не только размер массива, в котором хранится число, но так же и размер выделенной области памяти.

```
typedef struct
{
    int _mp_alloc;
    int _mp_size;
    mp_limb_t *_mp_d;
} __mpz_struct;

typedef struct
{
    __mpz_struct _mp_num;
    __mpz_struct _mp_den;
} __mpq_struct;
```

Рис. 2. Структура `mpq_class`

При передаче поле `_mp_size` можно считать лишним, т.к. гранулярность выделения памяти известна, и нам ее передавать необязательно. Интерфейс класса `mpq_class` расширен методами `int serialize(void *buff, int sizeofbuff)` и `int deserialize(void *buff)`, которые осуществляют упаковку массива `buff` произвольной длины в единый буфер и распаковку из буфера (рис. 3). Передачу буфера можно осуществлять стандартными средствами, но при этом возникают определен-

ные трудности с определением необходимого размера буфера на стороне приемника. Данная проблема решается вызовом стандартной функции `MPI_Probe`, которая позволяет узнать размер пришедших данных. После этого можно выделить необходимый буфер, получить данные, и десериализовать объект или массив объектов `mpq_class`.

Для удобства реализованы так называемые «обертки» для функций передачи/приема MPI. В результате такого подхода использование классов точных вычислений в среде MPI «прозрачно» для программиста и не требует внесения многочисленных изменений в код программы, например, при замене вещественных типов данных в существующем алгоритме на точные.

```

// FILE: gmpxx.h
...
#ifdef GMP_MPI
#define ERR_SMALLBUFF -1
#endif

template <>
class __gmp_expr<mpq_t, mpq_t>
{
private:
    typedef mpq_t value_type;
    value_type mp;
public:
    unsigned long int get_prec() const { return mpf_get_default_prec(); }
}

    void canonicalize() { mpq_canonicalize(mp); }

#ifdef GMP_MPI
// MPI interface
void dummy() { printf("GMP_MPI interface.\n"); }
int object_size();
int serialize(void *buff, int sizeofbuff);
int deserialize(void *buff);
#endif
...
};
...
#ifdef GMP_MPI
char *serialize_array(mpq_class *r, int count, int &buffsize);
void deserialize_array(void *buff, mpq_class *r, int count);
#endif

```

Рис. 3. Функции сериализации вектора rational.

ЗАКЛЮЧЕНИЕ

В основе вычислений всех коммерческих программ лежат типы данных с плавающей точкой, поэтому они не могут гарантировать высокую точность решения. Исключением из ряда продуктов, считающих приближенно, являются две открытых реализации симплекс метода, использующих в

вычислениях библиотеку точных вычислений GNU MP, что неизбежно ведет к существенному увеличению времени счета. Однако они не используют преимущества параллельного программирования, методы которого, при умелом использовании, позволяют сократить время расчетов и, следовательно, расширить круг решаемых задач. Для проведения научных расчетов, а также исследования методов параллельного и распределенного программирования, на кафедре ЭММиС был собран кластер (на основе имеющихся вычислительных средств учебной лаборатории). Данный кластер основан на свободно распространяемой реализации стандарта MPI – MPICH, и позволяет использовать преимущества и простоту данной библиотеки в программных реализациях. Для возможности использования классов точных вычислений в многопроцессорной среде MPI, был использован описанный в статье способ адаптации на основе сериализации объектов. Основной задачей было построение программы Plinprex решения задачи линейного программирования, основанной на параллельном алгоритме и использующей вычисления без округлений. Разработанный распараллеленный алгоритм использует всего две широкополосные коммуникации на каждой итерации основного цикла программы, что приводит к высокой оценке качества распараллеленного алгоритма.

Проведенный эксперимент показал, что реализованный метод эффективен на задачах разной размерности и соотношением количества ограничений к количеству переменных. В результате проведенного эксперимента можно сделать выводы о высокой эффективности распараллеленного алгоритма симплекс метода. Насколько позволяют оценить имеющиеся вычислительные средства эффективность распараллеливания близка к 100% (на количестве узлов до 10). Однако общее время выполнения алгоритма достаточно велико, и чтобы расширить диапазон решаемых задач, требуется дополнительная доработка в сторону оптимизации вычислений на узлах.

ЛИТЕРАТУРА

1. Интернет-Университет Информационных Технологий. – <http://www.intuit.ru/department/se/parallprog/4/1.html>
2. MPI: A Message-Passing Interface Standard. – <http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html>
3. MPI-2: Extensions to the Message-Passing Interface. – <http://www.mpi-forum.org/docs/mpi-20-html/mpi2-report.html>
4. Marc Snir. MPI: The Complete Reference/ Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, Jack Dongarra// Massachusetts Institute of Technology, 1996.
5. GNU Multiple Precision Arithmetic Library. – <http://swox.com/gmp/>

О ПРИМЕНЕНИИ GNU OCTAVE В МОДЕЛИРОВАНИИ СИСТЕМНОЙ ДИНАМИКИ

Игнатович А.С.

alexander_i@list.ru

аспирант, кафедра ЭММиС, ЮУрГУ, г. Челябинск

В статье рассматривается создание программной среды для создания и запуска системно-динамических моделей на базе пакета GNU Octave.

ВВЕДЕНИЕ

Усложнение экономических, социальных и политических систем привело к необходимости построения их моделей с целью изучения последствий воздействия на них различных внутренних и внешних факторов. Появление первых компьютеров в середине XX века привело к появлению нового инструмента системного анализа – имитационного моделирования, позволяющего строить сложные модели, учитывающие множество факторов и их взаимосвязи, и, самое главное, изучать поведение таких систем во времени. В контексте данного исследования рассмотрены возможности моделирования экономических, социальных, экологических и прочих систем, но не моделирование «физического уровня», используемое при проектировании, скажем, систем управления или корпуса самолёта или корабля.

Картина спроса на рынке имитационного моделирования отличается от общемировой, где данный метод исследования и построения систем давно и активно используется. Если в 2003г. спрос на ИМ со стороны бизнеса и государства только робко намечался, а в 2005г происходило интенсивное знакомство потенциальных заказчиков с подобными технологиями, то сейчас ИМ уже обосновалось в арсенале средств прогноза, анализа, и оптимизации [1].

Серьезным препятствием развития ИМ в России является высокая цена программных пакетов имитационного моделирования, составляющая не менее \$1000 за одну лицензию. В то же самое время, существуют несколько функциональных пакетов свободного программного обеспечения, предназначенные для математических расчетов, такие как Octave, SciLab, которые могут быть взяты за основу создания среды имитационного моделирования – программного пакета, предназначенного для создания и запуска имитационных моделей. Учитывая все вышеприведенное, можно сделать вывод о том, что создание такой среды является актуальной задачей, как для сферы практического использования, так и для образования.

Математически, системно-динамические модели задаются системой дифференциальных уравнений. Несмотря на кажущуюся, быть может, чрезмерной, простоту концепции, а также ограничения её исходных поло-

жений, как опыт развития и практического использования подхода, так и серьезные исследования, такие как [2] показали эффективность применения системной динамики.

1. СУЩЕСТВУЮЩИЕ СРЕДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В КОНЦЕПЦИИ СИСТЕМНОЙ ДИНАМИКИ

Основные, наиболее часто используемые и функциональные программные продукты, позволяющие создавать модели в концепции системной динамики: Anylogic [3], iThink / Stella [4], VenSim [5], PowerSim [6] (см. таблицу).

Таблица

Среды имитационного моделирования

| Наименование | Краткая характеристика | Цена |
|---------------|---|----------------------|
| iThink/Stella | Стандарт де-факто имитационного моделирования в концепции системной динамики. Имеет удобный интерфейс создания моделей, поддерживает множество математических и статистических функций, позволяет производить анализ чувствительности, запуск отдельных секторов модели. Представление результатов в виде графиков, анимации, экспорт в MS Excel и т.д. | \$1 899 |
| AnyLogic | Признана одним из лучших пакетов имитационного моделирования, поддерживает моделирование во множестве концепций. Создание java-апплетов. | \$4 800 – \$6 720 |
| VenSim | Поддержка создания дискретно-событийных моделей, наличие инструментов анализа и оптимизации, API и SDK. | \$4 990 |
| PowerSim | Удобный графический интерфейс, API и SDK, защита моделей, интеграция с SAP ® SEM-BPS, экспорт/импорт данных в электронные таблицы, СУБД, оптимизация, анализ и управление рисками, сценарии выполнения моделей. | \$1 498 |

Высокая цена приведенных программных продуктов (см. таблицу) обуславливает актуальность создания среды моделирования.

3. ПАКЕТ GNU OCTAVE И ЕГО ВОЗМОЖНОСТИ

GNU Octave представляет собой язык высокого уровня, предназначенный в основном для численных расчётов, и является привлекательной альтернативой коммерческому пакету Matlab компании The MathWorks. В отличие от простого набора команд, предлагаемых gnuplot, Octave предоставляет богатый язык математического программирования.

Octave представляет собой удобный командный интерфейс для решения линейных и нелинейных математических задач, а также проведения

других арифметических экспериментов, используя язык, в большинстве случаев, совместимый с Matlab. Кроме того, Octave можно использовать для пакетной обработки. Язык Octave оперирует арифметикой вещественных и комплексных скаляров и матриц, имеет расширения для решения линейных алгебраических задач, нахождения корней системы нелинейных алгебраических уравнений, работы с полиномами, решения различных дифференциальных уравнений, интегрирования системы дифференциальных и дифференциально-алгебраических уравнений первого порядка, интегрирования функций на конечных и бесконечных интервалах.

Пакет Octave может работать в режиме сценариев, интерактивно или посредством привязки к языку C или C++. В пакете Octave реализован богатый язык, похожий на C и обладающий очень большой библиотекой математических функций, в том числе специализированных функций обработки сигналов и изображений, обработки звука и автоматического управления. В качестве внутреннего механизма вывода Octave используется gnuplot [7].

3. РАЗРАБОТКА СРЕДЫ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

Разрабатываемый программный продукт должен выполнять следующие функции:

- обеспечить построение и вывод системно-динамических моделей в графической нотации (в терминах уровней, потоков, дополнительных переменных и связей между ними);
- обеспечить представление системно-динамических моделей на разных «слоях» – слое «карты модели», на котором должно выводиться графическое представление системно-динамической модели. На данном слое должно обеспечиваться создание «когнитивной карты» модели. Также пользователю должна быть представлена возможность задания комментариев к системно-динамической модели. Второй слой представления модели – слой «Модель» также должен предусматривать возможность задания когнитивной карты модели, но основная его задача – непосредственно задавать уравнения переменных, входящих в системно-динамическую модель. Третий слой «Интерфейс» должен предоставлять пользователю совокупность инструментов, позволяющих быстро и эффективно настраивать переменные модели, а также выводить результаты моделирования в удобном и наглядном виде. Четвертый слой «Уравнения» предназначен для просмотра пользователем системно-динамической модели в виде системы уравнений, а так же, на первом этапе, редактирования порядка их расчета;
- добавлять и изменять существующие имитационные модели, производить их настройку;
- добавлять и изменять переменные системно-динамической модели, задавать при необходимости набор значений, которые будет принимать переменная в заданные моменты модельного времени;

- добавлять и изменять настройки инструментов интерфейса системно-динамической модели;
- проводить имитационные эксперименты с созданной моделью, выводить результаты в виде таблиц и графиков.

Пример модели Лоуренца, известный как «эффект бабочки» приведен на рисунке (рис. 1).

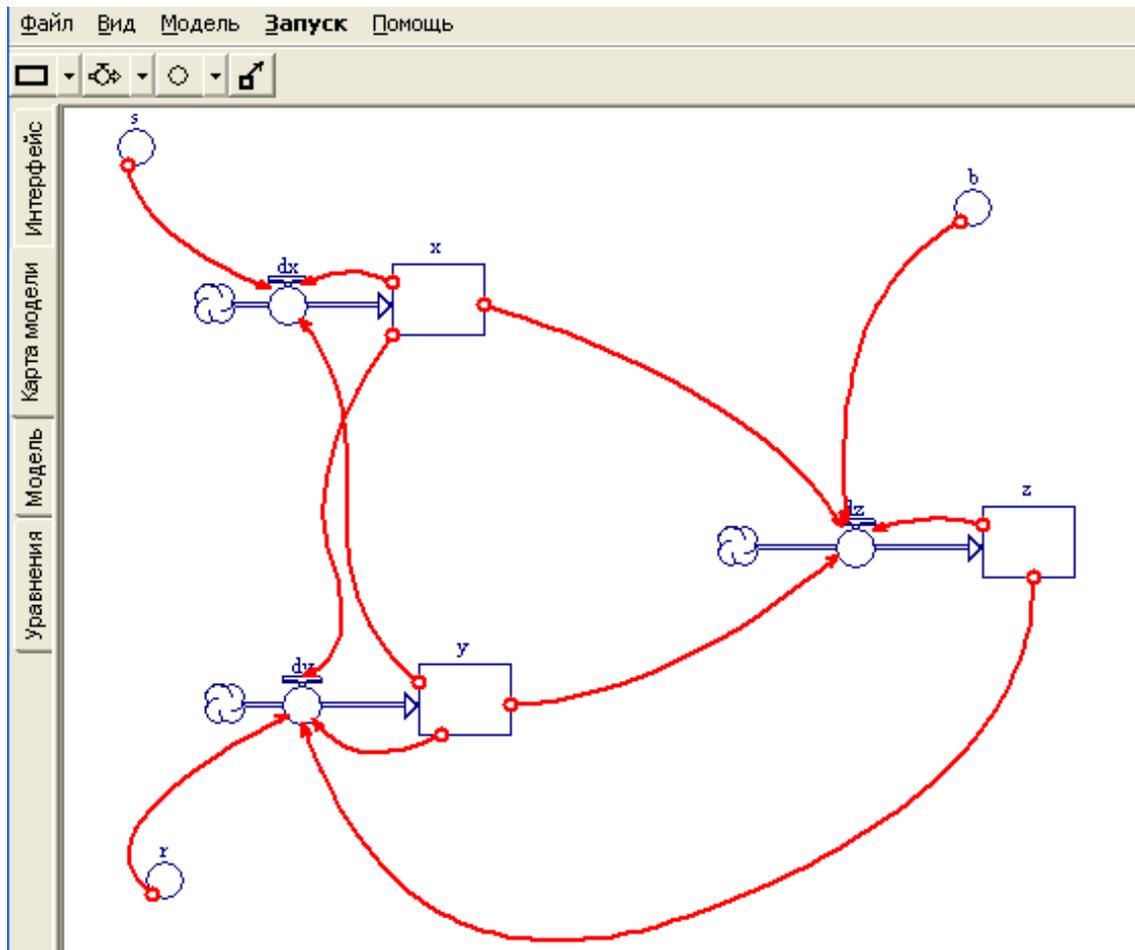


Рис. 1. Модель Лоуренца в разработанной среде имитационного моделирования

Для запуска имитационной модели необходимо сформировать программу для передачи ее среде GNU Octave. Приведенная модель «эффекта бабочки» на языке GNU Octave реализуется следующим образом:

```
# Аттрактор Лоренца
function y = lorenz( x, t )
# y(1) – x
# y(2) – y
# y(3) – z
b = 8/3;
r = 28;
```

```

s = 10;
y = [s * (x(2) - x(1));
x(1) * (r - x(3));
x(1) * x(2) - b * x(3)];
endfunction
# начальные условия
x0 = [3; 15; 1];
# границы и шаг моделирования
t = linspace (0, 25, 2500);
# непосредственное численное интегрирование
x = lode("lorenz", [3;15;1], t);
# вывод графика
plot(t, x);
print -deps foo.eps;
input("Press Enter to exit");

```

Результат выполнения программы приведен на рисунке (рис. 2.).

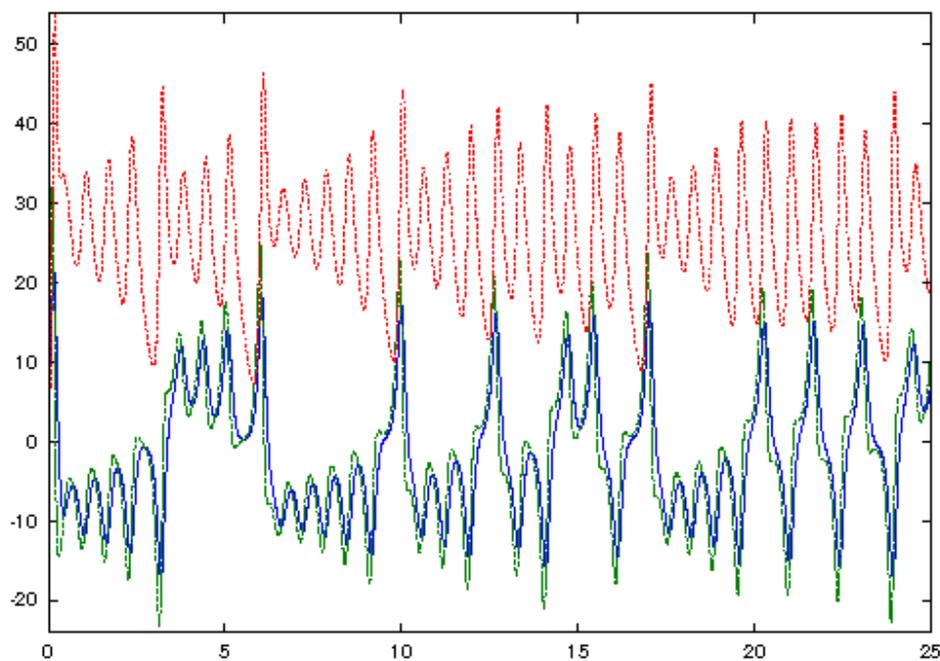


Рис. 2. Результаты расчета модели «эффект бабочки»

ЛИТЕРАТУРА

1. Борщев, А. От системной динамики и традиционного ИМ – к практическим агентным моделям: причины, технология, инструменты [Электронный ресурс] / А. Борщев, ООО «Экс Джей Текнолоджис» ХИ

Technologies www.xjtek.com и Санкт-Петербургский Государственный Политехнический Университет. – 2007.

2. Sterman, J. Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models / J. Sterman, H. Rahmandad // MIT Sloan Working Paper. – 2004. No. 4512-04.

3. Сайт AnyLogic. <http://www.xjtek.com/anylogic/>.

4. Сайт iSee Systems. <http://www.iseesystems.com/>.

5. Сайт Ventana Systems, inc. <http://www.vensim.com/>.

6. Сайт PowerSim Software. <http://www.powersim.no>.

7. Eaton, J. W. GNU Octave. A high-level interactive language for numerical computations / J.W. Eaton, D. Bateman, S. Hauberg // Manual Reference – 3rd Edition for Octave version 3.0.1 – 2007.

ТЕРМИНАЛЬНЫЕ РЕШЕНИЯ НА ПРЕДПРИЯТИЯХ

Рожков С.А.

ZserG@ZserG.ru,

инженер, НИИЦС ЮУрГУ, г. Челябинск

В настоящее время стал возрастать интерес к терминальным системам. Это связано, прежде всего, с тем, что накоплен обширный парк компьютерной техники, и эта техника уже не удовлетворяет запросам современных операционных систем, таких как Microsoft Windows Vista. Но для офисной работы ее вполне достаточно.

ВВЕДЕНИЕ

Терминалом называется устройство ввода и отображения информации. По сути – это бездисковая рабочая станция [1], загрузка которой осуществляется по сети. На сервер с клиента передаются нажатия клавиш и движения мыши. Обрато приходят снимки экрана. Все прикладные программы выполняются на терминальном сервере, но для пользователя все выглядит так, как если бы компьютер с такой же вычислительной мощностью, как сервер стоял у него на столе.

СХЕМА ПРИМЕНЕНИЯ

Сейчас большое распространение получили Windows терминалы на основе Windows CE, но у этих терминалов есть ряд минусов. Во-первых, программное обеспечение для таких терминалов является платным. Во-вторых, администратор ограничен в рамках одного протокола RDP, который позволяет подключаться только к Windows серверам, которые тоже стоят денег. В-третьих, администратор не имеет возможности изменить программную функциональность терминала, т.к. набор программного обеспечения задается изготовителем терминала. В-четвертых, невозможно передать данные на сервер с неподдерживаемых в протоколе RDP устройств. В-пятых, исходный код программного обеспечения закрыт а, сле-

довательно, в нем могут содержаться лазейки для разработчика. В-пятых для работы Windows терминала необходимо приобретать дорогостоящий Windows сервер и множество лицензий (на подключение к серверу, терминальные лицензии).

Все вышеперечисленные недостатки отсутствуют у терминалов на основе UNIX. Исходный код этих терминалов доступен, администратор имеет возможность добавить необходимые приложения и расширить функциональность. Также большим плюсом является возможность подключения не только к Windows, но также и к бесплатным UNIX серверам. Рассмотрим терминальный клиент WTPRO.

WTPRO – терминальный клиент для подключения к Windows и Unix серверам. Он не требует операционной системы и жесткого диска. В настоящее время эта разработка используется в более чем в двух сотнях организаций на территории бывшего СССР. Получены авторские свидетельства и свидетельства об отраслевой регистрации разработки [2,3,4,5]. Создан и поддерживается сайт <http://www.WTPRO.ru>. Разработка отражена в монографической литературе [6].

Настройка WTPRO производится через web-интерфейс. Возможно также использование Windows-утилиты для редактирования файла конфигурации.

Чаще всего WTPRO применяют для решения следующих задач.

- Создание мобильного офиса. В этом случае сервер находится на ноутбуке, а в качестве терминалов используются доступные в данный момент компьютеры. Развернуть подобный учебный класс или мобильный офис можно за считанные минуты.
- Организация учебных классов. Используется один мощный сервер, который обновляется, в случае необходимости. Терминальные клиенты, в силу нетребовательности к их ресурсам, обновлять не нужно. Преподаватель имеет возможность со своего рабочего места общаться с учеником.
- Организация презентаций. Не всегда есть возможность демонстрировать презентации на большом экране. Но если есть компьютерный класс, то можно вывести изображение на все компьютеры в классе.
- Развертывание операторных залов. Организация рабочих мест для большого числа пользователей, работающих с одним набором ПО.
- Организация работы бухгалтерии или отдела кадров и т.д.

АРХИТЕКТУРА

Минимальные системные требования для работы терминальной системы WTPRO:

- процессор: i486;
- оперативная память:
 - 6 Mb – без графического режима,

- 8 Mb – без перенаправления устройств,
- от 12 Mb – с перенаправлением устройств,
- от 16 Mb – с библиотекой Xwindow.
- видеокарта VGA, монитор;
- клавиатура, мышь;
- сетевая карта.

WTPRO поддерживает четыре терминальных протокола – два текстовых:

- telnet – небезопасный протокол, позволяет перехватить имя пользователя и пароль,

- ssh – защищенный аналог telnet;

и два графических:

- RDP – удаленный рабочий стол Windows,
- VNC – открытый протокол для доступа как к Windows, так и к UNIX-серверам.

Для подключения к Linux серверам можно использовать все эти протоколы.

Основной сложностью при разработке программного комплекса был компромисс для выполнения двух противоречивых требований:

- необходимо поддерживать большое количество оборудования.
- размер комплекса должен быть небольшим. Чтобы поместиться в оперативную память даже морально устаревших машин.

Для экономии оперативной памяти основная программа rdp, которая служит для подключения к терминальным серверам, была разбита на несколько модулей.

- rdp – содержит графическую подсистему, которая обеспечивает видеовывод, обработку операций мыши и клавиатуры.

- librdesktop.so – библиотека, основанная на коде проекта rdesktop, для доступа к функциям протокола rdp.

- libvnc.so – библиотека, основанная на открытых реализациях протокола vnc, предоставляет доступ к функциям протокола vnc.

- libscard.so – библиотека, основанная на патче Алексея Волкова к программе rdesktop для поддержки перенаправления на сервер смарт-карт.

- xrdr – аналог программы rdp, использующей библиотеку видеовывода xorg.

В WTPRO использует три вида драйверов для видеовывода.

Svgalib – облегченная библиотека, рекомендуется для использования на слабых машинах. Поддерживает не все видеокарты, а некоторые современные работают некорректно.

Framebuffer – данные видеодрайвера включены в ядро Linux. Для своей работы требуют большой объем оперативной памяти, рекомендуется использовать в том случае, если драйвера svgalib работают некорректно.

Xorg – наиболее ресурсоемкая библиотека, хорошо работает с современными видеокартами на машинах с объемом оперативной памяти от 16 Mb.

ЗАКЛЮЧЕНИЕ

Результатом данной работы является разработка, внедрение и поддержка терминального клиента WTPRO. Тонкий клиент имеет модульную архитектуру. Это позволяет запускать его как на устаревших, так и на современных компьютерах. Нами была произведена оптимизация кода и реализованы алгоритмы, для автоматического определения оборудования и загрузки его на терминал. Получившийся продукт может подключаться как к Windows, так и UNIX подобным серверам.

ЛИТЕРАТУРА

1. Таненбаум, Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум – СПб.: Питер, 2003. – 877 с.
2. Рожков, С. А. Терминальная система «ElinuxT» / С.А. Рожков // Свидетельство об официальной регистрации программы для ЭВМ № 2006611848. Зарегистрировано в Реестре программ 30.05.2006.
3. Рожков, С.А. Препроцессор "Elinux": свидетельство об отраслевой регистрации разработки № 5399 / С.А. Рожков. – № 50200501646 ; заявл. 08.11.2005.; опубл. 05.12.2005; Компьютерные учебные программы и инновации № 9(10). – 1 с.
4. Рожков, С.А. Терминальная система ElinuxT: свидетельство об отраслевой регистрации разработки № 5491 / С.А. Рожков. – № 50200501788; заявл. 08.11.2005.; опубл. 20.12.2005; Компьютерные учебные программы и инновации № 9(10). – 1 с.
5. Рожков, С.А. Защищенная терминальная система WTPRO: Свидетельство об отраслевой регистрации разработки № 12153/ С.А. Рожков. – Гос. координируемый центр информ. технологий. Отраслевой фонд алгоритмов и программ. М., 2009.
6. Стахнов, А. Linux-сервер в Windows-окружении./ А. Стаханов. – СПб.: БХВ, 2007, С. 656.

РАЗРАБОТКА ПРОГРАММЫ ДЛЯ РЕШЕНИЯ ЗАДАЧИ MAX TSP С ИСПОЛЬЗОВАНИЕМ СВОБОДНОЙ БИБЛИОТЕКИ GOBLIN

Тычинин С.А.

dettier@mail.ru,

преподаватель, кафедра ЭММиС, ЮУрГУ, г. Челябинск

В докладе приводится опыт интеграции и использования свободной библиотеки GOBLIN для решения вычислительных задач над графами. Библиотека позволила значительно увеличить размерность задачи коммивояжера на максимум, решаемой с помощью алгоритма дополнения подграфами

ВВЕДЕНИЕ

Автором доклада был разработан алгоритм приближенного решения задачи коммивояжера на максимум – алгоритм дополнения подграфами [1, 2]. Алгоритм обеспечивает достаточно высокую точность решения и в своей основе имеет подзадачи нахождения максимального взвешенного 2-фактора и максимального взвешенного паросочетания. Эффективное решение данных подзадач представляет собой непростую задачу. Для нахождения этих подграфов и повышения качества разработанного с целью проведения вычислительного эксперимента программного обеспечения было решено использовать свободно распространяемую библиотеку GOBLIN.

АЛГОРИТМ ДОПОЛНЕНИЯ ПОДГРАФАМИ И ЕГО РЕАЛИЗАЦИЯ

На входе: полный неориентированный граф $G = (V, E)$ с весовой функцией c . *На выходе:* гамильтонов цикл.

Шаг 1. Вычислить 2-фактор максимального веса $C = \{C_1, \dots, C_s\}$.

Шаг 2. Если $s = 1$ вернуть C_1 .

Шаг 3. Для каждого $i = 1, \dots, s$ положить

$$e_i = \{u_i, v_i\} = \arg \min_{e \in C_i} c(e), \quad C_i = C_i \setminus \{e_i\}.$$

Шаг 4. Положить $S = \bigcup_{i=1, \dots, s} \{u_i, v_i\}$, построить граф $H = (V(S), E - S)$, где

$V(S)$ – множество вершин, инцидентных ребрам множества S .

Шаг 5. Вычислить в графе H паросочетание максимального веса M .

Шаг 6. $C = C \cup M$. C_1, \dots, C_s – непересекающиеся циклы, образованные ребрами множества C . Перейти к шагу 2.

Конец алгоритма.

В процессе выполнения вычислительных экспериментов по определению практической точности алгоритмов встал вопрос об эффективной ре-

лизации вспомогательных алгоритмов нахождения подграфов – 2-фактора и паросочетания.

Приведем определения данных подграфов:

Подмножество $M \subset E$ ребер неориентированного графа $G = (V, E)$ называется *паросочетанием*, если каждая вершина $v \in V$ инцидентна ровно одному ребру в M .

Подмножество $M \subset E$ ребер неориентированного графа $G = (V, E)$ называется *2-фактором*, если каждая вершина $v \in V$ инцидентна ровно двум ребрам в M .

Для нахождения максимального паросочетания и максимального 2-фактора произвольного взвешенного графа существуют эффективные алгоритмы сложности $O(n^3)$ [3, 4]. Данные алгоритмы сложны в реализации, что являлось единственным препятствием для проведения вычислительного эксперимента на графах большой размерности. Были выполнены реализации алгоритма нахождения максимального паросочетания сложностью $O(n^3)$ и алгоритма нахождения максимального 2-фактора сложностью $O(n^6)$ [5]. Это позволило провести вычислительный эксперимент на полных графах размерностью до 160 вершин. Для проведения вычислительного эксперимента на графах большей размерности было решено воспользоваться библиотекой для работы с графами GOBLIN [6].

БИБЛИОТЕКА GOBLIN

GOBLIN это библиотека C++ классов, предоставляющая функции решения задач оптимизации на графах. Данная библиотека является программным обеспечением с открытым исходным кодом и распространяется под лицензией GNU LGPL.

Библиотека GOBLIN реализует самые эффективные на сегодняшний момент полиномиальные алгоритмы для решения следующих задач:

- нахождение кратчайшего пути в графах и в двудольных графах, имеющих отрицательные веса ребер;
- нахождение минимальных стягивающих деревьев и 1-деревьев;
- нахождение решений задач о назначении всех типов;
- нахождение паросочетаний, b-сочетаний и f-факторов;
- нахождение решений многих других задач.

Библиотека также включает средства для решения некоторых NP-сложных задач, таких как TSP (симметрическая задача коммивояжера), ATSP (асимметрическая задача коммивояжера), задача о раскраске.

ИНТЕГРАЦИЯ БИБЛИОТЕКИ И ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

Библиотека написана под компилятор GCC и во время ее интеграции в проект написанный под операционную систему Windows возникла необходимость в некоторой настройке используемого компилятора и внесении изменений в исходный код библиотеки.

Использование GOBLIN в реализации алгоритма дополнения подграфами для нахождения максимального паросочетания и максимального 2-фактора позволило значительно увеличить максимальный размер решаемой задачи. На компьютере с процессором Intel Core 2 Duo T7200 и объемом оперативной памяти 2Гб максимальный размер задачи решаемой не дольше 5 минут составил 570 вершин (для сравнения, предыдущая реализация позволяла найти решение задачи размерностью максимум 160). В дальнейшем это позволит провести более качественный вычислительный эксперимент и найти классы задач, на которых алгоритм дополнения подграфами наиболее эффективен.

ЛИТЕРАТУРА

1. Тычинин, С.А. Алгоритм дополнения подграфами для решения задачи MAX TSP / С.А. Тычинин // Информационный бюллетень ассоциации математического программирования N11: Конференция "Математическое программирование и приложения (тезисы докладов)". – Екатеринбург: УрО. РАН, 2007. – С. 217–218.
2. Тычинин, С.А. Модификация алгоритма Сердюкова с помощью алгоритма дополнения подграфами / С.А.Тычинин // Вестник ЮУрГУ Серия «Математика, физика, химия». – 2008. – С. 117–129.
3. Hartvigsen, D. Extensions of matching theory: PhD Thesis / D. Hartvigsen.– Pittsburg, PA: Carnegie Mellon Univ., 1984. – p. 148.
4. Gabow, H. An Efficient Implementation of Edmonds' Algorithm for Maximum Matching on Graphs / H. Gabow // Journal of the ACM, 1976. – №4. – P. 221–234.
5. Ловас, Л. Прикладные задачи теории графов. Теория паросочетаний в математике, физике, химии: Пер. с англ. / Л. Ловас, М. Пламмер. – М.: Мир, 1998. – 653 с.
6. GOBLIN Graph library / <http://goblin2.sourceforge.net>.

ПРАКТИКА ИСПОЛЬЗОВАНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УЧЕБНОМ ПРОЦЕССЕ

ИСПОЛЬЗОВАНИЕ СИСТЕМ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА ПРИ ПОДГОТОВКЕ СПЕЦИАЛИСТОВ-ДОКУМЕНТОВЕДОВ

Беспалова П.В.

chernookaya@mail.ru

*преподаватель, каф. информационной безопасности,
ЮУрГУ, г. Челябинск*

Вхождение России в современное информационное общество сопровождается коренными изменениями в различных сферах деятельности. В связи с этим определяется актуальная проблема, связанная с модернизацией профессионального образования. Современные реалии предъявляют новые требования для высшей школы – подготовка специалистов, обладающих профессиональной компетентностью. В условиях глобальной информатизации общества информационно-технологическая компетентность является одной из жизненно необходимых компетентностей для социальной и профессиональной деятельности человека.

Необходимость формирования ИТ-компетентности у будущих специалистов-документоведов обусловлена рядом факторов:

- наличие информационно-документационной базы в любой организации;
- активное использование и внедрение новых информационных технологий, в том числе систем автоматизации делопроизводства и электронного документооборота (СЭД), в различных сферах деятельности;
- формирование информационного пространства на уровне предприятия.

Современный работодатель отчетливо понимает, что переход от бумажных технологий управления к электронным жизненно необходим, поэтому ему необходим специалист, выполняющий функции системного технолога при использовании и внедрении систем автоматизации делопроизводства и электронного документооборота. Разработчики соответствующих программных продуктов в роли системного технолога видят именно специалиста по документационному обеспечению управления. Системный технолог настраивает систему электронного документооборота в соответствии с бумажным, описывает маршруты движения и виды доку-

ментов, обеспечивает работу остальных пользователей системы, настраивает систему на текущие изменения в структуре и документообороте организации. Во многом его функции переплетаются с функциями начальника службы документационного обеспечения управления организации.

Системным технологом должен быть работник предприятия, знающий принятый для данного предприятия порядок документооборота и обладающий навыками работы с компьютером. [1]. К сожалению, уровень ИТ-компетентности такого работника не всегда позволяет выполнять подобные функции, и руководителям предприятий часто приходится место системного технолога занимать ИТ-специалистом, хотя это в корне является неверным решением.

Основной упор на изучение систем электронного документооборота при подготовке по специальности «Документоведение и документационное обеспечение управления» делается на дисциплине «Компьютерные информационные технологии документационного обеспечения управления», которая входит в федеральный компонент. Соотношение лекционных занятий и практических часов по учебному плану 1:3. Во время изучения курса студенты получают знания по классификации СЭД, защите информации в таких системах, оценке эффективности внедрения систем автоматизации управления, по анализу рынка соответствующих программных продуктов. Однако при формировании информационно-технологической компетентности необходимы в большей степени практические занятия. Для этого необходимо внедрять учебные версии систем электронного документооборота. Причем, необходимо использовать максимально возможное количество таких программных продуктов. Организацию практических занятий целесообразно проводить с использованием активных форм обучения. Примером может быть деловая игра «Электронный офис», в которой студенческая группа является аналогом организации. При этом с заданной периодичностью необходимо менять роли в игре: от системных технологов до технических исполнителей.

Бесспорным лидером рынка систем автоматизации делопроизводства и электронного документооборота сегодня является система «ДЕЛО» компании «Электронные офисные системы», которая занимает около 50% российского рынка. Кроме того, именно эта компания первой начала активное сотрудничество с российскими вузами по программе «Электронный документооборот со студенческой скамьи». На сегодняшний день компании-разработчики СЭД по примеру «ЕОС» предоставляют учебные версии бесплатно, или по льготным условиям для вузов (системы «Летограф», «Евфрат», Lotus-Notes и т.д.). Как правило, функции системы в учебных версиях несколько ограничены, однако их бывает достаточно для практико-ориентированного обучения.

Еще одной особенностью сотрудничества с компаниями-разработчиками СЭД является их готовность бесплатно обучить препода-

вателей работе в системе. Для этого организуются семинары, деловые игры и т.д.

Исследования, проводимые на базе кафедры «Информационная безопасность» Южно-Уральского государственного университета, показали, что использование в учебном процессе систем электронного документооборота при подготовке документоведов существенно повышает уровень информационно-технологической компетентности. Кроме того, это является необходимым условием для качественной подготовки по специальности «Документоведение и документационное обеспечение управления», а выполнение этого условия ускорит процесс профессиональной адаптации выпускников и увеличит их конкурентоспособность на рынке труда.

ЛИТЕРАТУРА

1. Система автоматизации делопроизводства и электронного документооборота «ДЕЛО. Версия 8.8». Руководство пользователя. – Т. 1.

ВНЕДРЕНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В ПРОЦЕСС ОБУЧЕНИЯ ПРЕПОДАВАТЕЛЕЙ

Воронин И.В.

woronin05@yandex.ru,

*руководитель отдела информационных технологий,
ИПЛИТ РАН, Московская обл., г. Шатура*

В данной статье рассказывается об использовании СПО в системе РКЦ-ММЦ на примере ММЦ Шатурского района Московской области. Дается анализ проблем внедрения СПО, аргументируется необходимость финансирования и создания службы поддержки, даются рекомендации по поиску технически одаренных детей.

*Учатся взрослые
После работы.
С книжкой идут
На экзамен пилоты.
С толстым портфелем
Приходит невица,
Даже учитель
Не кончил учиться!*

А.Барто

Это стихотворение Агнии Барто актуально как никогда. На дворе 21 век, и современные дети в части изучения информационных технологий давно обогнали своих родителей и большинство преподавателей. Они с ходу осваивают мобильные телефоны, компьютеры, медиа-плееры и дру-

гие хай-тек новинки, на которые так щедр технический прогресс. Не все взрослые успевают за ними.

В наше время школьника невозможно заинтересовать учебой, используя только традиционные мел, доску и тряпку. Для выполнения процесса обучения приходится конкурировать с огромным количеством другой – яркой и привлекательной информации, имеется невероятное количество вариантов общения между современными школьниками с помощью интернет-технологий, новых программ, игр и т.д.

И для педагогов жизненно важно идти в ногу со временем. В образовании в последнее время (благодаря национальным проектам), наверное, как нигде, появилось так много нового оборудования, что без специальных курсов подготовки учителям обойтись очень сложно. Комплексный Процесс Модернизации Образования (КПМО) – дело не простое – это ответ на вызовы времени. Для того чтобы удачно модернизировать нашу промышленность – без освоения новых технологий в школе сейчас делать нечего. Также участие России в Болонском процессе подразумевает совершенно другое качество образования – единый стандарт, который позволит нашим выпускникам быть конкурентоспособными не только внутри страны, но и на европейском рынке труда. Именно поэтому преподавателю сейчас как ни когда необходимо сориентироваться в инновационной среде.

Кроме того, освоение новых технологий поможет учителю справиться с лавиной отчетности. Не секрет, что сегодня педагог все больше часов проводит за составлением бумаг о своей педагогической деятельности, на которую-то как раз времени у него остается все меньше и меньше.

В нашем Шатурском районе, Московской области, внедрение вычислительной техники в образовательный процесс средних школ и энерготехникума началось с 1988 года благодаря тому, что в городе Шатура тогда был основан ИПЛИТ РАН, в котором в данный момент работаю и я – руководителем отдела информационных технологий. Наши ученые – молодые специалисты помогали устанавливать в школах первые компьютеры, занимались программированием, когда-то даже вели уроки информатики.

В 1995 году педагог одной из наших городских школ Геннадий Краса получил звание заслуженного учителя именно за внедрение информационно-коммуникационных технологий (ИКТ) в преподавание физики и информатики. Его наработки и методические материалы, разработанные вместе с коллегами из ИПЛИТ РАН, широко известны.

Чтобы сегодняшним преподавателям было легче осваивать ИКТ, в некоторых субъектах федерации созданы региональные консультационные центры (РКЦ). Есть такой центр и в Московской области, он объединяет межшкольные методические центры (ММЦ), действующие в районах. Нисколько не умаляя необходимости работы традиционных методических кабинетов, хочу заметить, что система РКЦ-ММЦ появилась из-за того,

что в традиционных методических объединениях на местах зачастую нет необходимой базы для освоения современных технологий.

Межшкольный центр Шатурского района работает вот уже 3 года. За это время в нем прошли обучение и получили документы о повышении квалификации порядка 400 человек, это более трети от всего учителей в районе. Он действует на базе 2-й школы, и директор школы Ольга Шапкина совмещает обязанности и директора ММЦ. Она сама набирает преподавателей и определяет вместе с ними чему и как учить коллег. Волею судеб в ММЦ попал и я. Дело в том, что в этой школе учится мой сын, и когда привезли новый компьютерный класс, меня попросили подключить его. Конечно, я не мог отказать. Потом меня попросили прочитать лекцию... И с тех пор я преподаю учителям в ММЦ использование ИКТ, а также являюсь системным администратором. Плата за эту работу весьма символическая, но к работе я подхожу серьезно. Нарботаны методики, появился опыт дистанционного преподавания.

Учитывая, что свободное программное обеспечение (СПО) все шире используется в образовании и то, что я с Linux работаю на своей основной работе уже более уже 15 лет, было совершено логично развернуть в учебном классе пакет СПО от компании ALT Linux. Это произошло еще и потому, что при установке оборудования класса мы не смогли активировать ключ Майкрософта из-за отсутствия Интернет в тот момент, а также не смогли дозвониться в службу поддержки. Я предложил установить на компьютерах ПСПО, естественно, что возражений не последовало. Целью было – чтобы класс заработал, и эта цель с ПСПО была достигнута.

За прошедшие три года по курсу работы с офисным пакетом ПСПО уже прошло 6 групп, примерно по 15 человек в каждой. На занятиях мы создаем документы в Open Office, строим таблицы, создаем шаблоны для необходимых в повседневной деятельности преподавателям отчетов, учимся искать в Интернет картинки для презентаций, и конечно создаем сами презентации, (такая форма проведения уроков через презентации сейчас самая востребованная), а так же знакомимся с технологией wiki. Слушатели курсов получают навыки использования электронной почты и, как правило, к новогодним праздникам уже могут самостоятельно отправить открытку с поздравлениями в областное управление образования.

Конечно, некоторые педагоги даже эти несложные для современного школьника действия осваивают не сразу. Но успешное окончание курсов дает им возможность получать дополнительный разряд, а следовательно появляется финансовая заинтересованность. Для учителей предпенсионного и пенсионного возраста (а их 80–90%) – эти курсы позволяют продолжить работу в школе. Поэтому занятия пропускать не принято. Тем более что преподаватели проходят обучение не в Москве, куда из-за отдаленности района учителям сельских школ добираться проблематично, а в районном центре.

На курс отведено лишь 72 часа, тут и теория и практические занятия, но каждому отучившемуся мы вручаем кроме ПСПО и все его наработки на флеш-носителе и каждый может продолжить освоение пакета дома. В итоге учителя получают возможность повысить категорию, идти в ногу со временем и легко справиться с большим объемом отчетности. Важно, что по окончании курсов выдается удостоверение государственного образца.

Мультимедийное оборудование, которое сейчас поступает в школы, и в которое вложены огромные деньги, не должно простаивать, тем более что оно очень быстро становится морально устаревшим, и нужно осваивать все более новое. В начале внедрения ИКТ в школах района доходило до смешного – полученные компьютеры стояли за прочно закрытыми железными дверями, т.к. не было тех, кто мог бы на них работать сам и научить детей. Теперь такие проблемы уходят в прошлое, к тому же оборудование становится все более мобильным. Например «чудо-тележка» с большим плоским дисплеем, в которую «прячутся» ноутбуки, легко перемещается по школе старшеклассником, делая ненужными железные двери и решетки на окнах в отдельно взятом кабинете.

Тем не менее, обучение педагога работе с коммуникационным оборудованием не решает технические вопросы. Даже новое оборудование иногда выходит из строя, программы требуют переустановки и т.д. По закону Паркинсона это как правило происходит накануне важных презентаций, открытых уроков, ответственных мероприятий с участием высоких гостей... Кроме того, областное управление образования иногда само заводит педагогов в тупик. Например, требует отчета об установке фильтра контентной фильтрации. Естественно, что даже прошедший курс ИКТ преподаватель информатики не всегда понимает, что от него требуется информация о фильтре, не позволяющем детям выходить на сайты с сомнительным содержанием. Подобные вопросы возникают каждый день, поэтому остро необходимо созданию службы, которая бы оказывала школам техническую и информационную поддержку. В такой мобильной службе могут работать всего два человека, лишь бы они справлялись с обслуживанием всех учебных заведений на территории района. Конечно же, чтобы такая служба, даже если она и будет малозатратной, действовала эффективно, необходимо определить источник финансирования.

В школах США, например, такие затраты лежат на муниципалитетах. И престижность конкретного района проживания зависит от наличия в нем хорошей школы. Той, на которую тратятся деньги местных налогоплательщиков.

У американцев мы могли бы поучиться и использованию инновационных технологий в общеобразовательном процессе. Дети, начиная со школы, собирают и программируют роботов, и это – обязательная часть учебной программы. Проводятся международные олимпиады по робототехнике и высоким технологиям, в одной из которых, например, челябинские

школьники заняли почетное 2 место. Но для наших школьников – это единственный случай, а для американских – система.

Кстати, в той же Америке наработан уникальный опыт поиска технически одаренных детей. Для этого еще в 60-е годы 20-ого века была создана специальная программа. Говорят, это случилось после полета в космос первого человека – Юрия Гагарина. Тогда американцы решили, что-либо они выучат физику и математику, либо им придется учить русский язык. По этой программе ребенку-вундеркинду не запрещается делать чудовищные ошибки в диктантах, иметь низкий балл по гуманитарным наукам и даже числиться в неуспевающих, но они непременно должны хотеть и уметь разбираться в технике, например, собрать сенсорную сеть. У нас же научно-технической деятельностью с детьми практически никто не занимается. В нашем районе – всего один такой клуб («Маяк»), поддерживаемый за счет энтузиастов – но это не системный подход. Да, есть в области губернаторские премии хорошо успевающим ребятам, и есть возможность призерам олимпиад поступить по упрощенной системе в МГУ, но проблема в том, что технически увлеченный паренек может быть из неблагополучной семьи, пропускать занятия. А если из общей массы удастся найти и увлечь конструированием – моделированием на современном уровне одаренных ребят, то вложения окупятся с тройцей.

Нельзя сказать, что в России ничего не делается в этих направлениях. Например, Институт Новых Технологий, который курирует наш знаменитый ученый Евгений Велихов, ведет работу по формированию культуры использования ИКТ в образовании, например, создает варианты современных конструкторов – «Логомиры». Но это тоже пока не правило, а исключение.

Очевидно, что внедрение инновационных технологий в учебный процесс будет продолжаться и дальше, и оно невозможно без свободного программного обеспечения, хотя сейчас дискуссии на тему внедрения СПО в образование уже идут на спад – всем становится очевидна его необходимость. Возможно СПО будет востребовано более активно, как только появится прецедент, подобный произошедшему в Пермской области, и соответствующее указание поступят – из Министерства Образования РФ.

Поэтому обучение работе с СПО необходимо продолжать.

Кроме того, нужно развивать интерес школьников к научно-техническому творчеству в массовом порядке. Мотивировать преподавателей на поощрение и развитие такой активности у ребят.

Необходимо организовать поиск и поддержку детей, стремящихся к техническому творчеству и особенно тех, кто делает в нем заметные успехи. Проводить конкурсы, семинары и конференции – можно дистанционные, но с призовым фондом! Пусть небольшие, но вложения безусловно нужны.

И в завершение еще раз хотелось бы еще раз подчеркнуть, что без указания «сверху», без принятия организационного и политического решения на уровне правительства – Министерства Образования РФ, ожидать положительных сдвигов от проекта внедрения СПО в школах России и от общей информатизации образования – перевода на новые – инновационные технологии – рассчитывать не придется.

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В ПОДГОТОВКЕ БУДУЩИХ ПЕРЕВОДЧИКОВ

Волченкова К.Н.

jessicav@mail.ru

*старший преподаватель, кафедра английского языка,
ЮУрГУ, г. Челябинск*

В информационную эпоху невозможно себе представить профессиональную деятельность специалиста без использования компьютеров и компьютерных сетей. Это относится и к профессии переводчика. За последние 10–15 лет характер работы переводчика и требования к нему существенно изменились. В первую очередь изменения коснулись письменного перевода научно-технической, официальной и деловой документации. Сегодня, как правило, уже недостаточно просто перевести текст, пользуясь компьютером как пишущей машинкой. Заказчик ожидает от переводчика, что оформление готового документа будет соответствовать внешнему виду оригинала настолько точно, насколько это возможно, и при этом удовлетворять принятым в данной стране стандартам.

Индустрия перевода 21 века предъявляет к переводчику новые требования, поскольку объёмы перевода стали больше, сроки стали жёстче, и все чаще приходится работать в международных группах. Переводчику приходится быть менеджером проектов, компьютерщиком, документоведом, специалистом в издательском деле, лингвистом-исследователем.

От переводчика также требуется умение эффективно использовать ранее выполненные заказы, а работодатель, в свою очередь, рассчитывает на заметную экономию времени и средств при переводе повторяющихся или похожих фрагментов текста. Эти жесткие требования можно соблюсти лишь в том случае, если переводчик не только владеет родным и иностранным языком, но и уверенно ориентируется в современных компьютерных технологиях.

Сегодня от переводчика требуется владение современными компьютерными программами, направленными на оптимизацию его деятельности, а значит, готовность и способность осваивать эти программные продукты с

целью повышения привлекательности для работодателя и уменьшения затрат на «производственный процесс» письменного перевода.

Письменный перевод технической информации всегда был востребован. Но по данным статистики, переводчик без применения компьютерных технологий может перевести не более 2000 слов в день, что влечет за собой высокую стоимость перевода. Автоматизация существенно увеличивает производительность труда переводчика, позволяя снизить цену на конечный продукт.

В английской терминологии компьютерные технологии перевода текстов делятся на два больших класса – МТ (Machine Translation) и САТ (Computer-Aided Translation). Первый класс известен как автоматический или машинный перевод, то есть программы, с той или иной степенью успеха полностью заменяющие переводчика-человека. В отличие от них, системы САТ лишь автоматизируют и облегчают труд переводчика в различных его аспектах. В первую очередь это программы, реализующие концепцию памяти переводов, такие как Trados, DejaVu.

Идея сохранять для дальнейшего использования ранее переведенные тексты принадлежит японскому ученому Макото Нагао, который в 1982 году в одной из своих статей предложил новую концепцию машинного перевода. Она была основана на утверждении, что тексты должны переводиться по аналогии с текстами, ранее переведенными вручную. Наибольшее распространение данная идея получила в связи с развитием новейшей технологии перевода – систем класса Translation Memory (память переводов), которые все чаще в последнее время называют ТМ-инструментами.

Системы ТМ представляют собой интерактивный инструмент для накопления в базе данных пар эквивалентных сегментов текста на языке оригинала и перевода с возможностью их последующего поиска и редактирования. Переводчик в процессе работы сам формирует базу данных, и чем больше единиц она содержит, тем больше отдача от ее использования. Переводчик загружает переводимый текст в программную оболочку и, используя функции программы, выполняет поиск совпадающих сегментов в базе. Если совпадающего сегмента в базе не находится, переводчик сам переводит текущий сегмент текста и он добавляется в базу данных. Если полностью совпадающий или похожий сегмент найден, он подставляется в строку перевода. Система постоянно находится в режиме самообучения. Чем больше сегментов текста содержится в базе, тем более полезной она будет в работе. Наиболее эффективно использование данной технологии при переводе текста, содержащего большое количество повторяющихся частей.

Преимущества использования программ типа накопителя переводов несомненны, за одним исключением: как и любой качественно разработанный продукт они дорого стоят. Так программа «Trados», реализующая технологию памяти переводов стоит 895\$. По карману ли переводчику иметь

лицензионное ПО и рентабельно ли это? По карману ли вузу закупать многопользовательские версии таких программ для осуществления качественной подготовки будущих переводчиков, востребованных на рынке труда? Как-то так сложилось, что в России программа, либо «краденая», либо «лицензионная», за которую заплачены деньги и заключён договор с поставщиком. Соответственно, и переводчик как будто бы ограничен двумя альтернативами: использовать «взломанное» ПО или покупать оригинальные версии.

На самом деле, на этом альтернативы не заканчиваются. Существует и открытое ПО, причём «открытое» в данном случае не означает просто бесплатное. Эта концепция гораздо богаче. И, что немаловажно, она гораздо ближе научным, академическим кругам. Ведь для того, чтобы сохранить модель научного сотрудничества между разработчиками, необходимо обеспечить, чтобы исходные тексты программ, написанных разработчиками, оставались доступными для чтения и критики всему научному сообществу. Для этого было сформулировано понятие свободное программное обеспечение, в котором отразились принципы открытой разработки программ в научном сообществе, сложившемся в американских университетах в 1970-е годы. Основателем движения за свободное ПО является Ричард Столлман.

По его мнению, свободным может считаться ПО, если при его использовании соблюдаются следующие четыре свободы: программу можно использовать с любой целью («нулевая свобода»); можно изучать, как программа работает и адаптировать её для своих целей («первая свобода»); можно распространять копии программы («вторая свобода»); программу можно улучшать и публиковать свою улучшенную версию, с тем чтобы принести пользу всему сообществу («третья свобода»).

Письменный перевод, осуществляемый с помощью информационно-коммуникационных технологий процесс многоэтапный: переводчик должен уметь пользоваться почтовым клиентом при получении перевода от заказчика, уметь разархивировать большие по объёму файлы, сканировать необходимые документы, пользоваться электронными словарями и терминологическими базами данных при осуществлении процесса перевода, знать возможности программ локализации перевода и т.д. Нужно отметить, что практически для каждой программы, используемой переводчиком в работе, имеются свободно распространяемые аналоги.

В таблице представлены программы, которыми пользуются переводчики в профессиональной деятельности и аналоги свободного ПО.

| Назначение ПО | Проприетарное ПО | Свободный аналог |
|---------------------------------------|----------------------------|---|
| Электронная почта | TheBat, | Gmail, Mozilla Thunderbird |
| FTP-клиент | Cyte FTP | FileZilla FTP Client |
| Офисный пакет | Microsoft Office | OpenOffice.org |
| Электронный словарь | AbbyLingvo, Multilex | StarDict, (оболочка) Dicto (словарь) |
| Программы ТМ (Translation Memory) | Trados, SYSTRAN, Pragma | OmegaT |
| Программы МТ (Machine Translation) | PROMT, SYSTRAN, RETRANS | НЕТ АНАЛОГОВ |
| Программы верстки | Quark xPress | Scribus |
| Архиватор | WniZip, WinRar | 7-ZIP |
| Интернет браузер | InternetExplorer | Mozilla Firefox |

Для переводчика свободное ПО привлекательно по трём причинам: оно бесплатно, весьма надёжно, часто обновляется.

Пожалуй, две самые важные для переводчика свободные программы – офисный пакет OpenOffice.org и система памяти переводов OmegaT. Использовать их можно как под Windows, так и под GNU/Linux.

OpenOffice.org обладает практически всеми возможностями MS Office – включает текстовый процессор, электронные таблицы, векторный редактор, редактор формул, базу данных и т.д. В общем, для переводчика, пожалуй, более осмысленно выбирать именно его – как с точки зрения бесплатности, так и с точки зрения возможности экспорта документов в PDF.

OmegaT – это программа типа ТМ (Translation Memory). Программа является бесплатной, доступен исходный код программы, и можно свободно изменять и распространять как исходные, так и измененные копии программы при условии соблюдения лицензии GNU GPL. К основным возможностям OmegaT относятся: возможность запуска на любой поддерживаемой Java операционной системе; использование любого валидного ТМХ-файла для справочной базы переводов; гибкая сегментация по предложениям; поиск в проекте и в справочных базах памяти переводов; поиск в любой директории, содержащей поддерживаемые OmegaT файлы; предоставление нечетких совпадений; интеллектуальное ведение проектов; поддержка глоссариев; понятная документация; перевод на множество языков.

Несомненно, в некоторых областях возможности свободного ПО пока уступают коммерчески разработанным продуктам, но они позволяют осуществлять образовательный процесс не нарушая лицензионных прав компаний-разработчиков проприетарного ПО и являются незаменимыми помощниками в подготовке переводчиков при отсутствии возможности покупки коммерческого продукта.

ПРЕПОДАВАНИЕ КОМПЬЮТЕРНОЙ ГРАФИКИ В ШКОЛАХ С ИСПОЛЬЗОВАНИЕМ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Зильберман М.А.

school91-perm@mail.ru, учитель информатики,
МОУ «Средняя общеобразовательная школа № 91», г. Пермь

В статье представлено описание функциональных возможностей графических редакторов, входящих в дистрибутив СПО для школ от компании «Альт-Линукс», которые можно использовать для преподавания компьютерной графики на разных ступенях обучения в общеобразовательной школе. Для каждого редактора представлены ссылки на методические материалы, которыми может воспользоваться преподаватель для организации уроков по компьютерной графике.

ВВЕДЕНИЕ

Компьютерная графика – важная область информатики. Умение работать с различными графическими редакторами является важной частью информационной компетентности ученика. В среде ОС Windows до 2010 года учащимся школ доступны мощные графические редакторы Corel Draw, Adobe Photoshop. Для них написано много практикумов и руководств.

Однако под Линукс тоже написаны графические редакторы, пусть на данный момент уступающие их коммерческим аналогам, но всё-таки имеющие необходимый набор функций. Цель работы – описать функциональные возможности графических редакторов, входящих в школьный дистрибутив Альт-Линукс, с точки зрения их использования в преподавании компьютерной графики, а также дать ссылки на методические материалы по работе с ними.

1. ОСНОВЫ РАБОТЫ С ГРАФИЧЕСКИМИ РЕДАКТОРАМИ

В состав дистрибутива Альт-Линукс входят две программы, которые могут быть использованы для изучения основ работы с растровой и векторной графикой.

1.1. Tux Paint (Рисуй вместе с Tux!)

Растровый графический редактор **Tux Paint** идеально подходит для овладения базовыми принципами работы в растровом графическом редакторе. Он позволяет:

- создавать изображения с помощью различных кистей, линий, графических примитивов;
- применять для рисования «нестандартные» инструменты: кирпичные стенки, звёзды, радугу и др.;
- применять различные эффекты к нарисованным изображениям,

- создавать и редактировать текст,
- создавать изображения, используя готовые наборы штампов;
- создавать простую анимацию (чередование кадров);
- использовать готовые шаблоны для рисования и раскрашивания;
- создавать и сохранять готовые изображения, выводить их на печать.

Недостатки редактора:

- неудобно создавать графические примитивы;
- отсутствует функция выделения и копирования.

Редактор представляет большие возможности для творчества учеников любого возраста. Уроки можно проводить в виде ознакомления учащихся с отдельными способами рисования и создания изображений на их основе.

Учащимся старших классов (8–11 кл.) можно знакомиться с **Tux Paint** на элективных курсах по информатике.

Методические материалы:

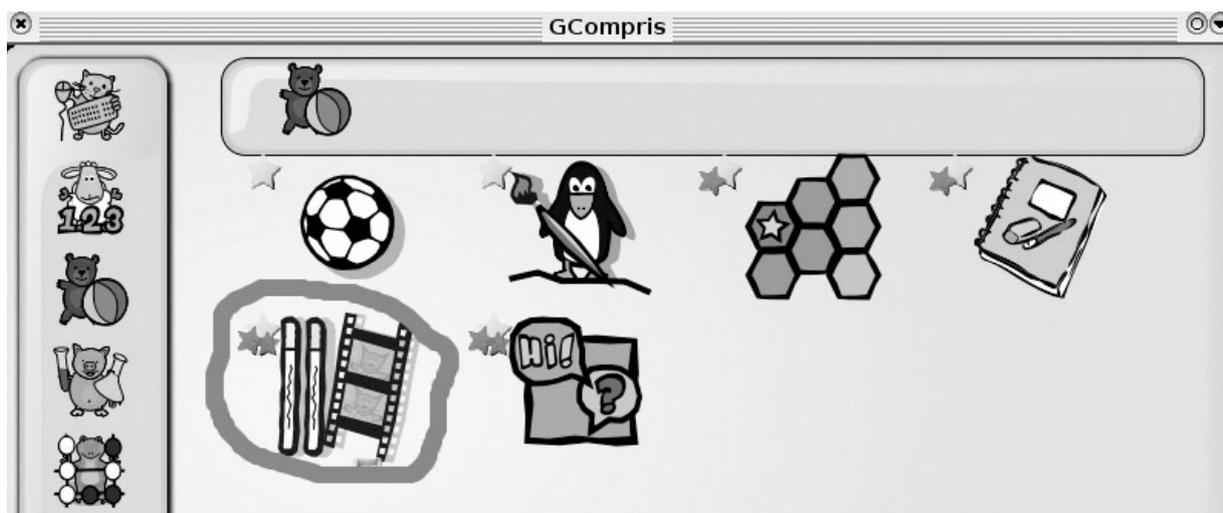
<http://freecode.pspo.perm.ru/307/index.html> – конспекты уроков и презентации;

<http://freecode.pspo.perm.ru/054/index.html> – конспекты уроков;

<http://freecode.pspo.perm.ru/339/index.html> – пособие по работе с редактором.

1.2. ВЕКТОРНЫЙ РЕДАКТОР В СОСТАВЕ GCOMPRIZ

В состав детского обучающе-развивающего комплекса программ **GCompris** (KDE – Игры – Детские игры – Образовательная программа для детей (Gcomriz)) входит достаточно простой в использовании векторный редактор (см. рисунок).



Графический редактор в составе GCompris (обведён рамкой)

Он позволяет:

- создавать изображения с помощью линий, графических примитивов и наборов готовых изображений;

- редактировать цвет и размер изображений;
- поворачивать и зеркально отражать объекты,
- перемещать изображения по слоям (перед/ после других объектов)
- создавать и редактировать текст,
- создавать простую анимацию путём создания и последовательного расположения кадров, каждый из которых можно впоследствии редактировать.

2. БАЗОВЫЙ УРОВЕНЬ

В состав дистрибутива Аль-Линукс входят 2 программы, которые позволяют овладеть базовыми навыками работы как в векторном, так и в растровом редакторах. Они полностью удовлетворяют требованиям стандартов основного общего образования.

2.1. ГРАФИЧЕСКИЙ РЕДАКТОР KOLOURPAINT

Растровый графический редактор **KolourPaint** похож по интерфейсу на известный графический редактор Microsoft Paint, а по возможностям превосходит его. Он позволяет:

- создавать изображения с использованием графических примитивов, карандаша, кистей, прямых линий, а также ломаных и кривых;
- использовать заливку и распылитель;
- добавлять текст, изменяя шрифт, кегль, начертание и цвет;
- стирать с помощью **ластика**, убрать и заменять цвета с помощью **очистки цвета**;
- выделять объекты с помощью прямоугольного, овального и произвольного выделения;
- изменять размер, масштабировать, отражать и поворачивать изображения;
- настраивать баланс изображения (яркость, контрастность, гамму, в том числе по каналам);
- применять эффекты: рельеф, вдавленный, инвертировать (всё или по каналам), резкость;
- сохранять результат в различных форматах, в том числе .png, .jpg и др.

Методические материалы:

<http://freecode.pspo.perm.ru/434/index.html> – пошаговые методические материалы для учителя и ученика.

2.2. ВЕКТОРНАЯ ГРАФИКА OPENOFFICE.ORG DRAW

Программа **OpenOffice.org Draw** предназначена для создания несложных векторных изображений и схожа по интерфейсу с встроенным в Microsoft Office Word векторным редактором, но превосходит его по функциональности. Он позволяет:

- создавать изображения, используя графические примитивы и другие готовые фигуры (фигурные стрелки, элементы рисования схем, выноски), а также объёмные фигуры;
- создавать и редактировать рисунки в кривых;
- добавлять и редактировать текст;
- применять к объектам различные виды заливок (однотонные, градиенты, текстуры, узоры), изменять параметры контура;
- применять к объектам различные виды взаимного расположения: выравнивания и расположения по слоям.

Методические материалы:

<http://freecode.pspo.perm.ru/090/index.html> – практикум с примерами работ.

3. ПРОФИЛЬНЫЙ УРОВЕНЬ

Знакомство с полнофункциональными векторным (**Inkscape**) и растровым (**GIMP**) редакторами возможно в рамках элективного курса «Компьютерная графика», а также в профильных группах по информатике.

3.1. РЕДАКТОР ВЕКТОРНОЙ ГРАФИКИ INKSCAPE

Векторный редактор **Inkscape** является полнофункциональным векторным редактором. Его можно использовать при преподавании компьютерной графики в качестве аналога **Corel Draw**. Он позволяет:

- создавать и редактировать рисунки, созданные с помощью графических примитивов;
- создавать и редактировать кривые;
- создавать различные заливки (однотонные, градиентные, текстурные) и контуры (изменять цвет, тип линий, толщину, соединение контуров). Возможность интерактивности заливок;
- создавать и редактировать текст;
- использовать функции выравнивания, распределения, изменения положения объектов относительно друг друга;
- выполнять логические операции с объектами (объединение, пересечение, исключение и др.);
- создавать объекты с использованием **Каллиграфического пера**;
- импортировать растровые объекты и экспортировать в растровые форматы.

Основные отличия от векторного редактора **Corel Draw**:

- отсутствие в **Inkscape** эффектов перетекания, экструзии (объёма), огибающей, искажения и менее развёрнутые настройки прозрачности;
- более сложный для освоения интерфейс, в частности, настройки цвета заливки и штриха, настроек градиентов, текстурных заливок.
- меньшие возможности работы с текстом.

Методические материалы:

<http://freecode.pspo.perm.ru/427/index.html> – наиболее полный практикум для учащихся.

<http://freecode.pspo.perm.ru/267/index.html> – практикум;

<http://inkscape.org/doc/index.php?lang=ru> – руководство от разработчиков.

3.2. РЕДАКТОР РАСТРОВОЙ ГРАФИКИ GIMP

Растровый редактор **GIMP** является полнофункциональным растровым редактором. Его можно использовать при преподавании компьютерной графики в качестве аналога **Adobe Photoshop**.

Он позволяет:

- создавать и редактировать растровые изображения;
- использовать различные выделения, маски;
- работать со слоями и каналами;
- использовать различные фильтры, эффекты слоя;
- рисовать с помощью кистей, создавать градиенты, использовать различные настройки наложения слоёв;
- редактировать изображения, используя штамп, осветление-затемнения;
- широкие возможности редактирования тоно- и цветокоррекции.

Есть существенные различия в интерфейсе с программой **Adobe Photoshop**, однако в школьном курсе компьютерной графики данный редактор можно использовать для выполнения большинства работ курса.

Методические материалы:

<http://freecode.pspo.perm.ru/425/index.html> – подробный практикум;

<http://freecode.pspo.perm.ru/374/index.html> – лекции по растровой графике: GIMP и Adobe Photoshop;

<http://freecode.pspo.perm.ru/423/index.html> – практикум;

<http://docs.gimp.org/ru/> – «Руководство пользователя» от разработчиков.

ПРИМЕНЕНИЕ СИСТЕМЫ GAP В НАУЧНЫХ ИССЛЕДОВАНИЯХ

Зюляркина Н.Д., Гаврилова Т.П.

*к.ф.-м.н., доцент кафедры ЦРТС, ЮУрГУ, г. Челябинск
старший преподаватель кафедры общей математики
ЮУрГУ, г. Челябинск*

Пакет GAP является системой компьютерной алгебры, задуманной как инструмент вычислительной теории групп, и впоследствии распространившейся на такие разделы алгебры, как теория графов, теория кодирования, кристаллографические группы. Система работает с многочленами от многих переменных, рациональными функциями, векторами и матрицами. С помощью GAP можно производить вычисления с гигантскими целыми и рациональными числами, допустимые значения которых ограничены только объемом доступной памяти.

Система GAP (группы, алгоритмы, программирование) была создана группой ученых из Германии специально для работы с различными алгебраическими и комбинаторными объектами: группами, кольцами, полями, векторными пространствами, графами, конечными автоматами и др. Эта система находит широкое применение в научных исследованиях, так как обладает рядом неоспоримых достоинств:

- наличие богатой библиотеки данных о широком круге известных объектов (групп, колец, графов и т.д.);
- большие возможности в области вычислений (работа с большими числами, объемными массивами, а также высокая скорость вычислений);
- простой язык программирования (например, при составлении программ не требуется описывать вид переменной, которая может в процессе вычислений принимать значения различных типов);
- бесплатное распространение (немаловажно в период кризиса!);
- высокая надежность (и это несмотря на бесплатность!);
- хорошая совместимость практически с любым программным обеспечением (хотя данная система создавалась для работы под LINUX и UNIX, она хорошо зарекомендовала себя при работе под WINDOWS и DOS).

В качестве основных аспектов использования GAP в научных исследованиях можно указать следующие.

- Применение GAP непосредственно для вычислений. Следует отметить, что в данной системе реализованы рациональные числа, некоторые расширения поля рациональных чисел, конечные поля, но нет действительных чисел и обычных элементарных функций типа $\sin(x)$, $\ln(x)$ и т.п. Очень эффективна работа с матрицами, коэффициенты которых содержатся в описанных выше множествах, а также с перестановками. Хорошие результаты могут быть получены и в случае комбинаторных вычислений.

Заметим, что вычисления с помощью GAP имеют точный, а не приближенный ответ.

- Использование GAP для выдвижения и проверки научных гипотез. Быстрота вычислений позволяет за короткий промежуток времени накопить большой статистический материал и произвести его анализ.

- Получение с помощью GAP информации об известных алгебраических и комбинаторных объектах. В качестве примера здесь можно привести таблицы характеров для достаточно большого количества конечных простых групп и таблицы пересечений для подгрупп конечных групп.

Таблицы характеров позволяют во многом охарактеризовать рассматриваемые группы и имеют широкое применение как в теории групп, так и в других областях математики и физики (например, при изучении групповых колец в теории колец и при изучении физических свойств кристаллических решеток в кристаллографии). Таблицы пересечений дают представление о подгрупповой структуре рассматриваемой группы.

Для построения алгебраических и комбинаторных объектов с заданными свойствами. Это и графы с различными условиями регулярности, и линейные коды над конечными полями с заданной надежностью и еще многое другое.

Хотя система GAP создавалась в основном для применения в алгебре и дискретной математике, она может быть применена в других областях современной науки. Это и теория вероятностей, физика, экономика и др.

В заключении отметим, что результаты, полученные с помощью GAP, не являются истиной в последней инстанции. Известны случаи получения недостоверной информации при использовании данной системы. Причины таких ошибок могут быть связаны не только с внутренними проблемами, но и с внешними факторами (например, с дефектами памяти или нехваткой ресурсов компьютера). Поэтому лишняя (лучше независимая) проверка не помешает.

ЛИТЕРАТУРА

1. Коновалов, А.Б. Система компьютерной алгебры GAP 4.4 / А.Б. Коновалов. – Челябинск: Изд-во ЮУрГУ, 2008. – 103 с.

ИСПОЛЬЗОВАНИЕ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ОБУЧЕНИЯ СТУДЕНТОВ В РАМКАХ ОТДЕЛЬНЫХ КУРСОВ

Карманов М.Л.

karmanov.maxim@gmail.com

аспирант, заведующий сектора ДО ЦЗИиДО ЮУрГУ, г. Челябинск

В данной статье проводится анализ достоинств и недостатков использования коммерческого ПО и СПО для проведения лабораторных работ в рамках обучения студентов по курсам «Вычислительные сети» и «Информационная безопасность в Интернет».

ВВЕДЕНИЕ

В рамках обучения студентов по программам курсов «Вычислительные сети» и «Информационная безопасность в Интернет» возникает необходимость в проведении ряда лабораторных работ, включающих в себя: «Анализ сетевого трафика в ЛВС», «Протокол SMTP», «Виды DNS записей», «Работа DNS сервера», «Протокол HTTP», «Web-уязвимости», «Настройка персонального межсетевого экрана», «Маршрутизация в ЛВС».

Для организации выполнения вышеперечисленных лабораторных работ необходимы следующие программные средства: анализатор сетевого трафика, почтовый сервер, TCP-клиент, DNS-сервер, Web-сервер, набор Web-приложений с уязвимостями, SQL-сервер, персональный межсетевой экран, генератор сетевого трафика.

Существует множество как коммерческих, так и свободно распространяемых программных продуктов, реализующих вышеперечисленный функционал. При этом коммерческие продукты в основной своей массе ориентированы на работу под управлением операционных систем семейства Windows, а свободно-распространяемые продукты во многом ориентированы только на работу под управлением операционных систем семейства *nix. В силу ограничения времени, отводимого на выполнение лабораторных работ, невозможно выполнить каждую лабораторную работу как с использованием коммерческого ПО, так и с использованием свободно-распространяемого ПО (СПО), поэтому перед преподавателем встает проблема выбора.

АНАЛИЗ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ РАЗЛИЧНОГО ПО

Рассмотрим достоинства и недостатки каждого из вариантов по ряду критериев.

1. **Стоимость ПО.** Многие в качестве главного аргумента использования СПО называют его бесплатность, но если речь идет об образовательном учреждении, то стоимость образовательных лицензий на коммерческое ПО обычно весьма невысока, а для ряда программных продуктов и

вовсе равна нулю, поэтому данный критерий в случае образовательных учреждений не является определяющим.

2. Распространенность ПО. При подготовке студентов необходимо учитывать то, с каким программным обеспечением они могут столкнуться после окончания ВУЗа. В настоящее время в России крайне слабо развито использование СПО. Связано это в первую очередь с тем, что до недавнего времени в РФ фактически не действовал закон об авторских правах и все организации совершенно спокойно использовали «пиратские» версии коммерческого ПО. Можно сказать, что «пиратские» версии коммерческого ПО являются СПО. В этом смысле многие небольшие предприятия и практически все домашние пользователи в настоящее время используют СПО [1]. В последнее время в связи с ужесточением действующего законодательства предприятия оказались вынуждены тратить большие средства на приобретение лицензионного коммерческого ПО, что дало существенный толчок процессу внедрения СПО с целью сокращения издержек. В связи с этим в ближайшие годы должен сильно вырасти спрос на специалистов, способных внедрять и использовать СПО.

3. Образовательные возможности ПО. В рамках курсов «Вычислительные сети» и «Информационная безопасность в Интернет» ставится задача изучения технологий и протоколов, а не конкретных программных продуктов. В этом отношении СПО является более выигрышным, так как Windows-ориентированные коммерческие продукты наследуют концепцию компании Microsoft, заключающуюся в максимальном упрощении пользовательского интерфейса. Это в свою очередь приводит к скрытию многих технических тонкостей, что является весьма серьезным препятствием в изучении непосредственно используемых технологий. В СПО, ориентированном на операционные системы семейства *nix ситуация прямо противоположная. Традиционно это ПО является весьма гибким и позволяет изучить все тонкости используемых технологий и протоколов. [2]

4. Необходимые аппаратные ресурсы. Для создания сервера с необходимым функционалом для проведения всех вышеперечисленных лабораторных работ на базе СПО достаточно компьютера с частотой процессора 100 МГц и 128 Мб оперативной памяти. Для создания аналогичного сервера на базе коммерческих продуктов (Windows-like) необходим компьютер с частотой процессора 1 ГГц и 512 Мб оперативной памяти. Сервера для выполнения лабораторных работ удобно готовить с применением технологий виртуальных машин, когда на пользовательском компьютере под виртуальной машиной запускается тот сервер, который необходим для выполнения конкретной лабораторной работы. При таком подходе ресурсы пользовательского компьютера распределяются между базовой операционной системой и операционной системой, работающей под виртуальной машиной. Использование СПО позволяет создать небольшие (менее 300 Мб)

полнофункциональные образы серверов для конкретных лабораторных работ.

Выводы

СПО позволяет осуществлять более качественную подготовку студентов в рамках курсов, связанных с информационными технологиями, но полностью переходить на использование СПО в настоящий момент времени нецелесообразно, так как после окончания ВУЗа большинство выпускников столкнутся на работе с использованием коммерческого ПО.

ЛИТЕРАТУРА

1. Аналитические статьи Business Software Alliance/ <http://www.bsa.org>.
2. Муратов, И., «Обращение Российского клуба Линукс по вопросам развития в Российской Федерации программного обеспечения с открытым кодом»/ И. Муратов // Интернет и образование. – 2008. – ноябрь.

ПРОЕКТ ВНЕДРЕНИЯ СПО В ОБРАЗОВАТЕЛЬНЫЙ ПРОЦЕСС ВЕРЕЩАГИНСКОГО МОУ «ГИМНАЗИЯ». ВТОРОЙ ЭТАП

Краев Е.С., Габова О.Ф.

*ya-kaban@ya.ru,
учитель информатики*

*gabova-of@ya.ru,
учитель химии и информатики*

Верещагинское МОУ «Гимназия», Пермский край, г. Верещагино

В данной работе представлены итоги первого этапа внедрения СПО в учебный процесс образовательного учреждения и технологическая карта механизма организации и реализации второго этапа проекта «Внедрение свободного программного обеспечения в образовательный процесс Верещагинского МОУ «Гимназия».

Целью первого этапа проекта было внедрение в учебный процесс Верещагинского МОУ «Гимназия» свободного программного обеспечения.

При достижении цели реализованы следующие задачи (табл. 1).

Результатом реализации данного локального проекта стало общешкольное мероприятие «Галерея работ учащихся», на котором ученики 5, 6, 7, 9, 11 классов представили свои творческие проекты, выполненные в прикладных программных продуктах ОС LINUX.

Таблица 1

Реализация задач первого этапа проекта

| Задача | Итог |
|--|--|
| <p>1. Установить СПО на всех ПК в кабинете информатики и ПК, задействованных в образовательном процессе.</p> | <p>На сегодняшний день программное обеспечение Linux установлено на 12 машинах в кабинете информатики, на 5 машинах в предметных кабинетах и в кабинете директора. Что составляет 60% от имеющихся ПК.</p> |
| <p>2. Организовать тренинги по общим принципам работы графических интерфейсов, вызову и функционированию основных программных продуктов СПО.</p> | <ul style="list-style-type: none"> • Проведено 7 разъяснительных бесед с учащимися 5–11 классов. (182 учащихся), на которых были объяснены причины установки данного ПО и подробное описание особенностей интерфейса и программных продуктов операционной системы LINUX; • на уроках информатики в 5-7 классах, были организованы практические работы, направленные на изучение прикладных программных продуктов в ОС LINUX; • учащиеся старших классов привлечены к экспертной оценке прикладных программных продуктов встроенных в ОС LINUX, по предложенным критериям (<i>Материал опубликован на сайте</i> http://pspo.perm.ru/news/perm/news/?id=2008050602 – «СПО в образовательном процессе глазами учеников»); • проведен теоретический семинар «Использование возможностей СПО в образовательном процессе» для педагогов гимназии; • проведены практические работы с педагогами, направленные на изучение прикладных программных продуктов в ОС LINUX. |
| <p>3. Обеспечить поддержку пользователей по вопросам использования комплектов СПО и методической литературы.</p> | <ul style="list-style-type: none"> • Для учащихся начальной школы (3-4 класс) организован и проведён кружок «Компьютерная графика для малышей» средствами программы Tux Paint. (Некоторые работы опубликованы на сайтах http://pspo.perm.ru/news/perm/news/?id=2008040701 – «Пионеры школьного линукса» и http://freeschool.altlinux.ru/?p=46 – «Малыши занимаются компьютерной графикой под Linux»); • графический редактор Gimp внедрен в программу специализированного курса «Компьютерная графика» для учащихся 10–11 классов (Работа выпускника данного специализированного курса по теме «Сравнение графических редакторов растрового типа Gimp и Photoshop CS3» на районном конкурсе исследовательских работ заняла 2 призовое место). • 10% уроков по информатике в 5–11 классах переработаны под ОС LINUX. |

Таблица 2

*Технологическая карта механизма организации и реализации проекта
«Внедрение свободного программного обеспечения в образовательный
процесс Верещагинского МОУ «Гимназия»*

| Мероприятия | | Сроки |
|---|---|-----------------------------|
| 1. Семинар по итогам проекта внедрения СПО в школы России и в учебный процесс МОУ «Гимназия». | | Январь 2009 г. |
| 2. Изучение методических материалов по использованию СПО, наработанных в рамках конкурса «Код свободы» и материалов, накопленных сообществом педагогов в сети Internet. | | Январь 2009 г. |
| 3. Апробация образовательных программных продуктов встроенных в ОС Linux Мастер на отдельных уроках по разным предметам. | | Январь – февраль 2009 г. |
| 4. Внедрение СПО в программу специализированных курсов. | Астрофизика – 9 класс СПО: KStars – настольный планетарий для KDE | Февраль – март 2009 г. |
| | Подготовка к ЕГЭ по географии – 11 класс СПО: KGeography – тренировка знаний по географии. | Апрель – май 2009 г. |
| 5. Внедрение СПО в уроки. | Математика – 5–7 класс СПО: KBruch – это небольшая программа, создающая задачи на вычисление дробей. KPercentage – небольшое приложение, помогающее школьникам улучшить их навыки решения простых уравнений с процентами. | Март – апрель 2009 г. |
| | Информатика – 5 класс СПО: KTouch – предоставляет тексты для тренировки навыков набора текста | Февраль 2009 г. |
| | Химия – 8, 9, 11 классы СПО: Kalkium предоставляет разнообразную информацию об элементах Периодической системы. | Март – май 2009 г. |
| 6. Создание базы тестов по разным предметам общеобразовательного цикла. | СПО: KEduca – позволяет создавать интерактивные тесты, основанные на бланках. | Апрель 2009 г. |
| 7. Обобщение опыта на открытой методической конференции в МОУ «Гимназия». | | Май 2009 г. |

Администрацией гимназии разработана система стимулирования педагогов, включающихся в апробацию СПО в учебном процессе.

Опыт работы по внедрению СПО в учебный процесс был представлен на совещании руководителей образовательных учреждений Верещагинского района, а также на Всероссийском творческом конкурсе «Код свободы», в котором авторы статьи вошли в число призеров.

По итогам проведения первого этапа проекта внедрения СПО в МОУ «Гимназия» можно сделать вывод об определенных успехах, но говорить о систематическом использовании СПО в учебном процессе не приходится.

Остается проблемой тот факт, что программное обеспечение, имеющееся на сегодняшний день в гимназии, теряет свою лицензию на использование в 2010 году.

Цель второго этапа внедрения – обеспечить возможность систематического использования СПО в учебном процессе.

Задачи.

1. Изучение методических материалов по образовательным приложениям, входящим в состав СПО.
2. Разработка циклов уроков и специализированных курсов с использованием возможностей СПО.
3. Разработка собственных методических рекомендаций на основании приобретенного опыта по использованию СПО.

В настоящее время создана инициативно-творческая группа педагогов для работы над реализацией проекта внедрения СПО в гимназии (табл. 2).

Критерии оценки.

1. Удовлетворенность участников проекта результатами деятельности.
2. Активность педагогов, учащихся гимназии в реализации проекта.
3. Качество выполненных работ по проекту.
4. Информированность родителей, учителей и учащихся о реализации проекта.

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В СИСТЕМЕ ОБРАЗОВАНИЯ: КОМПЕТЕНТНОСТНЫЙ ПОДХОД

Нилова С.В.

svnil@narod.ru

*кандидат пед. наук, доцент кафедры педагогики
Ивановский государственный университет, г. Иваново*

Автор описывает пути внедрения свободного программного обеспечения (СПО) и работу с ним через разные элементы системы образования: стандарты и программы, органы управления образованием, сеть образовательных учреждений, и соединяет эти составляющие с компетентностным подходом. В статье приводятся некоторые примеры из личного опыта использования СПО в Ивановском государственном университете и МОУ СОШ № 20 (г. Иваново).

ВВЕДЕНИЕ

Внедрение СПО в систему образования происходит в условиях интеграции компьютерных и педагогических технологий. Система образования приобретает новое качество, которое может быть описано через основные особенности технологий (концептуальность, системность, управляемость, эффективность, воспроизводимость). Именно эти особенности стали основой внедрения и использования СПО в МОУ СОШ № 20 г. Иванова. Инновация появилась в школе в 2007 г. благодаря социальному партнерству с родителями учеников школы и студентами Ивановского государственного университета, затем была представлена через курсы Института повышения квалификации учителей и проблемные семинары. Новый этап развития школы зависит от того, сможет ли она стать опорной площадкой внедрения СПО в области и не замкнется ли только на собственном совершенствовании. Своевременная организация курсов для ресурсных центров по линии Института развития образования Ивановской области и обеспечение технической поддержки со стороны студентов вузов может позволить начать качественное внедрение СПО в другие образовательные учреждения области.

В сложившихся условиях недостаточного финансирования путь постепенного внедрения СПО в школах, готовых обеспечить эту идею через самоорганизацию, наиболее оптимален. Важно познакомить школы с возможным выбором ПО, а не создавать очередную проблему информатизации образования. Для этого необходима координация усилий тех структур в области, которые заинтересованы или отвечают за этот инновационный проект. В статье рассмотрены лишь некоторые аспекты использования СПО через призму компетентностного подхода в системе образования.

КОМПЕТЕНТНОСТНЫЙ ПОДХОД И СИСТЕМА ОБРАЗОВАНИЯ

Основой современной модели образования, ориентированной на инновационное развитие экономики, и современных образовательных стандартов становится формирование базовых компетентностей современного человека: «информационной (умение искать, анализировать, преобразовывать, применять информацию для решения проблем); коммуникативной (умение эффективно сотрудничать с другими людьми); самоорганизации (умение ставить цели, планировать, ответственно относиться к здоровью, полноценно использовать личностные ресурсы); самообразования (готовность конструировать и осуществлять собственную образовательную траекторию на протяжении всей жизни, обеспечивая успешность и конкурентоспособность» [1].

Компетентностный подход предполагает сочетание элементов (знаний, умений, личного опыта), которые обеспечивают готовность выпускника вуза (или еще студента) осуществить ту или иную деятельность, в нашем случае внедрить и использовать свободное программное обеспечение. Внедрение СПО, как и компетентностный подход в системе образования, пока еще не осознается педагогическим сообществом в должной степени.

В проекте «Концепции развития разработки и использования свободного программного обеспечения в РФ» написано, что свободное программное обеспечение дает право пользователю свободно использовать, копировать, распространять, изучать, изменять и улучшать его [2]. Проект концепции появился в марте 2008 г. на сайте Министерства информационных технологий и связи РФ, но пока не принят соответствующими органами. Сокращение средств на внедрение СПО на федеральном уровне практически в три раза ставит субъекты РФ в ситуацию внедрения СПО в образовательные учреждения на основе эффективной самоорганизации через соответствующие департаменты, высшую школу регионов и институты повышения квалификации учителей. Однако отсутствие официальной федеральной программы сдерживает практику внедрения СПО в систему образования регионов.

В законе РФ «Об Образовании» представлено определение системы образования «как совокупности взаимодействующих: преемственных образовательных программ и государственных образовательных стандартов различного уровня и направленности; сети реализующих их образовательных учреждений независимо от их организационно-правовых форм, типов и видов; органов управления образованием и подведомственных им учреждений и организаций» [3]. Рассмотрим возможности использования СПО через каждый элемент системы образования.

ПРОГРАММЫ И СТАНДАРТЫ

Обновление содержания этих документов на федеральном уровне, их построение с учетом компетентностного подхода предоставляет возмож-

ность включить использование и изучение СПО в образовательном процессе различных образовательных учреждений и различных направлений подготовки. Например, анализ двух стандартов на историческом факультете ИвГУ по ступеням бакалавр (520800 – История) и специалист (020700 – История) показал, что в рамках цикла «Общие математические и естественнонаучные дисциплины» федерального компонента «Математика и информатика» есть раздел, обозначенный как «аппаратные и программные средства персональных ЭВМ, предназначенные для обработки информации». Для бакалавров возможна организация компьютерного практикума, который может быть организован и на основе СПО, если вуз готов предложить соответствующую техническую основу. Государственная политика в области образования предполагает вариативность, следовательно, информационная компетентность выпускника вуза должна предусматривать знакомство с СПО, и вузы должны это делать. Например, включение в рабочие программы этого учебного курса обязательного компонента сравнения операционных систем обеспечило бы нормативные требования к содержанию образования подготовки бакалавров, специалистов, магистров высшего образования.

Вузовский (школьный) или региональный компонент любого стандарта может предусматривать курс «Пользователь ПК» на основе СПО или курсы, ориентированные на сравнение возможностей двух операционных систем (Windows или Linux). Кроме этого в вузах может быть организована дополнительная образовательная программа по направлению «Администратор Linux». Подобный опыт возможен, если управленцы, от которых зависят эти решения, осознают вероятную потребность в таких специалистах и возможность экономии бюджетных средств на приобретение программного обеспечения в регионе.

Внедрение СПО в вузе может происходить не только на основе специальных предметов и дополнительных программ. Так, обычное сопровождение электронного учебно-методического комплекса (ЭУМК) в Интернете (<http://www.ivanovo.ac.ru/distant/>) по любому вузовскому предмету может способствовать развитию СПО. В комплексе студент может скачать лекции по курсу педагогики в формате документа OpenOffice.org. Это является развивающим элементом курса, практической встрече с которым предшествует знакомство с проблемой вариативности ПО. Наряду с этим, в рамках курса педагогики или курсов «Новые педагогические технологии», «Инновационная педагогика», «НИТ в учебном процессе», студентам может быть предоставлена возможность реализовать проект по изучению или использованию СПО. В курсах предусматривается блок «Программное обеспечение и культура личности». Таким образом, внедрение СПО через стандарты и программы во многом зависит от компетентности специалистов, которые осуществляют ту или иную деятельность.

УПРАВЛЕНИЕ СИСТЕМОЙ ОБРАЗОВАНИЯ

В практике применения ИКТ в образовательных учреждениях и, порой, на уровне регионального и муниципального управления отсутствует понимание того, что будет происходить с ИКТ через год или через несколько лет. Перспектива использования СПО в системе образования пока не осознается большинством педагогического сообщества, включая и региональное управление.

Сама система управления образованием меняется за счет активной реализации принципа государственно-общественного управления. Так, Ивановская область является участником комплексного проекта модернизации образования, и этому принципу в системе образования уделяется достаточное внимание. В школах созданы управляющие советы, в которые входят учителя, родители и школьники. Практически каждая школа разрабатывает свои стратегические программы развития.

Вопрос внедрения и использования СПО может быть решен через управляющие советы образовательных учреждений и стать важной составляющей экономии бюджетных средств, когда в 2010 г. закончится действие лицензий на пакет «Первая помощь 1.0». Продление срока действия лицензии на коммерческое программное обеспечение может оказаться не по карману большинству школ Ивановской области, соответственно через эту стратегическую идею и можно начать региональную работу. Но уже сейчас необходимы точные сведения о стоимости лицензий для системы образования. Так, некоторые частные школы, которые не получили «первой помощи», уже вкладывают достаточные суммы на ПО. Внедрение Linux требует гораздо меньших материальных затрат, и не станет обременительным для бюджетов муниципальных объединений, если уже сейчас начать последовательную работу за счет возможностей областных Институтов развития образования и созданных ресурсных центров информатизации образования в районах. Соответственно эти учреждения, реализуя свою миссию, могут опираться на уже накопленный опыт опорных школ проекта внедрения СПО.

Пока для большинства школ сделать шаг в сторону СПО достаточно сложно. А если вдруг Microsoft решит совсем не брать денег за ПО для системы образования? Наше «авось» поддерживается еще и нежеланием преподавателей и управленцев что-либо менять в своей работе; компетентностью администраторов в вопросах СПО. Пока нет полного перехода на финансирование школ по нормативно-подушевому финансированию, велика роль указаний сверху. От уровня компетентности специалистов областного департамента, который отвечает за информатизацию образования, зависит достаточно много. Если бы в оценке деятельности органов управления образованием всех уровней присутствовал критерий экономии бюджетных средств и из их фонда оплаты труда брали деньги на закупку

лицензий для ПК, то такая зависимость заставила бы всех активно меняться и находить внутренние резервы для развития СПО. Даже просто ответ на вопросы: «Куда направить деньги? На закупку проприетарного ПО и поддержку зарубежной компании? Или создать дополнительные рабочие места для молодого специалиста в районе, способного обеспечить поддержку СПО?» – заставил бы задуматься некоторых управленцев при выборе ПО.

Внедрение СПО происходит в России как сверху, так и снизу. Самоорганизация на уровне отдельных школ РФ показала, что идея внедрения СПО снизу весьма продуктивна, «главным показателем успешности проекта можно считать добровольное присоединение к нему более 990 образовательных учреждений по всей территории РФ, которые абсолютно самостоятельно приняли подобное решение» [4].

СЕТЬ ОБРАЗОВАТЕЛЬНЫХ УЧРЕЖДЕНИЙ

Этот элемент рассмотрим через потенциальную возможность организации сети образовательных учреждений, внедряющих СПО в отдельном субъекте или объединении школ на уровне Российской Федерации.

Важные стороны организации сети рассмотрим на примере федерального конкурса «Код свободы» (2008 г.). Подключение школ не из пилотных территорий (Республика Татарстан, Пермский край, Томская область) к конкурсу «Код свободы» показал, что в субъектах РФ уже есть точки роста для внедрения СПО. Важным этапом внедрения СПО в 2009 г. может быть опора на такие образовательные учреждения, а не только действие через вертикаль управления.

Однако на сайте проекта мы не нашли ответа на вопрос, а какие школы, а может быть и вузы (сгруппированные по субъектам РФ) имеют опыт работы с СПО, кроме пилотных территорий, которые имеют сайты поддержки. Это важная информация для создания сообщества и дальнейшего продвижения идеи не требует финансовых вложений и, наверное, может быть выложена на сайте проекта внедрения СПО до появления соответствующего портала. В условиях организации конкурсов инновационных школ в регионах подобная информация является плюсом школам, применяющим СПО, и привлекает внимание к внедрению СПО в регионе. Возможное объединение участников по разрабатываемым проблемам использования СПО, в рамках даже минимальных средств, может привести к хорошим результатам. Так, следующим этапом в организации конкурса «Код свободы» может быть участие сетевых команд, объединенных общими проблемами. Например, МОУ СОШ № 20 г. Иванова (руководителем проекта внедрения СПО в этой школе является автор статьи) готова решать общие проблемы: 1) преподавание информатики на основе СПО в средней школе; 2) использование СПО для начальной школы (в том числе, как дополнительную образовательную услугу); 3) разработка тестов по различным

предметам; 4) работа школьного терминального сервера; 5) использование мультимедийной доски на основе СПО; 6) документооборот на основе Хронографа, работающий под СПО, и другие, которые могут быть связаны с развитием школы как опорной площадки внедрения СПО в регионе.

Рассмотрим модель создания сети в регионе на примере МОУ СОШ № 20 г. Иванова, когда инициатива идет снизу от управляющего совета школы и позиции родителей, которые способны добровольно сопровождать инновационный процесс в школе в сотрудничестве с учителями. Инновационная деятельность школы проходит по теме «Информатизация образовательного процесса» и включает в себя проект 2002–05 гг. «Дистанционные технологии в образовательном процессе школы» и проект «Внедрение и использование свободного программного обеспечения (СПО) в образовательном процессе» 2007–10 гг. Каждый из этих проектов начал существовать в школе благодаря позиции родителей и готовности школы включиться в это сотрудничество.

Рассмотрим второй проект. В 2006 г. школа получила новый компьютерный класс без ПО, который практически около года не действовал из-за отсутствия средств на программное обеспечение. В 2007 г. началась работа по установке и настройке СПО, которая проходила с участием студентов Ивановского государственного университета (в рамках учебного модуля «ИКТ в учебном процессе») под руководством системного администратора и вузовского преподавателя, являющимися родителями учеников этой школы [5]. В 2008 г. на базе школы был организован учебный курс «ИКТ в воспитательном процессе» для преподавателей области (по линии областного ИПКиППК) на основе СПО, где уже в качестве преподавателя работал учитель информатики этой школы М. В. Голубева. Этот курс прошли и 10 преподавателей школы. Третья часть педагогического коллектива школы, включая директора школы Е. А. Горюнову, получила представление о СПО. Следующим этапом стало проведение городского проблемного семинара по внедрению СПО для учителей информатики города и участие школы в городском образовательном форуме с презентацией проекта «Внедрение и использование свободного программного обеспечения в образовательном процессе». Таким образом, на уровне города появились предпосылки для формирования сети проекта, с учетом потенциала одной опорной школы и привлечения студентов вуза для администрирования в других образовательных учреждениях (для студентов важно освоение стандартных моделей настройки Linux в школьных компьютерных классах).

Следующий этап работы школы был связан с ситуацией поиска средств на внедрение СПО в регионе через участие молодежи в этом проекте и развития школы как опорной площадки. Так, в рамках проектной деятельности курса педагогики студентами юридического и физического факуль-

тета ИвГУ (вузовский преподаватель комплексно подходит к формированию компетентностей студентов разных факультетов, иногда в рамках проектов объединяет их вместе и обогащает тем самым содержание учебного процесса) в школе был организован круглый стол по проблемам внедрения СПО в Ивановской области. В обсуждении приняли участие представители Молодежной общественной палаты и Областного молодежного правительства, представители различных политических молодежных объединений и управления образованием г. Иванова. Этот этап внедрения позволил на безфинансовой основе построить перспективу социального партнерства в решении обозначенных проблем. Внедрение СПО – это задача для молодежи. Однако во многом дальнейшие шаги по использованию СПО в Ивановской области будут зависеть от согласованных усилий в регионе. На наш взгляд, координаторы федерального проекта внедрения СПО могут рассматривать отдельную школу как площадку для последовательного и эффективного внедрения СПО в систему образования регионов.

Комплексное решение проблемы возможно через привлечение студентов и соответствующие учебные практики вузов, способствующие развитию опыта на основе возможных стандартных решений. Компетентностный подход в подготовке специалистов предполагает увеличение роли высших учебных заведений в развитии региональной системы образования.

ЛИТЕРАТУРА

1. Современная модель образования, ориентированная на решение задач инновационного развития экономики // Материалы Министерства образования и науки РФ для обсуждения на августовских педагогических совещаниях работников образования «Актуальные задачи современной модели образования». – 2008.

2. Концепция развития разработки и использования свободного программного обеспечения в РФ // Сайт Министерства информационных технологий и связи РФ.

<http://minkomsvjaz.ru/ministry/documents/1117/3264.shtml>

3. Об образовании: Закон РФ // Собрание законодательства Российской Федерации. – 1996. – № 3.

4. Якушин, А. А. Результаты разработки и апробации пакета свободного программного обеспечения (ПСПО) для общеобразовательных учреждений Российской Федерации в 2007–2008 годах / А. А. Якушин // Свободное программное обеспечение в высшей школе: тез. докл. конф. – Переславль. – 2009. – 30 января – 1 февраля. <http://www.altlinux.ru/media/book-thesis-Pereslavl-2009-4.pdf>

5. Нилова, С. В. Стратегия развития ИКТ в системе образования / С. В. Нилова // Вестник ИвГУ Серия «Естественные, общественные науки» – 2008. – Вып. 1. – С. 9–18.

ОПЫТ МИГРАЦИИ НА СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В КЮТ «ПРИБОРИСТ»

Павлова Н. С.

nata_love_dm@mail.ru

педагог дополнительного образования, вторая квалификационная категория, ГОУ ДОД Омская областная станция юных техников, филиал КЮТ «Приборист», г. Омск

В статье описан опыт перехода на СПО в Омской областной станции юных техников. Опыт приобретен в результате самостоятельного изучения проблемы использования СПО в Российской Федерации, проведения бесед и дискуссий с ребятами об изменении в законодательстве в сфере защиты авторских и смежных прав, о приоритетном национальном проекте «Образование», знакомства учащихся с письмом Министерства образования и науки РФ от 1 февраля 2007г (№15-51-4бин/01-10), с документацией, входящей в состав ОС Linux (Linux в школьном образовании, под ред. К. Маслинского). проведён опрос учащихся. Приведены данные мониторинга результатов опроса учащихся по результатам перехода на СПО.

ВВЕДЕНИЕ

Столкнувшись с изменениями в законодательстве и не возможностью приобрести проприетарное ПО – встал вопрос о переходе на СПО, как одном из способов выхода из сложившейся ситуации. Так как средствами клуба невозможно было закупить даже ОС на каждый компьютер – встал вопрос о полном переходе на СПО.

Для решения данной проблемы я с февраля 2008 года самостоятельно изучала проблемы использования СПО в России: востребованность обществом перехода на СПО, практику внедрения СПО в образовательный процесс, недостатки и достоинства перехода на СПО и непосредственно саму ОС Linux.

Установив и на рабочем компьютере (второй системой) Linux я постепенно подготавливала учащихся к переходу на СПО. Также рассказывала про изменения в законодательстве, показывала преимущества ОС Linux (отсутствие вирусов, зависаний и т.д.), делала сравнительную характеристику стоимости проприетарного ПО и Свободного ПО, рассказывала про различные дистрибутивы Linux и т.д.

В августе 2008 г. установила на каждый компьютер главной системой ALTLinux Junior 4.0. Учащиеся восприняли переход на СПО хорошо (в т. ч. и новички).

1. ПОДГОТОВКА К МИГРАЦИИ

КЮТ «Приборист» – филиал Омской областной станции юных техников. В классе 8 компьютеров (включая компьютер преподавателя). Обучение ведётся по адаптированной образовательной программе «Введение в информатику». Программа предназначена для детей и подростков 3–11 классов и рассчитана на изучение основ информатики и вычислительной

техники в системе дополнительного образования в течение пяти-шести учебных лет.

Основные причины миграции на СПО

1. Изменения в законодательстве.
2. Высокая стоимость легализации. Невозможность приобретения проприетарного ПО.
3. Высокая конкурентоспособность СПО.
4. Соответствие программного обеспечения обучению по образовательной программе.
5. Нечувствительность ОС Linux к компьютерным вирусам.
6. Личное предпочтение преподавателя.

При выборе главной операционной системы рассматривались 2 варианта: EDUbuntu и ALTLinux Junior 4.0. Источники дистрибутивов – Омская и Томская региональные компьютерные сети.

Основные критерии выбора дистрибутива

Наличие дружественного интерфейса (близкого к Windows).

1. Наличие русифицированного меню и программ.
2. Наличие программ, соответствующих обучению по образовательной программе.
3. Наличие поддержки.
4. Личное предпочтение педагога.

В итоге на 7 компьютеров был установлен ALTLinux Junior 4.0, а на компьютер преподавателя – ALTLinux Desktop 4.0.

Достоинства перехода на СПО

1. Простота использования и настройки, полная совместимость форматов файлов и документов с Windows.
2. Возможность выбора графической среды (Gnome, KDE, XFCE).
3. Стабильность и надёжность функционирования системы.
4. Отсутствие вирусов.
5. Повышение познавательного интереса учащихся.

Недостатки перехода на СПО

1. Отсутствие технической поддержки в г. Омске.
2. Отсутствие обучающих курсов и семинаров по внедрению и использованию СПО.

2. ОПЫТ ИСПОЛЬЗОВАНИЯ СПО В КРУЖКЕ

В начале учебного года проводила беседы и дискуссии с ребятами об изменении в законодательстве в сфере защиты авторских и смежных прав, рассказывала о приоритетном национальном проекте «Образование», ознакомила учащихся с письмом Министерства образования и науки РФ от 1 февраля 2007г (№15-51-46ин/01-10), ознакомила с документацией, входящей в состав ОС Linux (Linux в школьном образовании, под ред. К. Мас-

линского). Также для повышения интереса учащихся к СПО использовала сравнение стоимости необходимого минимума программного обеспечения на 1 компьютер (см. таблицу).

Для обучения учащихся младшей группы («компьютерная графика как средство развития творческого потенциала») использую программы GCompris, TuxPaint, Inkscape, позволяющие усваивать учебный материал в легко доступной форме. Программа Kturtle позволяет познакомить учащихся с основами программирования и используется для обучения как младшей группы, так и старшей. Для изучения основ программирования старшей группой используется FreePascal. Изучение OpenOffice практически не отличается от изучения Microsoft Office, напротив, позволяет повысить интерес учащихся к освоению программ офисного пакета.

Выводы

В январе 2009 года для мониторинга результатов перехода на СПО учащимся было предложено ответить следующие вопросы.

1. Работали ли Вы с СПО до начала обучения на курсах (если, да, то указать с каким);
2. Что Вам понравилось при работе с ALTLinux Junior?
3. Что Вам не понравилось при работе с ALTLinux Junior?
4. Каких программ Вам не хватает при работе с ALTLinux Junior?
5. Хотели бы Вы изучить другой Linux (если да, то указать какой).

На первый вопрос 100% учащихся ответили – нет, т.е. до посещения кружка ребята ни разу не работа с операционными Linux-системами. Основными достоинствами Linux ребята выделили:

- 1) разнообразие программ;
- 2) графический интерфейс;
- 3) простота и удобство использования;
- 4) не «зависает», как Windows.

Недостатков ребята практически не увидели. Основными ответами были: «Меня всё устраивает», «Менее привычный, чем Windows» и «Я его недостаточно изучил, чтобы говорить о том, что не понравилось».

На 4-й вопрос подавляющее большинство учащихся ответило – «всего хватает». А желание изучить другой дистрибутив Linux изъявили всего лишь 20% учащихся (Mandriva 2009)².

² Второй системой на компьютер педагога установлена Mandriva 2009.

Стоимость программного обеспечения

| Проприентарное ПО | | СПО | |
|--|-----------------|--------------------|-----------------|
| Программа | Стоимость, руб. | Программа | Стоимость, руб. |
| Microsoft Windows XP Home Edition | 2 736,00 | ALTLinux Junior | 0 |
| Microsoft Office Home and Student 2007 Win32 Russian BOX | 3 024,00 | OpenOffice.org | 0 |
| ESET Antivirus NOD32 Standart | 1 080,00 | – | – |
| Far Manager | 750,00 | Midnight Commander | 0 |
| Winrar | 1 390,00 | ARK | 0 |
| CorelDRAW Graphics Suite X4 Rus | 12 350,00 | Inkscape | 0 |
| Adobe Photoshop CS3 | 28 500,00 | GIMP | 0 |
| Итого: | 49 830,00 | Итого: | 0 |

Из всего вышеперечисленного можно сделать следующие выводы.

1. Переход на СПО произошёл практически безболезненно, как для преподавателя, так и для учащихся.
2. При использовании СПО отпала необходимость переустанавливать систему, сократилось количество сбоев и «зависаний» компьютера, что положительно повлияло на степень усвоения образовательной программы.
3. Использование СПО позволяет повысить интерес к предмету, способствует развитию логического мышления и творческого потенциала учащихся.
4. Для обучения по образовательной программе идеально подходит ALTLinux Junior 4.0 (что подтверждается результатами тестирования).

РАЗРАБОТКА СИСТЕМЫ ОЦЕНОК И КОНТРОЛЬНЫХ ЗАДАНИЙ ДЛЯ РАБОТЫ В ГРАФИЧЕСКОМ РЕДАКТОРЕ GIMP

Панюкова Т.А.

kwark@mail.ru,

к.ф.-м.н., доцент, кафедра ЭММуС, ЮУрГУ, г. Челябинск

В статье предложены критерии оценок заданий, предлагаемых студентам в качестве курсовых и семестровых работ. Разработана шкала оценок.

ВВЕДЕНИЕ

Очень быстро, на глазах одного поколения, произошли революционные изменения в технологии подготовки цветных изображений. В наше время на персональных компьютерах готовятся к публикации не только высококачественные издания, презентационные материалы, но и документы, содержащие деловую графику: диаграммы, схемы и пр. Можно сказать, что проблема управления цветом стала массовой. Современные технические и программные средства дают возможность удовлетворительного решения значительной части этой проблемы.

ПРОГРАММЫ РАССМАТРИВАЕМЫХ КУРСОВ

Изучение пакетов прикладных программ, которые позволяют провести дорепечатную подготовку изображений, содержится не только в учебных планах дисциплин типа «Пакеты прикладных программ» творческих и гуманитарных специальностей (например, «Связи с общественностью», «Реклама» и пр.), но и ряда технических и экономических специальностей, поскольку задача грамотного представления, оформления и публикации данных, составления презентаций актуальна для организаций и структур всех форм собственности.

Например, фрагмент учебной программы дисциплины «Пакеты прикладных программ», составленных для специальностей «Статистика» и «Математические методы в экономике») имеет следующий вид (табл. 1).

Таблица 1

| Вид учебной работы | Всего часов |
|-------------------------------|-------------|
| Общая трудоемкость дисциплины | 60 |
| Аудиторные занятия | 34 |
| Лабораторные работы | 34 |
| Самостоятельная работа | 26 |
| Вид итогового контроля | зачет |

Из 34 часов программе GIMP отводится 14 часов, из них 8 часов лабораторных работ и 6 часов самостоятельной работы.

Фрагмент учебной программы дисциплины «Компьютерные издательские системы» приведен в табл. 2.

Таблица 2

| Вид учебной работы | Всего часов |
|-------------------------------|------------------------|
| Общая трудоемкость дисциплины | 60 |
| Аудиторные занятия | 54 |
| Лекционные занятия | 18 |
| Лабораторные работы | 36 |
| Самостоятельная работа | 6 |
| Вид итогового контроля | зачет, курсовая работа |

Из общего объема дисциплины на изучение GIMP отводится 4 лекционных часа, 4 часа лабораторных работ и столько же часов предусмотрено на самостоятельную работу студента.

Оба курса содержат в вышеперечисленном объеме занятий следующие разделы.

1. Основные концепции работы в The GIMP, сравнение возможностей GIMP и Photoshop. Работа с кистями. Понятие слоя. Цветовые режимы изображения. Размеры и разрешение изображения. Сравнительный анализ и особенности графических форматов. Сравнительная характеристика GIMP и Photoshop. Рисование с помощью кистей. Создание пользовательских кистей.

2. Выделенные области в GIMP. Основные принципы создания коллажей. Основы цветокоррекции. Редактирование сканированных изображений. Различные типы выделений. Анализ выделяемой области и особенности выбора инструментов выделения. Работа с областями прозрачности и масками. Анализ композиции и выбор способа наложения слоев. Основные этапы создания фотоколлажа. Цветовая и тоновая коррекция изображения. Повышение контрастности и яркости сканированного изображения или фотографии документа. Подготовка изображения для помещения в текстовый документ и дальнейшей печати.

3. Работа с текстом и контурами в GIMP. Создание листовок и приглашений. Однострочный и многострочный текст. Форматирование символов и абзацев. Особенности текстового слоя.

4. Фильтры и GIF-анимация в GIMP. Основные принципы работы с анимацией. Фильтры для стилизации фотографий, искажения изображений, рисования.

ПРИМЕРЫ ТВОРЧЕСКИХ ЗАДАНИЙ ДЛЯ РАБОТЫ В GIMP

Разработанные задания составлены таким образом, что они охватывают не только знакомство с основными функциями программы, рассматривают основные концепции GIMP, но и соответствуют конкретным разделам программы изучаемой дисциплины.

Составлено 4 творческих задания.

Рисование кистями. Нарисовать изображение, состоящее из фона и как минимум трёх пересекающихся между собой слоев. При создании изображения можно пользоваться кистями, ластиком, инструментами размытия и осветления/затемнения. Запрещено использовать заливку и выделенные области.

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.

Афиша. Придумать и создать афишу, на которой присутствует как однострочная, так и многострочная текстовая информация. Применить эффекты для акцентирования внимания на текстовой информации.

Структурная схема. С помощью преобразования выделения в контуры придумать и нарисовать разветвленную структурную схему. Схема должна иметь блоки разной формы, а линии переходов между блоками должны заканчиваться стрелками (или другими указателями).

Очевидно, что компьютерное изображение или фотомонтаж должен удовлетворять правилам построения композиции, цветопередачи, работы со световыми источниками. Но эти критерии носят несколько неформальный характер, и не охватывают оценки знаний технических особенностей используемой программы.

Разработка шкалы критериев для творческих заданий по компьютерной графике является актуальной задачей, т.к. при ее отсутствии в распределении оценок не участвует такой важный критерий как степень ознакомления обучающегося с программным продуктом.

Оценку за каждое задание разобьем на **две составляющие**: техническое исполнение (степень ознакомления с инструментами GIMP) и субъективный критерий (композиция, качество исполнения).

Заметим, что в каждом из заданий присутствует усложняющий элемент. Например, в первой задаче предлагается создать изображение со взаимопересекающимися слоями. Такие ограничения в задании с одной стороны усложняют задачу студенту, с другой – ограничивают его возможности. Все это необходимо потому, что всех обучаемых можно условно разделить на три категории.

1. *Энтузиасты*, увлеченные полученным заданием. Обычно такие люди самостоятельно осваивают программное обеспечение вглубь и представляют работы высокого качества.

2. *Отличники*, выполняющие полученное задание досконально с целью получить высокую оценку. Их работы обычно не так интересны, нежели работы энтузиастов, но при этом имеют достаточно высокое качество и соответствуют выданному заданию.

3. *Минималисты*, которые выполняют задание только для того, чтобы получить зачет или оценку любого уровня. Работа выполняется по минимуму, формально соответствует приведенным в постановке задачи требованиям, но при этом имеет низкое качество.

Введение ограничений в формулировках заданий в первую очередь касается третьей категории студентов, которые при отсутствии критериев получили бы высокий балл за использование инструментария: формально требования к заданию выполнены. При отсутствии ограничений студент, работающий по минимуму, не осваивает многих основных концепций работы GIMP.

СИСТЕМА ОЦЕНКИ ЗАДАНИЙ В GIMP

При работе с графикой у любого пользователя всегда есть возможность проявить творческие способности. Это приводит к тому, что возникает необходимость формализации критериев оценки таких работ, что всегда является не простой задачей.

Для повышения качества выполненных заданий целесообразно не ограничивать обучающихся некоторым множеством операций и элементов, выполняемых над тем или иным изображением. Более эффективным будет введение критериев, используемых при оценке выполненных работ. Каждый критерий имеет некоторый вес (количество баллов за представленный элемент). Таким образом, для получения оценки обучающийся должен набрать некоторый минимальный суммарный балл по всем критериям. Система оценок должна быть открытой, чтобы каждый мог выбрать те элементы, которые он будет реализовывать в своей работе.

Рассмотрим систему оценок на примере задания «Рисование кистями»

При выполнении задания обучающийся должен уяснить, каким образом создаются многослойные изображения, научиться раскладывать объекты по слоям, уметь использовать кисти различной формы и разновидности.

Так, в число обязательных элементов следует включить разбиение композиции на слои и предлагается назначить следующую систему оценки этого действия:

- 3 – четкое 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).

В качестве произвольных элементов можно указать все возможные инструменты кисти, и для каждого из них указать шкалу:

- 4 – созданы и использованы пользовательские кисти;
- 3 – использование стандартных кистей различных форм и размеров;
- 2 – использование различных форм кистей стандартного размера;
- 1 – использование одного вида кисти различного размера;
- 0 – использование кисти одного вида стандартного радиуса.

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.

Таким образом, минимальный возможный балл, достаточный для получения отличной оценки, равен $3+4*3=15$. В результате, для получения зачета 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».

Аналогичным образом можно разработать подобные шкалы и для других творческих заданий. Для этого необходимо выделить обязательные инструменты или элементы композиции, определить оценки за каждый объект и подсчитать минимальный допустимый балл для получения зачета.

Количество баллов, необходимое для получения оценки, вычисляется как процент от минимального балла: 50–60% – оценка «удовлетворительно», 60–80% – оценка «хорошо», 81% и выше – оценка «отлично».

ЗАКЛЮЧЕНИЕ

Разработанная система критериев позволила значительно увеличить качество выполняемых заданий, формализовать, а в некоторых случаях и уменьшить, возможное проявление субъективности со преподавателя, а также повысить качество усвоения разделов соответствующих курсов, связанных с изучением редактора растровой графики GIMP.

РАЗВИТИЕ ТВОРЧЕСКОГО ПОТЕНЦИАЛА УЧАЩИХСЯ СРЕДСТВАМИ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ПРИМЕРЕ ИСПОЛЬЗОВАНИЯ ГРАФИЧЕСКОГО РЕДАКТОРА TUX PAINT НА УРОКАХ ИНФОРМАТИКИ

Перевушина А.В.

*avpervushina@mail.ru,
МОУ гимназия № 93, г. Челябинск*

Главным аспектом нашей работы является анализ использования свободного программного обеспечения в развитии творческого потенциала учащихся на примере внедрения графического редактора Tux Paint на уроках информатики. Для достижения данной цели необходимо дать определение понятия «творчество».

М.Г. Ярошенко в социально-психологическом словаре приводит следующее определение понятия «творчество» – деятельность, результатом которой является создание новых материальных и духовных ценностей. Оно предполагает наличие у личности способностей, мотивов, знаний и умений, благодаря которым создается продукт, отличающийся новизной, оригинальностью, уникальностью.

Новейший философский словарь следующим образом раскрывает понятие «творчество» – высшая форма универсально понимаемой креативности, имманентно присущая всем уровням иерархии бытия; способствует самосохранению и воспроизведению сущего посредством качественных трансформаций их структур. По мнению американского ученого П. Хилла, «творчество – это успешный полет мысли за пределы неизвестного, оно дополняет знания, способствуя созданию вещей, которые не были известны ранее». Польский исследователь А. Матейко считает, что сущность творческого процесса заключается в реорганизации имеющегося опыта и формировании на его основе новых комбинаций.

Одна из главных задач школы состоит в том, чтобы не только дать знания учащимся, но и пробудить личностный мотив, привить интерес к обучению, тягу к самосовершенствованию, «научить учащихся учиться». В современной педагогике все больше намечаются тенденции к перемещению акцента с методов, обеспечивающих процесс усвоения знаний, на технологии, позволяющие обеспечить общее развитие личности ребенка. Одним из главных методов данного развития является творчество.

Творческий потенциал³ представляет собой сложное, интегральное понятие, которое включает в себя природно-генетический, социально-личностный и логический компоненты, в совокупности, представляющие собой знания, умения, способности и стремления личности преобразовать (улучшить) окружающий мир в различных сферах деятельности в рамках общечеловеческих норм морали и нравственности. Проявившийся в той или иной сфере деятельности «творческий потенциал» представляет собой «творческие способности» личности в конкретном виде деятельности, а также сложное личностно-деятельностное образование, включающее мотивационно-целевой, содержательный, операционно-деятельностный, рефлексивно-оценочный компоненты, отражающие совокупность личностных качеств и способностей, психологических состояний, знаний, умений и навыков, необходимых для достижения высокого уровня его развития. Сам термин часто может употребляться как синоним «творческая личность», «одаренная личность». Ценность творчества, его функции, заключаются не только в результативной стороне, но и в самом процессе творчества.

В основе современной педагогической науки лежит понимание о человеке как о существе творческом. Именно в творчестве раскрывается его сущность как преобразователя мира, творца новых технологий и идей. В обществе все чаще и чаще, в связи с проблемой творчества, говорят о творческой личности, которую наделяют такими чертами, как направленность, самопознание, умение видеть проблему, анализировать ситуацию, мобилизовать знания, выдвигать гипотезы, оценивать результаты, критически мыслить и т.д.

В основу технологии опыта положена идея о развитии творческих способностей учащихся на уроках информатики через организацию проектной деятельности и направленности учебно-познавательной деятельности на результат. Внешний результат можно будет увидеть, осмыслить, применить на практике. Внутренний результат – опыт деятельности – станет бесценным достоянием, соединяющим знания и умения, компетенции и ценности.

Главенствуясь данной идеей, нами был внедрен графический редактор TUX PAINT, использующийся на уроках информатики в МОУ гимназии № 93 г. Челябинска. Кроме этого были разработаны необходимые методические рекомендации для обеспечения качественного образовательного процесса.

Опыт использования данной программы показал следующие результаты.

³ Потенциал (от лат. – сила) – в широком словоупотреблении трактуется как средства, запасы, источники, имеющиеся в наличии, а также средства, которые могут быть мобилизованы, приведены в действие, использованы для достижения определенной цели.

1. За период работы с TUX PAINT число детей, занимающихся в творческих кружках, увеличилось на 18%.
2. Была организована выставка детских работ, показавшая заинтересованность учащихся всех возрастов в использовании данной программы.
3. Работая в команде с учителем ИЗО, был выявлен следующий ряд показателей:
 - восприятие цвета улучшилось в среднем на 12%;
 - восприятие графических объектов повысилось на 24%;
 - восприятие общей композиции улучшилось на 6%.
4. Консультация психолога выявила:
 - общий уровень творческой активности в группах вырос;
 - замкнутые, малокоммуникабельные дети начали проявлять социальную активность.
5. Вырос интерес к использованию информационных технологий в повседневной жизни и в частности к операционной системе Linux.

ОС FREEBSD КАК ПЛАТФОРМА ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ СЕТЕВЫХ ТЕХНОЛОГИЙ

Прокопов И.И.

sub@drts.susu.ac.ru,

доцент, кафедра ЦРТС, ЮУрГУ, г. Челябинск

Рассматриваются основные возможности операционной системы UNIX FreeBSD при создании защищенной локальной сети. Особо отмечается возможность подробного изучения механизмов настройки системы при создании защищенных конфигураций ОС и при настройке различных сетевых служб. Данная ОС может использоваться как многофункциональный элемент для обеспечения безопасности в локальной сети при реализации различных топологий. Неприхотливость к аппаратным ресурсам и межплатформенная мобильность также является одним из достоинств, которое позволяет использовать систему при подготовке администраторов по безопасности.

ВВЕДЕНИЕ

Современные локальные сети строятся на основе структурирования (сегментации) и защиты (изоляция) отдельных сегментов друг от друга с помощью программно-аппаратных устройств [1, 2] (управляемые коммутаторы, маршрутизаторы, межсетевые экраны) или программным способом при специальной настройке ОС на ЭВМ с несколькими сетевыми интерфейсами. Сеть Интернет, если есть подключение к ней, также можно считать отдельным сегментом.

При организации защищенной сети удобно использовать ЭВМ с операционной системой FreeBSD для применения и изучения механизмов маршрутизации, методов структуризации и защиты сети. Нетребовательность

к аппаратным ресурсам позволяет снизить расходы на создание учебных сетевых полигонов по сравнению с программно-аппаратными устройствами. При этом с помощью даже одного компьютера можно смоделировать практически весь функционал программно-аппаратного комплекса из нескольких дорогих устройств.

ОС FreeBSD как маршрутизатор и шлюз в Интернет

Для нормального функционирования сети требуется структурировать ее (разделить на сегменты) [2]. Идеальная интрасеть разделена, по крайней мере, на три компонента [1]. Разделение интрасети на сегменты осуществляется установкой защитных средств, таких как межсетевые экраны (МСЭ). Средства защиты могут быть частью готового аппаратного решения, функцией автономного маршрутизатора или программой, выполняемой на специальном сервере.

Одним из способов деления на сегменты является физическая изоляция. В этом случае в соответствующий пограничный (между сегментами) сервер под управлением ОС FreeBSD устанавливаются несколько сетевых карт, каждая из которых подключена к своему сегменту. С помощью настроек программного обеспечения сервера можно добиться нужной степени связи сегментов друг с другом, либо полного ее отсутствия. Недостатком такого способа является малое количество «коммуницируемых» сегментов из-за физического ограничения на количество сетевых карт (3–4 шт.), но для многих случаев этого достаточно.

Одним из частых применений является использование ОС FreeBSD в качестве шлюза для подключения корпоративной сети к Интернет. При этом на сервере необходимо реализовать функции файрвола, трансляции адресов (NAT), и при необходимости, учета трафика. Иногда, в силу специфики предприятия, требуется реализовать особые конфигурации, связанные с возможностями работы с сетью Интернет.

Обычно под интрасетью понимают сеть, созданную для работников данного предприятия. В ряде случаев требуется обеспечить доступ извне к ресурсам компании другим группам пользователей. Одним из способов обеспечения совместного доступа к информации в интрасети является организация экстрасети (extranet), которая представляет собой часть интрасети, доступную извне из других сетей, в том числе из Интернет.

В качестве дополнительного средства защиты в сети может применяться нейтральная зона (DMZ), или экранированная подсеть. Она гарантирует, что предоставление широкого доступа к открытым ресурсам не приведет к нарушению защиты внутри сети. Существует три наиболее распространенных варианта организации DMZ [3]:

- с одним брандмауэром, имеющим минимум три интерфейса (внутренняя сеть, внешняя, зона DMZ);

- с двумя брандмауэрами. DMZ расположена между внешним и внутренним брандмауэрами. Уровень безопасности повышенный, т.к. существует два рубежа защиты внутренних ресурсов;

- с тремя брандмауэрами. Две зоны DMZ – внутренняя и внешняя. Чрезвычайно высокий уровень безопасности.

ОС FreeBSD предлагает два различных механизма пакетной фильтрации: IPFW и IPFILTER. Брандмауэр IPFW является традиционной частью программного обеспечения, поставляемого с FreeBSD. Это система фильтрации и учета пакетов, поддерживаемая ядром системы. С помощью гибкой системы правил можно регулировать прохождение пакетов через шлюз на уровне номеров портов, IP адресов, имен сетевых служб. Также посредством настроек брандмауэра можно предотвращать некоторые виды сетевых атак и воздействий от неправильно функционирующих сетевых устройств.

На базе одного сервера под управлением ОС FreeBSD, имеющего три сетевые карты и включенный встроенный брандмауэр IPFW, можно настроить первый вариант DMZ. Один интерфейс – связь с Интернет, второй – для систем, использующих преобразование адресов NAT, третий – для систем, которые не должны использовать NAT (демилитаризованная зона).

КОНФИГУРИРОВАНИЕ ЯДРА

Одной из интересных возможностей ОС UNIX FreeBSD в плане использования для работы и обучения является хорошо документированная, в том числе на русском языке, методика настройки и сборки нового ядра операционной системы. Конфигурирование ядра представляет собой центральную часть настройки FreeBSD и подобных ей систем [4, 5, 6]. Такая процедура необходима при реализации на сервере защитных функций:

- трансляции адресов NAT с опцией ядра IPDIVERT;
- файрвола с опцией ядра IPFIREWALL;
- учета и регулирования трафика с опцией ядра IPFIREWALL_VERBOSE;
- других возможностей, для которых требуется запуск служб в привилегированном режиме.

В особо враждебном окружении можно задавать повышенные уровни защиты ядра с помощью опции `kern_securelevel` в файле `/etc/rc.conf`. При повышенной защите ядро нельзя изменять или устанавливать новое (замещение ядра). Это позволяет надежно защитить ядро от посягательств злоумышленников. Но для этой опции требуется провести ряд экспериментов с разными конфигурациями, чтобы выяснить приемлемость выбранного повышенного уровня защиты ядра для сервера конкретной организации. В некоторых случаях система может быть ограничена в запуске служб до уровня бесполезности.

Также перекомпиляция ядра дает возможность значительно повысить производительность системы [5], а это в свою очередь дает возможность использования менее производительного оборудования в качестве аппаратной платформы сервера FreeBSD, что важно для машин, которые отвечают минимальным требованиям этой ОС.

Конфигурация ядра FreeBSD задается с помощью текстовых файлов. Такой метод может показаться сложным, однако он обеспечивает гибкость, которой нет у визуальных методов настройки. Применение текстового файла конфигурации имеет следующие достоинства:

- позволяет поддерживать несколько конфигураций, хранящихся в разных файлах;
- позволяет увидеть одновременно все настройки и опции устройств, которые поддерживаются данной конфигурацией ядра;
- можно воспроизвести конфигурацию другой аналогичной системы, скопировав ее `config`-файл и пересобрав ядро;
- можно взять настройки системы из предыдущих версий полностью или частично при полной переустановке системы с новой версией ОС, с необходимой коррекцией;
- можно передавать файл конфигурации другим людям для изучения и использования;
- легкость возврата к предыдущим конфигурациям ядра, в том числе к основной (стандартной) GENERIC.

На основе разных версий текстового файла конфигурации можно собрать несколько разных ядер, уже готовых к работе, и иметь возможность достаточно быстро замещать текущую версию ядра на новую экспериментальную. При этом время на перекомпиляцию ядра (20-75 минут в зависимости от быстродействия ЭВМ) уже не потребуется. Время тратится только на выполнение команды замещения ядра и перезагрузку системы.

В учебном плане изучение конфигурации ядра помогает понять принципы построения высокопроизводительных и защищенных ядер и осознанно подходить к защите ОС в целом.

СЕТЕВАЯ БЕЗОПАСНОСТЬ

Обеспечение сетевой безопасности предполагает решение многих задач, одной из которых является обеспечение внутренней безопасности, и в частности, борьба с нелегальными подключениями чужих ЭВМ или устройств к сети. Одним из механизмов защиты от такого рода подключений является сопоставление MAC и IP адреса хоста в специальной таблице в файле `/etc/arphosts`. Такая таблица позволяет не допустить присвоение действующих IP-адресов в сети чужими компьютерами и заблокировать выход в Интернет через шлюз на базе FreeBSD. Соответственно все эти нарушения будут запротоколированы и пересланы администратору системы.

Работа в сети предполагает использование большого количества служб и сетевых протоколов. Уязвимость операционной системы повышается при увеличении количества задействованных служб, предоставляя злоумышленникам возможности для сетевых атак на каждую из служб. Сетевые сервисы обеспечиваются специальными демонами FreeBSD, которые загружаются автоматически после установки и все время находятся в оперативной памяти, ожидая запросы. При редком поступлении запросов напрасно расходуются системные ресурсы и нет дополнительного контроля обращений к сервисам.

ОС FreeBSD предоставляет возможность полного контроля списка активных служб, а также дополнительного контроля при поступлении запросов к службе из сети. В большинстве UNIX-систем для вызова демонов сетевых служб используется супердемон `inetd` [5, 6]. Сервер, вызванный супердемоном, не становится демоном и прекращает свое существование после закрытия соединения. В результате в памяти остается только один `inetd`. Если `inetd` активирован в файле `/etc/rc.conf`, то далее необходимо в файле конфигурации `inetd.conf` открыть необходимые сервисы. Изначально все строки этого файла закомментированы, т.е. сервер никаких сетевых соединений через `inetd` принимать не будет. Таким образом, по умолчанию реализована модель безопасности «все запрещено». Это позволяет при установке ОС в сети не подвергать сервер атакам через активные сервисы в момент его проверки и конфигурирования. Супердемон позволяет также с помощью специального встроенного механизма `tcp-wrappers` указывать с точностью до отдельных хостов, кому можно предоставлять запрошенный сервис, а кому нельзя. Перечень ограничений указывается в стандартном файле `/etc/hosts.allow`.

Выводы

1. Свободно распространяемая операционная система Unix FreeBSD имеет в своем составе все необходимое для реализации основных компонентов, требуемых при организации защищенных локальных сетей (маршрутизатор, брандмауэр, транслятор адресов).

2. Наличие возможности пересборки ядра позволяет создавать оптимизированные хорошо защищенные системы в зависимости от потребностей предприятия и конкретной конфигурации аппаратной платформы сервера.

3. Открытый код системы и возможность получения любого количества экземпляров ядра системы позволяет в учебных центрах и индивидуально изучать механизмы функционирования и защиты системы. Это дает возможность подготавливать грамотных системных администраторов и администраторов по безопасности.

4. Система позволяет во многих случаях производить замену аппаратной платформы с сохранением всех настроек сетевых служб либо без за-

мены ядра на измененное, либо с небольшими изменениями при пересборке ядра.

ЛИТЕРАТУРА

1. Милославская, Н.Г. Интрасети: доступ в Internet, защита: учебное пособие для вузов / Н.Г. Милославская, А.И. Толстой. – М.: ЮНИТИ Пресс, 2000. – 527 с.
2. Олифер, В.Г. Компьютерные сети. Принципы, протоколы, технологии: учебник для вузов / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2006. – 960 с.
3. Хилл, Б. Полный справочник по Cisco.: пер. с англ. / Б. Хилл. – М.: Издательский дом «Вильямс», 2008. – 1088 с.
4. Лепаж, И. UNIX. Библия системного администратора.: пер. с англ. / И. Лепаж, П. Яррера. – М.: Дилектика, 1999. – 528 с.
5. Брайан Т. FreeBSD. Энциклопедия пользователя.: пер. с англ. / Т. Брайан, М. Эбен. К.: ООО «ТИД «ДС». – 2002. – 736 с.
6. Торн, А.В. FreeBSD: установка, настройка, использование / А.В. Торн, А.В. Федорчук. – СПб.: БХВ-Петербург, 2003. – 640 с.

ПРИМЕНЕНИЕ НАУЧНЫХ ПАКЕТОВ С ОТКРЫТЫМ КОДОМ ПРИ РАБОТЕ СО ШКОЛЬНИКАМИ И СТУДЕНТАМИ МЛАДШИХ КУРСОВ

Рожков А.В., Гаврилова Т.П.

д.ф.-м.н., ros@kb.susu.ac.ru

профессор, профессор кафедры ЦРТС, ЮУрГУ, Челябинск

*старший преподаватель кафедры общей математики,
ЮУрГУ, Челябинск*

В научном мире широко известен пакет GAP (группы, алгоритмы, программирование), текущая версия 4.4.12 свободно распространяемый, официальный сайт <http://www.gap-system.org/>, содержащий кроме самого ядра системы, со встроенным языком программирования, более 70 различных пакетов по многим разделам алгебры, теории чисел и дискретной математики. Пакет разработан под операционную систему Unix/Linux, но имеет эмуляцию под Windows. Данный пакет хорошо приспособлен для проведения обширных вычислительных экспериментов в области алгебры и теории чисел, где до сих пор остается много интересных задач, доступных для понимания школьников.

ВВЕДЕНИЕ

В математике имеется много трудных и интересных задач до сих пор не поддающихся полному решению. Далеко не все из них требуют обширных и глубоких познаний. Особенно много подобных задач в алгебре. Еще в 19 веке Ф.Клейн писал, что в алгебре можно прочитать несколько

определений и сразу садиться за серьезную научную работу. Во многом эта ситуация сохраняется и сегодня, особенно в разделах, связанных с теорией чисел. Формулировки некоторых задач весьма просты, но также далеки от решения как сотни лет назад.

При кружковой работе со школьниками нами решались некоторые из подобных задач.

Пакет GAP достаточно нетребователен к аппаратным ресурсам, имеет встроенный паскалеподобный язык программирования и несколько тысяч операторов-функций. В каком-то смысле GAP это обширный калькулятор для целочисленных и символьных вычислений в алгебре.

ЗАДАЧИ ИЗ ТЕОРИИ ЧИСЕЛ

В теории чисел много трудных и интересных задач, доступных понимающей школьникам и студентам младших курсов. Аналитическое решение многих из них не найдено за несколько сотен лет усилий многих ученых. Однако проведение вычислительного эксперимента вполне доступно и может быть полезным даже для серьезной науки.

Пакет GAP оказывается весьма полезным при проведении подобных вычислительных экспериментов в силу своей бесплатности и высокой функциональности.

Хорошо известно, что натуральное число называется совершенным, если оно равно сумме своих собственных делителей, т.е. Делителей, отличных от самого числа.

Таких чисел сейчас известно довольно много, самые знаменитые из них:

$$6 = 1 + 2 + 3, 28 = 1 + 2 + 4 + 7 + 14.$$

Со времен Евклида известно, что числа вида $2^{p-1}(2^p-1)$, где p и (2^p-1) – простые, всегда совершенны. Второй сомножитель в этой формуле знаменит сам по себе – это простое число Мерсенна, Кстати, самое большое простое число известное на данный момент человечеству (февраль 2009 г.), на его поиск было потрачено более года распределенных вычислений, как раз и есть число Мерсенна $2^{4311609} - 1$, предположительно 46 по счету.

Среди проблем о совершенных числах самая известная – это утверждение, что совершенное число нечетным быть не может. Нечетных совершенных чисел пока не найдено, но и доказательства, что их нет, тоже не предъявлено.

Можно проводить вычислительные эксперименты, в надежде найти нечетное совершенное число.

Обобщением совершенных чисел являются так называемые дружественные числа. Поскольку мы хотим обобщить это понятие еще больше, то введем некоторые определения.

Пусть A – натуральное число, $S(A)$ – сумма собственных делителей этого числа.

Совершенное число – это неподвижная точка введенного отображения, поскольку $S(A) = A$.

Дружественные числа, это такие числа A и B , что $S(A) = B$, $S(B) = A$, что равносильно выполнению равенства $S(S(A)) = S^2(A) = A$.

Изменяя эту формулу можно ввести очень естественное обобщение понятия совершенных и дружественных чисел:

есть ли числа вида $S^n(A) = A$, если $n > 2$?

Ответ на этот вопрос не известен. Школьниками, в кружковой работе, проводились обширные вычислительные эксперименты по поиску таких чисел. А вернее, просто изучалось поведение функции $S(A)$.

Осуществлялись итерации этой функции, прослеживалось поведение, образов отображения S . Встретилось лишь три варианта – ответа:

– совершенное число – цикл длины 1,

– дружественные числа – цикл длины 2,

– последовательности, заканчивающиеся числом 1. Это происходило, когда одна из сумм равнялась простому числу, у которого, естественно, сумма собственных делителей равна 1.

Например, $S(100) = 117$, $S(117) = 65$, $S(65) = 19$, $S(19) = 1$.

Следует отметить, что в вычислительных экспериментах встречались и ситуации, когда последовательность итераций функции S явно начинала уходить в бесконечность. Здесь срабатывали ограничения на вычислительные ресурсы, и процесс завершался.

Неизвестно, есть ли последовательность $S(A)$, $S^2(A)$, $S^3(A)$, ... , уходящая в бесконечность.

Другая задача о целых числах – это поиск опровержения гипотезы Гольдбаха – одного из первых криптографов России, служившего Отчеству во времена Екатерины Великой.

Гипотеза формулируется в полстрочки – «каждое четное число – это сумма двух простых чисел». В настоящее время гипотеза проверена и подтверждена для всех четных чисел меньших $12 \cdot 10^{17}$.

Поиск контрпримера также очень простая задача, доступная школьникам.

Еще одна, любопытная задача, связанная с целыми числами, это поиск числа N такого, что $L(N!) = N$. (задача из серии занимательной арифметики, но вряд ли простая). Функция L – это сумма цифр в десятичной записи числа N , а восклицательный знак означает произведение всех натуральных чисел от 1 до N .

Например, $L(6!) = L(1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6) = L(720) = 9$, $L(10!) = L(3628800) = 27$.

ЗАДАЧИ В ОБЛАСТИ АЛГЕБРЫ

Трудных задач в алгебре тоже достаточно, но простых по формулировке меньше, чем в теории чисел. Со школьниками решались следующие задачи.

Первая из них важна для теории периодических групп.

Пусть p – простое число. Рассмотрим векторное пространство V размерности p над полем Галуа $GF(p)$, содержащим p элементов. Требуется найти минимальный набор векторов W , координаты которых могут быть только 0 и 1, такой, что каждый вектор пространства V , ортогонален, хотя бы одному вектору из множества W .

Очевидно, что множество W обязательно содержит вектор $(1, 1, \dots, 1)$ и стандартный базис $(1, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (0, \dots, 0, 1)$. Т.е. множество W содержит как минимум $p+1$ элемент. Но уже для случая $p=5$ вычисление точного значения для числа элементов во множестве W , а главное, перечисление самих множеств W , затратная вычислительная задача.

Следующая интересная, в первую очередь для криптографии, задача, это нахождение примитивных элементов конечных полей. Примитивным элементом называется любой порождающий элемент мультипликативной группы поля. Примитивные элементы конечных полей используются во многих криптографических протоколах, а также для создания цифровой подписи.

Также для криптографии важным является нахождение неприводимых многочленов над конечными полями. Они используются в задачах генерации псевдослучайных последовательностей и протоколах аутентификации. Все эти задачи вполне по силам школьникам, а алгоритмы их решения прекрасно программируются в GAP.

ЛИТЕРАТУРА

1. Рожков, А.В. Воскресная математическая школа. Вып. 1–7 / А.В. Рожков – Челябинск: Изд-во ЧелГУ, 1989–1994.

2. Рожков, А.В. Теоретико-числовые методы криптографии. Учебное пособие / А.В.Рожков, Ниссельбаум О.В. – Тюмень: Изд-во ТюмГУ, 2008. – 140 с.

3. <http://www.gap-system.org/>

ОСОБЕННОСТИ ВНЕДРЕНИЯ СВОБОДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ В УСЛОВИЯХ ЛИЦЕЙСКОГО ОБРАЗОВАНИЯ

Ханнанова С.Т.

licey78_chelny@mail.ru,

*учитель высшей квалификационной категории информатики и ИКТ,
Лицей № 78 им. А.С. Пушкина, г. Набережные Челны, респ. Татарстан*

После первых же уроков становится ясно – опасение напрасно, учащимся нравится работать в Linux. Все по-русски!!! Работа над разработкой методического материала к урокам показала, что в Linux это делать удобней. ПСПО – разработчики снабдили множеством программ, которые можно использовать и по другим предметам школьного курса. Появляются организационные проблемы из-за малой изученности Linux. Благодаря возможности устанавливать ОС Linux на виртуальную машину VirtualBox или установки пакета программ OpenOffice.org 3.0, Mozilla Firefox, Mozilla Thunderbird, Gimp на Windows в лицее начинают применять в своей деятельности СПО учителя-предметники, администрация. Наличие Live CD и Live DVD дисков, позволяет устанавливать и изучать учащимся Linux дома. Пока нет достаточных навыков администрирования ОС Linux, наиболее оптимальным является установка виртуальной машины и использование дисков Live DVD.

ВВЕДЕНИЕ

Мы, учителя информатики, привыкли, что программное обеспечение у нас меняется с периодичностью в 2–3 года. В 2008 году, уходя в отпуск, было принято решение, что в новом учебном году переходим на Windows Vista и Microsoft Office 2007, что было обещано ученикам. Выйдя на работу после отпуска, увидела письмо «Об использовании пакета свободного программного обеспечения для общеобразовательных учреждений Российской Федерации». У нас почти всегда есть выбор, он был и у меня: установить Linux, доложить о выполнении работы и забыть, впереди еще два года можно жить спокойно, так как можно использовать пакет «Первой помощи», или попытаться все-таки использовать новое программное обеспечение в учебном процессе. Решение пришло само собой. По совместительству являюсь «Специалистом по обслуживанию ЭВМ», и установку ПО выполняю сама. Пришлось выбирать какую версию Linux из предложенных устанавливать, а за тем заниматься установкой. Процесс установки на 25 компьютеров довольно таки продолжительный, параллельно ознакомилась с Мастер Linux и осмысливала возможные варианты внедрения данного программного обеспечения в процесс обучения и деятельность нашего лицея.

LINUX, УЧИТЕЛЬ-УЧЕНИК

В нашем лицее курс информационные технологии изучают с 5-го класса мальчики в рамках предмета технология, а программа курса разработана мной, как пропедевтический курсу информатики и ИКТ. В 7-м классе

предполагалось изучать WEB-технологии. С HTML-языком знакомство началось у них в конце 6-го класса, в 7-м классе предполагалось, что они будут разрабатывать сайт с помощью визуальных средств. Такое средство в Alt Linux нашлось, более того его интерфейс оказался проще и понятней начинающим, чем знаменитая Dreamweaver – это программа – *Quanta Plus*. В интернете нашлось учебное пособие [1]. Любой учитель информатики знает, что если не работает техника или программа работает не стабильно, урок вести крайне сложно, а что ожидать от не изученной программы – это вопрос. В сентябре с ребятами ведется предварительная беседа:

- объясняем, что такое Linux, зачем его надо изучать;
- объясняем, что программа мало изучена и что многие аспекты придется изучать вместе и учитель может не знать ответа: почему так;
- объясняем, что переход на эту ОС даст возможность будущим нашим специалистам, которые хотят заниматься профессиональным программированием работать на собственный бренд.

После первых же уроков становится ясно – опасение напрасно, учащимся нравится работать в Linux (рис. 1). Все чему они научились работать в Windows: работа с файлами, создание презентаций, набор и оформление текста, работа с графикой, можно применять, а главное все по-русски. Общие принципы такие же. Простейший графический редактор *KolourPaint* оказался более функциональным, чем его аналог Paint в Windows. Он решал все проблемы по подготовке графики для подключения к web-страницам, мы использовали его и обошлись без так называемых профессиональных графических средств (рис. 2).

Особенно мне, как учителю, понравилось следующее. При изучении тем, связанных с кодированием информации, форматами файлов учащиеся получают оценки «5», а когда надо применить эти знания на практике они удивленно произносят: «Это разве было про это!!!». Теперь ещё не начав изучать указанные темы, они четко представляют разницу в кодировках *UTF-8, Windows-1251*; прекрасно разбираются, почему какой то файл не читается в программе и без особых усилий, выучили какие форматы в какой программе обрабатываются, потому что в Linux все расширения отражаются программе *Konqueror* в отличие от его аналога Проводника в Windows.



Рис. 1. Объяснение материала



Рис. 2. Практическая работа

Началась работа над разработкой методического материала к урокам. В этой области больших проблем не возникло, программы из Linux читали и перерабатывали, хотя и с небольшими неточностями, форматы документов Windows. Вся методическая копилка, собранная за многие годы, подлежит последующему использованию. Сделала открытие, если готовить методический материал в Linux сразу, то его можно применять и в Windows, просто надо сохранить в нужном формате. Мною давно уже используется безбумажный вариант задания, это файлы-задания. При выполнении таких работ, учащиеся тяжело переключались из одного документа (выполнения задания) в другой (текст задания). Если давался шаблон, то надо было долго объяснять учащимся, откуда его открывать и как сохранять. Все время было опасение, что исходный документ будет случайно изменен учащимися, несмотря на то, что на него устанавливался атрибут «только чтение». В OpenOffice.org Writer, есть возможность сохранять в разных форматах, соответственно многие проблемы решались намного проще.

Учащиеся теперь не путают задания от заготовки. Текст задания сохраняется в формате *.pdf* и он открывается в другой программе *KPDF*, такой документ учащимся случайно изменить затруднительно. Текст – заготовку для обработки в OpenOffice.org Writer, в которой учащиеся выполняют задания, сохраняешь как шаблон (*.ott*) и он открывается без имени. Учащиеся уже понимают, что надо дать имя, а не пытаются сохранить со старым. Теперь облегчается переключение через панель задач, так как отличаются внешне значки программ. А мне как учителю, стало легче редактировать свои методические разработки и изменять их в зависимости уровня обученности класса.

Замечательные открытия продолжаются. СПО разработчики снабдили множеством программ, которые можно использовать на практических работах по темам: десятипальцевый метод печати, кодирование и измерение информации, системы счисления, файловая система и графический и тек-

стовый редакторы. А так же программами, которые можно применять по другим предметам школьного курса (рис. 3).

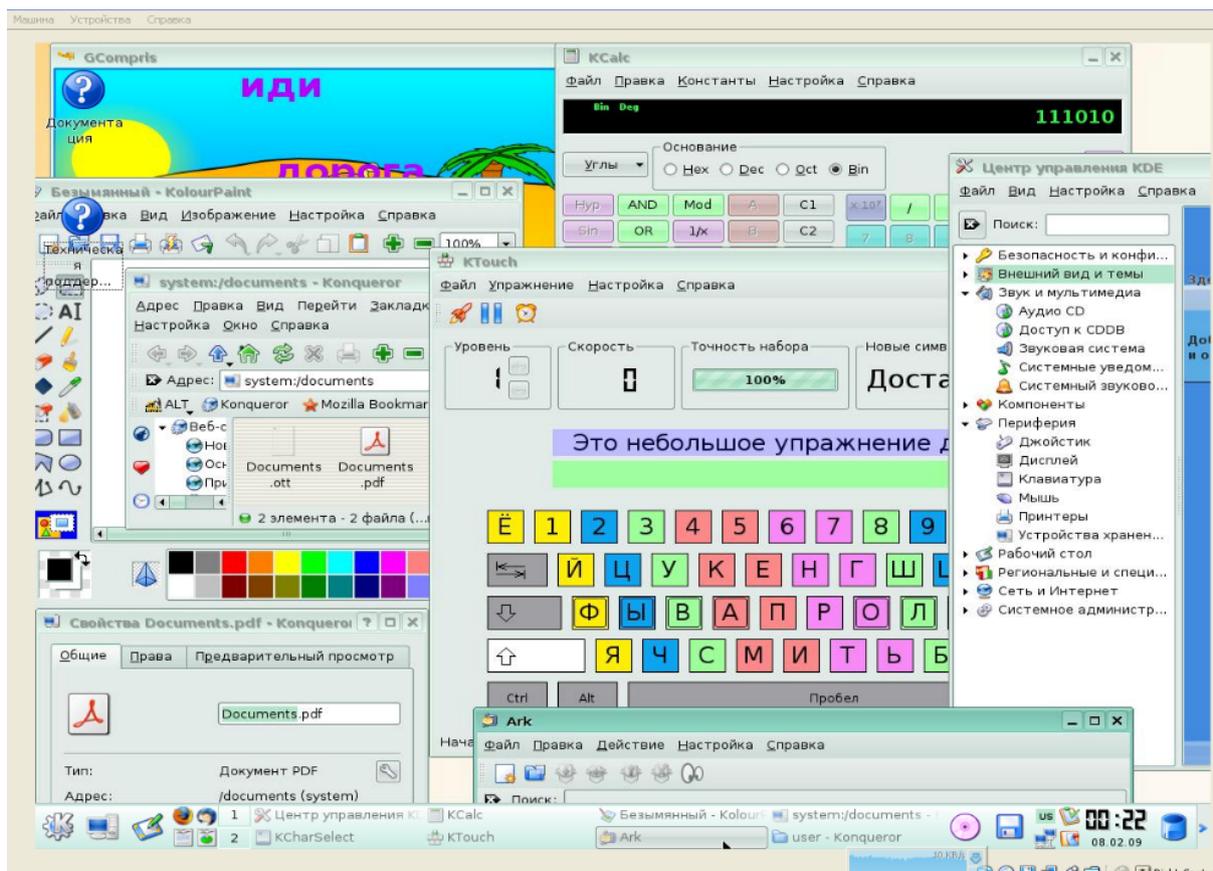


Рис. 3. Программное обеспечение СПО

Поэтому было принято решение, что пора знакомить с Linux 9-е классы. Но программу не растянуть, а у учащихся дома в основном стоит Windows и убедить их, что надо изучать никем не используемый на данный момент Linux, просто бесполезно. По программе в данный период было изучение текстового процессора. Практические и лабораторные работы (как и было обещано весной), выполняли в *Microsoft Office Word 2007*, а на контрольную работу загружали Linux и текстовый процессор *OpenOffice.org Writer*. Текст задания даётся тот же самый, что и при выполнении практической части, а учащимся предлагается найти те же функции в программе OpenOffice.org Writer. Справляются без особых проблем. Особо радуются, что с незнакомым интерфейсом они могут достичь нужного результата, что ничего нет особенного в этом загадочном Linux. И возникают осознанные вопросы: А что, скоро все будут работать на Linux? А в чем разница? А почему тогда предприятия не переходят на Linux? А правда что из под Linux мы более защищены от вирусов? А что если, я буду знать программирование я и сам смогу, что-то изменить, так как мне хочется, в операционной системе и в любой её программе? А как дома его

поставить? И особенно популярным у нас в лицее становятся Live DVD, Live CD.

ДВЕ СТОРОНЫ МЕДАЛИ: ПРИЕМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА

С другой стороны, у меня на уроках, возникает ряд проблем. Учащиеся сразу же заметили, что в Linux контроль за ними ослаблен, есть возможность поиграть. Последнее меня как учителя настораживало, с одной стороны играть было разрешено официально, как стимул, но с другой стороны при такой свободе объем изучения материала резко сократился, ослабла дисциплина. Технические характеристики компьютеров в учебных классах в нашем лицее достаточно высокие (компьютеры на базе x86, тактовая частота процессора 2,4 ГГц, оперативная память – 2 Гб, жесткий диск – 80 Гб, монитор: SVGA, разрешение 1024x768 пикселей, 16,7 млн. цветов), несмотря на это проектор не работал с двумя мониторами одновременно (скорее всего драйверов видео карты NVIDIA GeForce 9800 GT для Linux еще не разработано), поэтому демонстрировать учебный материал во время объяснения, как мы привыкли, на экраны ПК учащихся или на экран с помощью проектора не было возможности, приходилось объяснять на одном ПК «кружочком».

Стала искать нужные аналоги программных средств в Linux, которые позволяли мне: контролировать и управлять ученическими ПК, как в Windows – NetOp School, возможность подключения проектора с моей видео картой, возможность использования MIMO пока из под Windows.

На республиканском семинаре в Казани, который был в октябре услышала, что работа по этим направлениям ведется. Более того, что те учителя, которые участвуют в конкурсе «Код свободы», прислали работы с решением некоторых проблем. Приходится ждать...Задумываюсь! Потом надо будет учиться все это настраивать – это опять время. А пока были найдены другие пути решения данных проблем.

В нашем лицее ведется систематическая научно-исследовательская работа учащихся, в результате которой, они имеют возможность участия в различных конференциях. Для выполнения таких работ комплекта программ из «Первой помощи» недостаточно. Поэтому по просьбе учителей информатики лицей выписывает журналы «Computer Bild» и «CHIP», в которых есть диски с новейшим программным обеспечением, многие из них являются свободно распространяемыми. Благодаря данным источникам [3, 4] узнала, что есть так называемая виртуальная машина *VirtualBox*, которая позволяет установить вторую ОС внутри другой. Установив эту программу, а затем Linux в ней, я для себя решила несколько организационных проблем сразу.

- Самая главная: возможность эксперимента, безболезненно для учебного процесса. Если что-то ни так настроил, просто переустановить Linux. А это значить эксперименты, все смелее и смелее!!! Ведь настройка и уста-

новка новых программ и драйверов в Linux, для нас неопытных, это большой риск.

- Не нужно перезагружаться из одной ОС в другую на ПК для того, чтобы вести урок.

- Появляется возможность контролировать деятельность учеников с учительского ПК, пока хотя бы, средствами WINDOWS – NetOp School, мы привыкли с ней работать и отказываться от того сервиса не хочется.

- Решена проблема проектора, теперь становится возможным объяснять урок и демонстрировать все на экран посредством проектора (см. рис.1).

- Учащиеся без проблем смогли установить Linux на своих домашних ПК в качестве второй системой и продолжают изучать его самостоятельно.

СПО ВЫХОДИТ ЗА ДВЕРИ КАБИНЕТА ИНФОРМАТИКИ

Если что-то получается, а вокруг идут разговоры о том что это неприемлемо, то конечно же хочется закричать всем, что это не так. Я соглашусь с высказыванием Д.Томсона «Из всех услуг, какие могут быть оказаны науки, величайшая из них – введение в её обиход новых идей».

Именно, поэтому для регионального семинара учителей информатики (рис. 4), который проводился базе нашего лицея, было подготовлено выступление и фрагмент урока с учащимися 7-го класса по теме: «Приемы преподавания web-технологий с помощью Quanta Plus в СПО». Реакция присутствующих учителей была неоднозначна. При этом 90% коллег, присутствующих на уроке, обрадовались и успокоились. Многие кто не начал использовать в своей практике СПО, ушли с твердыми намерениями, что они будут работать в Linux.



Рис. 4. Урок на региональном семинаре

Так как внедрение свободного ПО в наших школах началось, то стала приходить почта с непонятными форматами для наших секретарей и администрации. Опять возникла дилемма: бегать и делать самой или учить их. Вот только времени на это ни у кого нет. В нашей школе два учителя информатики, обязанности «Специалиста по обслуживаю» они делят между собой, учебная нагрузка более 34 часов в неделю на каждого, кроме этого занимаются подготовкой к олимпиадам и руководством научных работ. Когда же, тогда заниматься обучением администрации? Выход опять подсказал журнал «Computer Bild» [3], пакет программ *OpenOffice.org 3.0*,

Mozilla FireFox, Mozilla Thunderbird, Gimp из-под Windows и все русскоязычное. Установка данного программного обеспечения на ПК – это возможность **плавного** перехода к ОС Linux.

Сначала они научатся обрабатывать текстовые, табличные и графические документы с помощью этих программ в Windows (рис. 5.), а потом и в Linux будут с ними работать, так как пользователь не очень ощутит разницу в ОС, ведь интерфейс у них очень похож. А когда дистрибутивы даешь установить на домашние компьютеры и объясняешь, что эти программы в отличие от «Первой помощи», не требуют регистрации и ключей, то есть можно свободно установить на ПК и не бояться, что тебя обвинят в пиратстве – это восторг!!!



Рис. 5. Зам. директора по информатизации Мухтярова Э.И.

ЗАКЛЮЧЕНИЕ

Слова Д.Медведева: «Если Россия не будет независимой в области программного обеспечения, то все сферы, в которых мы хотим быть независимыми и конкурентоспособными, окажутся под угрозой» подтвердили правильность моего выбора в сентябре. Рада, что не отнеслась к письму формально, а эти слова были главным аргументом при объяснении учащимся, почему надо изучать Linux, хорошим стимулом, чтобы отнестись спокойно к глюкам Linux и неожиданностям, которые преследовали меня в течение всего времени и помогли мне найти свою тактику внедрения СПО в нашем лице. Мой опыт показывает следующее.

- Изучать и внедрять СПО надо постепенно – это установка СПО на Windows, а затем переход к ОС Linux.
- Разделы учебного плана несущие прикладной характер изучать по схеме: Windows изучение нового материала, Linux – контроль знаний.
- Современный ученик должен знать и уметь работать в обеих операционных системах и с различным программным обеспечением
- В теоретических курсах, для проведения лабораторного практикума, как можно чаще, использовать ПО Linux, так как средства представленные разработчиками более нагляднее и проще в использовании.
- Надо организовывать научную работу учащихся в направлении изучения и разработки программ из под Linux.
- Пока нет достаточных навыков администрирования ОС Linux, наиболее оптимальным является установка виртуальной машины и использование дисков Live CD и Live DVD.

С появлением СПО заметно повысился интерес учащихся к программированию, учащиеся лица проявляют интерес к Linux и пытаются выполнять свои творческие работы для конкурсов с её использованием.

За период участие в апробации разработан методический материал к урокам по темам: текстовый процессор (OpenOffice Writer), текстовый редактор (Kwrite и Kate), создание сайтов (web-редактор Quanta Plus), десятипальцевый метод (тренажер Ktouch). Это позволило участвовать в конкурсе «Код свободы» и стать её призером. Указанные методические разработки организаторы конкурса разместили на сайте [5].

Благодаря найденному техническому решению: диски Live CD и Live DVD, установка ОС Linux на виртуальную машину и установка пакета программ OpenOffice.org 3.0, Mozilla FireFox, Mozilla Thunderbird, Gimp из под Widows, в лицее начинают применять в своей деятельности СПО учителя– предметики и администрация.

Хотелось отметить что ни смотря на начальную реакцию и различное мнение ОС Linux все же занимает свое место у пользователей XXI века. Благодаря стараниям разработчиков, сам Linux довольно быстро развивается и думаю скоро не будет уступать ни в чем Windows, а может даже и опережать.

ЛИТЕРАТУРА

1. Машковцев, И.В. Создание и редактирование Интернет-приложений с использованием Bluefish и Quanta Plus (ПО для создания и редактирования Интернет-приложений): учебное пособие / И.В. Машковцев. – М. – 2008. –74 с.
2. Яковлев, С. Первые шаги в Linux / С. Яковлев // Журнал Chip, №11. – 2008.
3. Computer Bild. Журнал. – №13–24. – 2008.
4. <http://freecode.psp.ru/360/index.html> .

РЕАЛИЗАЦИЯ КОНТЕКСТНОГО ПОДХОДА К ОБУЧЕНИЮ СПЕЦИАЛИСТОВ ПО ИНФОРМАЦИОННЫМ ТЕХНОЛОГИЯМ СРЕДСТВАМИ ПОРТАЛА ITEM74.RU

Цытович П.Л.

pavel.tsytovich@item74.ru

к.т.н., доцент кафедры ЭВМ, ЮУрГУ, г. Челябинск

Подготовка специалиста в области информационных технологий требует качественно иного подхода к организации процесса обучения основам программирования и проектирования программного обеспечения по сравнению с традиционной схемой «лекция-практика-лабораторная работа». Основная проблема любого специалиста подготовленного в современной высшей школе состоит в оторванности от реальных условий будущей работы. Для разрешения указанного противоречия мы предлагаем использовать образовательный портал как средство реализации контекстного подхода к обучению, предложенный А.А. Вербицким, который заключается во введении элементов профессиональной деятельности студентов в процесс обучения. Контекстное обучение опирается на теорию деятельности, в соответствии с которой, усвоение социального опыта осуществляется в результате активной, пристрастной деятельности субъекта. В нём получают воплощение следующие принципы: активности личности; проблемности; единства обучения и воспитания; последовательного моделирования в формах учебной деятельности слушателей содержания и условий профессиональной деятельности специалистов. Особое внимание обращается на реализацию постепенного, поэтапного перехода студентов к базовым формам деятельности более высокого ранга: от учебной деятельности академического типа к квазипрофессиональной деятельности (деловые и дидактические игры) и, потом, к учебно-профессиональной деятельности [1].

Современный специалист в области разработки программных систем, как минимум, должен обладать следующими компетенциями:

- анализ предметной области, постановка задачи, направленной на решение приоритетных проблем заказчика;
- работа в команде от 2 до 6 человек;
- применение современных подходов к проектированию и программированию;
- применение средств централизованного хранения кода;
- тестирование программного кода на уровне отдельных модулей (классов) и на уровне системы.

Процесс формирования готовности к выполнению перечисленных компетенций должен осуществляться в специально созданной среде и условиях работы. В качестве такой среды автором используется портал Online-поддержки учебного процесса ITEM74.RU, реализованный на базе свободно распространяемой CMS Moodle [2]. Данный портал представляет собой целый комплекс средств, предназначенных для дистанционного обучения, и ориентирован на различные образовательные активности. Среди наиболее используемых можно выделить:

- форумы;
- ссылки на ресурсы;
- задания;
- базы знаний (Wiki-страницы);
- тесты;
- опросы.

На портале создан отдельный раздел, в котором сгруппированы необходимые активности для создания необходимой среды. Студенты могут задавать вопросы в тематических форумах, в которых автор и его помощники из числа студентов старших курсов отвечают на вопросы или поддерживают дискуссию, возникшую, например, в ходе обсуждения лекционного материала.

Важным аспектом применения портала является обеспечение средств хранения документации, кода, и контроль за соблюдением календарных сроков работ, выполняемых в ходе семестра. С этой целью в разделе курсового проектирования созданы образовательные активности, такие как «Задание» и «Базы знаний», а на вспомогательном сервере развернута служба централизованного хранения кода на базе Subversion. С помощью активности «Задание» студенты могут загрузить требуемые документы, описывающие тот или иной аспект создания программной системы, например, концепция проекта или архитектурные решения. Преподаватель может проверить загруженный документ, утвердить или отклонить его, с указанием ошибок, допущенных в документе. С помощью базы знаний, которая создается для каждого студенческого проекта, преподаватель может судить о том, насколько качественно студенты подошли к анализу поставленной задачи, насколько проработана проблема, сформулированная прямо или косвенно в поставленной им задаче.

В прошедшем I семестре 2008/09 уч.г. был проведен эксперимент по соединению в единый производственный цикл двух дисциплин «Технология программирования», читаемая на третьем курсе, и «Теория и практика тестирования программного обеспечения», читаемая на четвертом курсе. Суть эксперимента состояла в том, что вместо классических лабораторных работ за командой четвертого курса, состоящей из 2 человек и играющей роль тестировщика, закреплялся один курсовой проект, выполняемый ко-

мандой третьего курса, состоящий из 2–4 человек. Задача тестировщиков состояла в тестировании создаваемых артефактов третьим курсом, а задача третьего курса заключалась в исправлении указанных тестировщиками ошибок. Вся деятельность контролировалась в реальном режиме времени с помощью системы отслеживания ошибок Mantis, развернутой на портале ИТЕМ74.RU и интегрированной с основной системой учетных записей пользователей сайта.

Вместе с тем, возникает еще ряд вопросов, связанных с мотивацией студентов. Среди этих вопросов, на наш взгляд наиболее значимыми являются:

- заинтересованность студента в качестве конечного результата;
- создание условий для самообразования студента;
- опыт творческой деятельности путем поиска новых интересных проектных решений.

Для решения указанных задач и создания необходимых педагогических условий, автором в течение ряда лет применяется разработанная рейтинговая система «Демократия и ответственность™», которая основана на том, что студентам предоставляется заранее документ-требование, в котором описаны возможные технические, проектные, и программные решения, а также стоимость этих решений в баллах [3]. В ходе семестра у студента всегда есть возможность, реализовав эти решения и обосновав их, на защите курсового проекта, получить определенную сумму баллов. На экзамене по дисциплине «Технология программирования» студенту, опять же в соответствии с заранее известными критериями, необходимо выполнить проектное решение задачи, связанной с проектированием информационной системы. При этом сами задачи являются облегченными версиями реальных проектов, выполненных автором или его дипломниками, или проектов выполненных сторонними организациями, описание которых выложено в свободном доступе. Применение рейтинговой системы позволило создать своего рода рыночные механизмы, когда студент получает «вознаграждение» за проделанный труд. Однако данную систему следует применять в сочетании с другими средствами мотивации, чтобы баллы не стали единственным мотивом деятельности.

Дополнительным средством активизации образовательной деятельности студентов является система самоконтроля, построенная с помощью еженедельных отчетов, представляющая собой форум, в котором каждый студент выкладывает отчет о проделанной работе за неделю в строго установленной форме:

- задачи, решенные на прошедшей неделе;
- задачи, которые не решены на текущей неделе, с указанием причины, которые не позволили выполнить запланированный объем работ;
- задачи, планируемые к реализации на следующей неделе;

- предложения – необходимость дополнительной консультации, проработки каких-то тем на лекции или практических занятиях.

Применение таких отчетов приводит к тому, что студент начинает видеть, как продвигается его работа, учится правильно распределять свое время, учится быть ответственным и самоорганизованным, что является необходимыми качествами будущего профессионала.

Вместе с системой отчетов студентам предоставляется возможность в специальной активности CMS Moodle.org «Технический проект» самостоятельно разбивать свой проект на подзадачи, назначать выполнение задач каждому члену команды, тем самым повышая эффективность работы над проектом.

Таким образом, использование портала является средством реализации контекстного подхода к обучению, который способствует повышению уровня подготовки по специальности, повышению мотивации к образовательной деятельности, формированию профессионально значимых качеств будущего специалиста.

ЛИТЕРАТУРА

1. Вербицкий, А.Л. Активное обучение в высшей школе. Контекстный подход / А.Л. Вербицкий. – М: «Высшая школа», 1991.
2. Course Management System. [Электронный ресурс] — Режим доступа: <http://moodle.org>.
3. Цытович, П.Л.. Применение бизнес модели IT-проектов в обучении студентов IT-специальностей / П.Л. Цытович // Материалы IV всероссийской конференции по проблемам IT-образования. Челябинск, 2005.

СВОБОДНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ ВНЕШКОЛЬНЫХ ЗАНЯТИЙ С ДЕТЬМИ

Якшин М.М.

greycat.na.kor@gmail.com

МГТУ им. Н. Э. Баумана, г. Москва

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

В последние годы актуальными становятся вопросы применения свободного ПО в образовании. Использование свободного ПО в образовании, в частности, в школьной практике, активно изучается и поддерживается на государственном уровне программами внедрения [1]. Тем не менее, кроме школьного образования и воспитания, существует ещё сфера внешкольных

занятий – как правило, творческих. Дети занимаются с преподавателями в детских лагерях, кружках, студиях и т.п. В сфере внешкольных программ, кроме традиционных занятий спортом, танцами, актёрским мастерством, живописью, скульптурой, декоративно-прикладными искусствами, с недавних пор добавились ещё и занятия на компьютере – как правило, тоже с творческим уклоном.

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

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

- занятия с детьми сотрудников базы отдыха «Наука» при Южно-Уральском государственном университете [4].
- занятия с детьми, отдыхающими в детском оздоровительном лагере «Березка» при Южно-Уральском государственном университете [4].
- занятия с детьми во всесоюзном лагере информационных технологий «Страна КОМПЬЮТЕРИЯ».
- прочие разовые занятия по приглашениям в Дворцах детского творчества, школах и т.п.

На основании проведенных работ, можно выделить следующие **типичные классы вопросов и проблем**, возникающих при проведении подобной программы.

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

1.1. Дети – все очень разные, в том числе в отношении получения практических навыков ведения творческой деятельности с помощью компьютера. Особенно при больших размерах групп всегда чувствуется, что часть схватывает материал быстрее и требует идти вперед, а часть – медленнее и требует подольше оставаться на месте. Необходимо структурировать курс таким образом, чтобы внутри одного занятия все обучаемые шли в примерно одинаковом темпе. Опыты варьирования длины занятий показывают, что оптимальная длина занятия играет в этом не последнюю роль и, в зависимости от коллектива, в котором проводятся занятия. По возможности стоит варьировать длину занятий от 40-45 минут (если разброс навыков детей значительный) до полутора-двух часов (в старших группах, где все дети достаточно быстро усваивают материал).

1.2. Один из основных педагогических приемов – «повторение» – далеко не всегда адекватно воспринимается детьми при обучении навыкам работы с ПО. Условно можно разделить детей на тех, кто повторяет за преподавателем, тех, кто повторяет за своими сверстниками и тех, кто ничего

ни за кем не повторяет. Основную проблему представляют как раз такие дети, которые принципиально не хотят повторять то, что им предлагается, переоценивают свои силы и тратят много сил, пытаясь придумать что-то свое, но в итоге ничего не успевают. Если подобных детей немного – то часто помогает персонализация обращений к ним и уделение им лично большего внимания.

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

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

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

Старшая (12–18 лет) – наиболее сложная группа – в ней дети уже с большой долей вероятности проходили какие-то предыдущие занятия и имеют значительный опыт общения с Windows-системами, который может оказывать негативное влияние и вызывать резко отрицательный эмоциональный настрой. На занятиях требуется показать наиболее эффектные и привлекательные особенности предлагаемого свободного ПО (зачастую по сравнению с проприетарными аналогами) – спецэффекты в GIMP, продвинутые возможности Inkscape (тени, градиенты, размытие, фотореалистичные изображения), создание видеоклипов на продвинутом уровне. Определенную сложность представляет и то, что старшие дети, как правило, уже определились с некоторыми жизненными приоритетами и интересами – они могут совпасть или не совпасть с предлагаемыми материалами. Это наиболее открытая часть курса, которая требует от преподавателей максимальной отдачи и эрудиции за рамками обозначенной изначально программы курса.

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

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

2.1. Техническая проблема: зачастую организаторы не в состоянии предоставить отдельные компьютеры/лабораторию для проведения занятий и

приходится проводить занятия на тех же компьютерах, где установлен Windows. Это возможно либо с помощью установки второй ОС (dual boot) на компьютеры (инсталлятор должен иметь возможность быстрого и надежного уменьшения размера разделов с другой системой и инсталляции загрузчика для dual boot), либо с помощью использования Live CD (менее предпочтительный вариант, т.к. создает сложности с сохранением работ детей).

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

2.3. Соседство с другими курсами: часто в организациях, занимающихся проведением подобных курсов для детей, уже есть поставленные на поток занятия и обученные преподаватели, которые работают по собственным программам и используют проприетарное ПО. Достаточно сложно сочетать как просто программы обучения (в том числе, если эти программы эксплуатируются не первый год и дети имеют определенный опыт прохождения по программам в прошлых годах), так и проводить параллели между используемым проприетарным и свободным ПО – так, чтобы дети понимали разницу между ними, понимали, для чего это необходимо и т.п.

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

При всем этом, стоит отметить следующие положительные моменты проведенных занятий.

- Многократно показано, что дети на творческих занятиях с успехом могут использовать в качестве инструментов как проприетарное, так и свободное ПО.

- В случаях, когда детям копировался Live CD для продолжения занятий дома, отклик достаточно высок [4], но наблюдаются определенные проблемы технического плана, связанные с запуском Live CD (в первую очередь из-за низкой компьютерной грамотности у родителей). Предложенная в 2008–2009 годах и реализованная в ALT Linux Children 4.1 схема позволяет существенно снизить прецеденты возникновения этих проблем.

ЛИТЕРАТУРА

1. Распоряжение Правительства Российской Федерации от 18 октября 2007 г. №1447-р.
2. Панюкова, А. А. Создание обучающего курса для детей на базе Linux / А.А. Панюкова // Третья конференция "Свободное программное обеспечение в высшей школе", Переславль, 2–3 февраля 2008 г. – М., ALT Linux: 2008.
3. Панюкова, А.А. Методика проведения учебных занятий с использованием свободно распространяемого программного обеспечения / А.А. Панюкова, М.М. Якшин, Т.А. Панюкова // Роль и место самостоятельной работы студентов в образовательном процессе вуза. Юбилейная региональная научно-методическая конференция (4–6 февраля 2008 г.): Сб. науч. тр. – Челябинск, Издательство ЮурГУ: 2008. – Т. 1.
4. Panyukova, A. Organization and methodics for realization of computer graphics studying using free software / A. Panyukova, M. Yakshin, T. Panyukova // Proceedings of the 10th International Workshop on Computer Science and Information Technologies, Antalya, Turkey, September 15–17, 2008: Сб. науч. тр. – Уфа, Редакционно-издательский комплекс УГАТУ: 2008. – Т. 1. – С. 238.

ИНТЕГРИРОВАННЫЕ ЗАНЯТИЯ ПО ИНФОРМАТИКЕ И ТЕОРИИ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКОЙ СТАТИСТИКЕ С ИСПОЛЬЗОВАНИЕМ СВОБОДНО- РАСПРОСТРАНЯЕМОГО МАТЕМАТИЧЕСКОГО ПАКЕТА «SCILAB»

Мирасов В.Ф., Дударева В.И.

Wild-vadim@yandex.ru

ассистент, кафедра ЭММиС, ЮУрГУ, г. Челябинск

vdvd12@mail.ru

к.т.н., доцент, кафедра ЭММиС, ЮУрГУ, г. Челябинск

В данной статье рассмотрена возможность использования на интегрированных занятиях по информатике и теории вероятностей и математической статистике свободно распространяемого математического пакета «Scilab».

ВВЕДЕНИЕ

В течение последних лет современная школа (в том числе и высшая) переживает период совершенствования образования. За это время изменился и курс информатики. Занимая достойное место среди других образовательных дисциплин, информатика имеет очень интересную и важную особенность – возможность легко интегрироваться и быть важным фактором при получении новых знаний в различных областях.

Интегрированные занятия по информатике, теории вероятностей и математической статистике (и другим предметам) наглядно демонстрируют, что учебные предметы не существуют независимо друг от друга в нашем развивающемся информационном обществе. А знания, полученные на занятиях по одному предмету, можно успешно применять и развивать на занятиях по другому предмету.

«SCILAB» НА ЗАНЯТИЯХ ПО ТЕОРИИ ВЕРОЯТНОСТЕЙ И МАТЕМАТИЧЕСКОЙ СТАТИСТИКЕ

На данный момент существует большое количество программных продуктов, которые могут достаточно эффективно применяться в процессе обучения, в частности на занятиях по теории вероятностей и математической статистике. Однако во многих вузах, имеющих современные компьютерные классы, программно-прикладные средства на занятиях по теории вероятностей и математической статистике используются не так уж часто. Это связано в основном с тем, что программные продукты не имеют лицензии или просто с отсутствием установленных на рабочих местах программных комплексов. В операционной системе ALT Linux существует много программных продуктов, которые могут быть использованы преподавателями при проведении современных занятий с использованием новых информационных технологий.

Рассмотрена возможность использования на занятиях по теории вероятностей и математической статистике математического пакета «Scilab», который распространяется бесплатно, что расширяет возможность применения этой программы.

Широкие возможности пакета позволяют использовать его не только на занятиях по теории вероятностей и математической статистике, но, и на занятиях по другим предметам, в частности, в общем курсе высшей математики, т.к. он позволяет решать задачи линейной алгебры, дифференциальные уравнения и т.д.

На показательное занятие выбрана тема, описанная в [1] «Обработка экспериментальных данных. Метод наименьших квадратов». Студенты часто сталкиваются с проблемой наглядного изображения данных. Данный математический пакет помогает студентам справиться с этой задачей и закрепить знания, полученные в курсе информатики.

Метод наименьших квадратов позволяет по экспериментальным данным подобрать такую аналитическую функцию, которая проходит настолько близко к экспериментальным точкам, насколько это возможно.

В [1] показано, как с помощью данного пакета, обладая экспериментальными данными, можно определить уровень регрессии, рассчитать коэффициент корреляции, подобрать такую функциональную зависимость заданного вида, вычислить коэффициент регрессии. На основе этих знаний

созданы варианты задач, предложенные студентам для самостоятельного решения.

R SYSTEM

R – это язык и пакет для построения графиков и проведения статистических вычислений. Это GNU-проект, аналогичный языку S, разработанному в лабораториях Bell (бывшая AT&T, в настоящее время Lucent Technologies) [2]. Пакет R может рассматриваться как модификация S, но в нем есть и много принципиальных отличий.

Программа R предлагает пользователю широкий диапазон статистических (линейное и нелинейное моделирование, классические статистические тесты, анализ временных рядов, классификация, кластерный анализ и пр.) и графический инструментарий. Она является легко расширяемой. Язык S можно рассматривать как хорошее средство для решения статистических задач, а R обеспечивает решение таких задач с помощью свободно-распространяемого программного обеспечения.

Одной из основных возможностей пакета R являются простота его использования, предоставление хорошо оформленных графиков с применением всей необходимой математической символики и формул.

Программа R является свободно-распространяемой и ее можно свободно скачать согласно открытой лицензии GNU. Этот пакет может быть откомпилирован и запущен под широким диапазоном UNIX-систем (например, Free BSD и Linux), а также Windows и MacOS.

R является интегрированным пакетом для анализа данных, проведения расчетов и графической иллюстрации проведенных расчетов. Среди всех возможностей пакета основными являются [2]:

- эффективная обработка и хранение данных;
- набор операторов для операций над массивами и матрицами;
- комплексный набор инструментов для анализа данных;
- графические инструменты для анализа данных и отображения результатов;
- простой в обращении язык программирования (называемый S), включающий в себя условные операторы, циклы, определяемые пользователем рекурсивные процедуры, функции ввода и вывода данных (заметим, что большинство прикладных функций в системе R написаны на языке S).

Выводы

Использование пакетов прикладных программ на занятиях по теории вероятностей и математической статистике стимулирует самостоятельную работу студентов, что является одной из основных задач высшей школы.

ЛИТЕРАТУРА

1. Алексеев, Е.Р. Scilab:Решение инженерных и математических задач / Е.Р.Алексеев, О.В.Чеснокова, Е.А. Рудченко. – М. : ALT Linux; БИНОМ. Лаборатория знаний, 2008. – 260 с.
2. R Project for Statistical Computing. <http://www.r-project.org/>

