

ALT Linux
Unigine
TLUG (Tomsk Linux User Group)

Всероссийская научно-практическая конференция
Свободное программное обеспечение:
разработка и внедрение

Томск, 17–18 мая 2008 года

Тезисы докладов

Томск
Издательство ИОА СО РАН
2008

УДК 004.413
С25

С25 Свободное программное обеспечение: разработка и внедрение: Тезисы докладов Всероссийской научно-практической конференции. — М.; Томск: Издательство ИОА СО РАН, 2008. — 100 с. : ил.

ISBN 978-5-94458-086-6

В книге собраны тезисы докладов, одобренных Программным комитетом первой конференции «Свободное программное обеспечение: разработка и внедрение» (Томск, 17–18 мая 2008). Тезисы печатаются на основе электронных форм, представленных авторами, которые несут ответственность за содержание и оформление текста.

ISBN: 978-5-94458-086-6

© Коллектив авторов, 2008

Программа конференции

17 мая

09.00–09.45: Регистрация в холле гостиницы «Рубин», кофе

09.45–10.00: Вступительное слово Оргкомитета. Открытие конференции

Общие вопросы развития СПО

10.00–12.00

10.00–10.30: А. В. Смирнов

СПО в России: с просёлка на большую дорогу.

Трудности масштабирования 8

10.30–11.00: А. Ю. Запрягаев, Д. Шергин

Использование СПО при разработке Unigine 11

11.00–11.30: Д. А. Лихачёв, Д. Н. Бараксанов, Е. В. Лапин, А. Лёвин

Использование СПО в работе команды веб-разработчиков . 14

11.30–12.00: М. А. Гусаров

Менеджмент крупных открытых проектов. ALT Linux

Sisyphus: проблемы и перспективы —

Внедрение СПО**12.00–14.00**

- 12.00–12.30: М. Е. Кривушин
 Централизованное управление системами на базе GNU/Linux 17
- 12.30–13.30: Обед
- 13.30–14.00: А. Я. Чичак
 Миграция коммерческих структур на СПО. Проблемы и
 способы их решения 19

СПО в высшей школе**14.00–15.30**

- 14.00–14.30: Ф. Е. Татарский, А. А. Князева
 Опыт использования СПО для обучения программистов в
 ВУЗе 21
- 14.30–15.00: П. Г. Губин
 Внедрение СПО в учебный процесс кафедры ПрЭ ТУСУР . —
- 15.00–15.30: С. Н. Постников
 Созрел ли АЛТ Linux для высшей школы? —
- 16.00–17.00: Переход в Институт Оптики Атмосферы, кофе

Круглый стол по внедрению СПО в школах**17.00–20.30**

- 17.00–17.30: В. Д. Корнеев
 Особенности внедрения СПО в школы Томской области ... —
- 17.30–17.50: Е. Г. Пьяных
 Об опыте педагогического университета по повышению
 квалификации в области СПО различных категорий
 работников образования 26
- 17.50–18.10: С. В. Кучин
 Linux глазами сельского учителя и его учеников 28
- 18.10–18.30: В. Б. Смирнов
 Сопровождение сервера в школе 30

18.30–18.50:	Л. А. Татарникова	
	Переход на СПО: методическое обеспечение	32
18.50–19.10:	Д. В. Ефанов	
	Организация школьного сервера на базе дистрибутива Debian GNU/Linux	35
19.10–19.20:	Перерыв	
19.20–20.30:	Дискуссия, ведущий — А.В. Смирнов (Москва, ALT Linux)	

18 мая

09.00–09.30: Стендовая секция

СПО в высшей школе (продолжение) 09.30–10.00

09.30–10.00:	В. М. Вымятин	
	СПО в ВУЗе: необходимое, но не достаточное	37

Разработка СПО 10.00–16.00

10.00–10.30:	С. И. Иевлев	
	ALT Linux Installer	39
10.30–11.00:	М. С. Пожидаев	
	Обзор информационной системы для торговли и складского учёта Indocs	41
11.00–11.30:	Кофе, стендовая секция	
11.30–12.00:	С. В. Подушкин	
	ОфисМастер — сервер инфраструктуры для «чайников» . . .	42
12.00–12.30:	Д. В. Левин	
	hasher: технология безопасной сборки пакетов	45
12.30–13.00:	А. Г. Боковой	
	Новая Samba	49
13.00–14.00:	Обед	

14.00–14.30:	О. Л. Садов	
	Международная инициатива Scientific Linux и проект НауLinux для школьного образования	51
14.00–15.00:	М. А. Шигорин	
	Linux Terminal Server Project	55
15.00–15.30:	А. Варламов, С. Забелин, С. М. Мишура	
	Apache Harmony: разработка Java-платформы как свободного ПО	58
15.30–16.00:	М. Ю. Фурсов	
	Apache Harmony: компоненты для переиспользования	—
16.00–16.30:	Кофе, стендовая секция	

**СПО в научных исследованиях
16.30–18.00**

16.30–17.00:	А. П. Щербаков	
	Разработка макета автономного робота на основе платформы GNU/Linux	59
17.00–17.20:	Г. Э. Куликов	
	Создание автоматизированной системы сбора и обработки данных физического эксперимента на основе платформы GNU/Linux	62
17.20–17.40:	А. И. Привезенцев	
	Разработка модуля «Мета+» информационной системы «Молекулярная спектроскопия» для формирования онтологического описания	64
17.40–18.00:	А. Ю. Ахлестин	
	Промежуточное программное обеспечение для создания научных информационных систем	67
18.30–19.00:	Заккрытие конференции	

Стендовая секция

М. С. Пожидаев	
Обзор компонентов и возможностей средств для работы без зрительного контроля в дистрибутивах ALT Linux .	69

Е. В. Лапин, Д. Н. Бараксанов, Д. А. Лихачёв, А. Лёвин	
Опыт внедрения открытой CMS	70
С. В. Кратов	
Организация взаимодействия пользователей распределённых систем на базе свободного ПО	73
В. В. Сергеев, С. Ю. Коростелев, С. Г. Псахье	
Приложение «Analyzer 3D» для обработки и анализа результатов моделирования методом частиц	76
Ю. А. Поплавский, А. П. Щербаков, Г. Э. Куликов	
Использование распределённой сетевой архитектуры и платформы GNU/Linux в непрерывных спектрофотометрических анализаторах	79
Д. Д. Моисеенко	
Стохастические клеточные автоматы: моделирование роста дендритных структур	—
Н. А. Лаврентьев	
Климатические модели в Интернет-доступной ИВС	82
А. А. Вичугова	
Опыт использования СПО в учебном процессе технического ВУЗа	84
Д. Н. Бараксанов, Е. В. Лапин, Д. А. Лихачёв, А. Лёвин	
Свободные решения для построения веб-пространства университета	88
И. А. Хахаев	
Опыт эксплуатации ALT Linux Desktop Edu	91
А. Н. Гороховский	
Internet-проект «Assistant» — самопроверка расчётно-технических работ студентов	93
С. В. Косаченко	
Опыт использования СПО в школьном образовании	96
О. Н. Грибан	
Мультимедиа технологии в обучении истории: свободный взгляд	—

А. В. Смирнов

Москва, ALT Linux

Проект: Разработка концепции развития СПО в России

http://www.altlinux.ru/community_news/spo_conception_in_russia.html

СПО в России: с просёлка на большую дорогу

Использование свободного программного обеспечения в России распространено значительно шире, чем это кажется на первый взгляд, достаточно обратиться к статистике использования ПО в российском сегменте Интернет, где более 90% веб-серверов занимает Apache и более 80% ОС — это Linux или FreeBSD. В то же время разработка свободных программ и их внедрение оставались долгие годы на периферии IT-индустрии. Сейчас мы наблюдаем резкий рост интереса к СПО как со стороны бизнеса, так и со стороны государства. Я полагаю, что этот интерес вызван не только ужесточением авторско-правового законодательства и необходимостью легализации ПО, но и пониманием, что это основной путь приобщения к современным технологиям и включения в международное сообщество разработчиков, путь изменения в свою пользу сложившегося международного разделения труда.

Отдельные работы, которые велись по заказу Минэкономразвития с 2003 года, отдельные тендеры, по которым поставлялись школьные компьютеры с предустановленным Linux, разовые внедрения СПО в небольших подразделениях госучреждений (прежде всего на серверах) переросли в систематический интерес со стороны государства. В 2006 году профильное министерство — Мининформсвязи — рассмотрело вопросы развития СПО в России на Коллегии, была создана Рабочая Группа Мининформсвязи по Свободному ПО, которая подготовила свои рекомендации. На их базе была разработана «Концепция развития разработки и использования свободного программного обеспечения в Российской Федерации». Это первый официальный документ в российской нормативной практике, вводящий понятие «Свободного программного обеспечения» (СПО) и другие базовые понятия. В концепции формулируется «государственная политика в области разработки и использования программ для ЭВМ и принци-

пы развития разработки и использования СПО», ставятся связанные с СПО правовые вопросы, рассматриваются «инфраструктура разработки и использования СПО», организация повышения квалификации, популяризация СПО, поддержка проектов СПО и некоторые другие вопросы. В декабре 2007 года Концепция была одобрена Коллегией Мининформсвязи. За эти годы удалось в значительной мере «расчистить пространство», решив ряд юридических и организационных препятствий к продвижению СПО. В качестве примера приведу отмену в 2005 году Постановления Правительства от 1999 года, которое предписывало госзаказчикам требовать от разработчиков передачи исключительных авторских имущественных прав, это требование, как правило, не могло быть выполнено разработчиками свободных программ.

Заметным шагом было также Распоряжение Председателя Правительства РФ о подготовке ПСПО — комплекта свободных программ для школ — наряду с поставкой в школы комплектов «Первой помощи». Одним из трёх пилотных регионов по внедрению ПСПО является Томская область. Хочу обратить внимание на мотивировку, сформулированную Дмитрием Медведевым, тогда ещё Первым вице-премьером, осенью 2007 года: речь идёт не о снижении издержек на лицензирование, а об обеспечении независимости государства в сфере информационных технологий.

На этой волне государственного интереса к свободному программному обеспечению провозглашался и лозунг создания «Отечественной операционной системы». Прежде, чем давать ему какие-либо оценки, давайте разберёмся, какие задачи он пытается решить, и какое ПО можно было бы назвать отечественным. Вряд ли стоит связывать отечественную принадлежность ПО с гражданством разработчиков, с местом регистрации фирмы или с гражданством её владельцев: разработанное российскими оффшорными программистами ПО вряд ли можно считать отечественным, а владельцы фирмы могут и поменяться. Если речь идёт о доступности программного обеспечения для государства, то должны быть обеспечены как минимум наличие экземпляра программы (закон об обязательном экземпляре пока действует), наличие достаточных для обеспечения жизненного цикла авторских имущественных прав и доступность сборочной среды. Последнее лучше всего обеспечить сборкой разработанного по госзаказу ПО в контролируемой государством эталонной сборочной среде (возможно, не единственной).

Периферийная роль свободного ПО в течение значительного времени обеспечивала элитарность разработчиков, «чистоту рядов» и вовлечённость заметного их числа в международные проекты разработки различных свободных программ. Начавшаяся быстрая коммерциализация, и тем более проявленный со стороны государства интерес, могут быстро изменить картину, и важно выбрать верную стратегию развития. Распространены три подхода. Первый состоит в распространении готовых зарубежных продуктов, желательно без изменений или с минимальными доработками. Это путь организации дистрибьюции, при котором любая собственная разработка становится обузой. Очевидно, что такой путь не способствует ни развитию собственной высокотехнологичной разработки, ни созданию ответственной техподдержки. Второй — изоляционистский — подход в своей крайней форме состоит в попытке полностью самостоятельной разработки, в этом случае большинство преимуществ свободного ПО теряется, да и ресурсов для такой разработки, как правило, нет. В более мягкой форме изоляционистского подхода международные проекты разработки СПО рассматриваются исключительно как источник кода, в то время как собственные разработки не отдаются в апстрим. Но при таком подходе придётся самостоятельно переносить все свои изменения и доработки во всё новые и новые версии продукта, и по любым вопросам рассчитывать на помощь международной команды не приходится. Примеры такого подхода есть в России, в Китае и ряде других стран.

Наиболее эффективным представляется подход, ориентированный на тесное взаимодействие с международными командами разработки свободных программ, с разворачиванием собственной инфраструктуры разработки и сборки. При всех преимуществах международного сотрудничества обеспечивается развитие своей инфраструктуры разработки, что в свою очередь обеспечивает возможность качественной поддержки и развития смежных бизнесов.

А. Ю. Запрягаев, Д. Шергин

Томск, Unigine Corp.

Проект: Unigine

<http://unigine.com/>

Использование СПО при разработке Unigine

Аннотация

Прозрачное использование кросс-платформенных-инструментов (SCons, Python, Vim) и библиотек. Параллельное использование свободных решений с закрытым исходным кодом: MS Visual C ++ Express Edition, MS Platform SDK. Эффективная разработка с использованием GNU toolchain: GCC, GDB, distcc, ccache. Вопросы обеспечения лицензионной чистоты проприетарного продукта, при разработке которого используется СПО. Опыт взаимодействия с командами разработки драйверов современного 3D «железа» под GNU/Linux. Использование платформи-независимых решений для управления проектом: Subversion и Trac.

Unigine — это высокотехнологичный кросс-платформенный 3D-движок для систем виртуальной реальности, включающий в себя рендер, модуль физической симуляции, интерпретатор языка UnigineScript, модуль GUI и звуковую подсистему. Основные сферы применения — современные игры и системы виртуальной реальности. Разработка ведётся с 2004 года, проект имеет open source корни, но в настоящее время является проприетарным по ряду причин.

Большинство компаний, разрабатывающих проприетарное ПО, используют платные решения для разработки. Но платное — не всегда означает лучшее, даже если проект ориентирован только на Windows, не говоря уже о кросс-платформенности.

Процесс разработки

Целевыми платформами Unigine на текущий момент являются 32- и 64-битные версии MS Windows и GNU/Linux. Разработка, в основном, ведётся из Linux-окружения. Используемые компиляторы — GCC (Linux) и MS Visual Studio Express Edition + MS Platform SDK (Windows), распространяемые бесплатно. В качестве системы сборки используется кросс-платформенный SCons. Для автоматизации рутинных операций широко используется Python. Не используется никаких платформозависимых IDE, только редакторы с поддержкой plain

text (в основном — VIM, отлично работающий и в Linux, и в Windows). Альтернативные компиляторы не устраивают по ряду причин — например, MinGW не имеет поддержки DirectX 10 из-за инертности разработчиков, а Intel Compiler не даёт работающих сборок под Linux. Огромным плюсом разработки в GNU/Linux-окружении является наличие ccache (системы кеширования объектных файлов), ускоряющего пересборку проекта на порядок. Ещё один используемый инструмент для ускорения сборки — distcc (система распределённой компиляции, доступна только в Linux). Для отладки используются GDB, Visual Studio IDE и самый мощный инструмент — голова разработчика.

Профилирование производительности производится, в основном, в Windows из-за наличия совместимых удобных инструментов, включая специфичные для графических приложений: NVIDIANGPUPerfHUD, AMD GPU PerfStudio и AMD Shader Analyzer (которые тоже доступны бесплатно). Изредка используется gprof, но его применение ограничено из-за низкой скорости работы с крупными проектами и не очень удобных текстовых отчётов.

Для разработки используется выделенный сервер, предоставляющий доступ к одним и тем же файлам по NFS и SMB, что при наличии сети 1 Гб/с вполне комфортно для разработчика. Оттестированные изменения фиксируются в системе контроля версий Subversion. Официальные сборки создаются на одной машине под управлением Linux, Windows работает на ней в эмуляторе (VirtualBox). Для автоматизации процесса финальной сборки SDK, предоставляемого клиентам, создан набор скриптов на Python и Bash.

Для организации совместной работы группы разработчиков используется open source система управления проектами Trac (web-based), которая содержит багтрекер, трекер задач, wiki, и хорошо интегрируется с SVN. Документация частично генерируется с помощью Doxygen, в целом же для документирования разработана эффективная система на основе Python, PHP и XML/XSLT.

Открытые библиотеки

Unigine использует ряд открытых библиотек: zlib, pnglib, jpeglib, freetype, oggvorbis, theora, для каждой из которых написаны сборочные правила для SCons, чтобы избежать путаницы в разнообразии систем сборки и конфигурирования. Лицензии каждой из этих биб-

лиотек совместимы с проприетарным ПО, этому вопросу уделяется пристальное внимание.

Драйверы под Linux

Не секрет, что одной из основных проблем Linux признаётся ситуация с драйверами современных видеокарт. К счастью, недавно Linux-драйвер от AMD (он сильно отставал по качеству от NVIDIA) был серьёзно переработан (код, касающийся OpenGL, синхронизирован с Windows-версией) и сейчас вполне пригоден к использованию. К сожалению, у AMD и NVIDIA поддержка Linux до сих пор производится по «остаточному» принципу, т. к. основные ресурсы брошены на поддержку Vista и DirectX 10. Несмотря на это, нужно отметить, что темпы исправления ошибок в драйверах в последнее время значительно возросли, большая часть проблем уже в прошлом.

Ещё одной проблемой в Linux являются современные звуковые карты от Creative Labs, большая часть возможностей которых недоступна пользователям из-за отсутствия нормальных драйверов и спецификаций. К счастью, недавно производитель открыл доступ к документации на флагманскую линейку X-Fi для разработчиков OSS и ALSA, скоро ожидаются драйверы с поддержкой 3D для Linux.

Заключение

Опыт разработки Unigine показывает, что использование свободного программного обеспечения позволяет создавать крупные кроссплатформенные проекты ничуть не хуже, чем с помощью дорогих проприетарных решений. Компания Unigine Corp. относится к сообществу open source с признательностью, в меру возможностей внося свой вклад в общее дело. Так, предыдущие версии Unigine доступны под GPL, интенсивное взаимодействие с производителями аппаратных средств помогает улучшать драйверы для Linux, сотрудники компании активно участвуют в продвижении свободного программного обеспечения.

Д. А. Лихачёв, Д. Н. Бараксанов, Е. В. Лапин, А. Лёвин
Томск, ТУСУР

Использование СПО в работе команды веб-разработчиков

Аннотация

Основные задачи (контроль версий, планирование рабочего времени, среда разработки) и средства построения интегрированного рабочего окружения веб-разработчика.

Исходные данные

Команда разработчиков: 4 человека.

Разрабатываются веб-приложения на базе J2EE технологий.

Одновременно ведётся 2–4 проекта. Порядка 5–10 проектов постоянно сопровождаются. В проектах часто изменяются требования.

Существующие проблемы

В конце 2006 года команда столкнулась с проблемой лицензирования используемого ПО. Было принято решение о полном переходе на СПО. В то же время назрела необходимость применения методологии, позволяющей систематизировать процесс разработки ПО. Анализ существующих методологий разработки ПО показал, что в исходных условиях наиболее применимыми являются «гибкие» методы разработки [6].

Большинство гибких методологий нацелены на минимизацию рисков путём сведения разработки к серии коротких циклов, называемых итерациями, которые обычно длятся одну-две недели. Каждая итерация сама по себе выглядит как программный проект в миниатюре и включает все задачи, необходимые для выдачи прироста по функциональности: планирование, анализ требований, проектирование, кодирование, тестирование и документирование. Хотя отдельная итерация, как правило, недостаточна для выпуска новой версии продукта, подразумевается, что гибкий программный проект готов к выпуску в конце каждой итерации. По окончании каждой итерации команда выполняет переоценку приоритетов разработки [1].

Таким образом, перед командой встали следующие задачи:

- организация планирования разработки;
- контроль версий;
- накопление и обмен знаниями внутри команды;
- выбор среды разработки.

Организация планирования разработки

Планирование является одним из ключевых этапов жизненного цикла проекта [2]. Регулярное планирование позволяет отслеживать текущее состояние проекта и прогнозировать сроки его завершения.

В качестве инструмента, обеспечивающего поддержку планирования, используется Redmine (<http://www.redmine.org>). Redmine позволяет отслеживать задачи, вести учёт фактически потраченного времени по каждому проекту, планировать итерации, обеспечивать интерактивную обратную связь с пользователями.

Контроль версий

Если в команде не используется система управления версиями, то в исходных кодах проектов отсутствует порядок; на жёстком диске участников команды появляются директории вида `new`, `new2`, `old`, `very_old`. Параллельная работа членов команды становится невозможной из-за сложности слияния внесённых изменений.

Система управления версиями — программное обеспечение для облегчения работы с изменяющейся информацией. Система позволяет хранить несколько версий одного и того же документа и, при необходимости, возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение [3].

В качестве системы управления была выбрана Subversion (<http://subversion.tigris.org>). Система обладает хорошей инструментальной поддержкой и тесно интегрируется с другими системами и средами разработки.

Накопление и обмен знаниями внутри команды

Классическим примером реализации базы знаний является вики. Вики — это набор связанных веб-страниц, предназначенных для быстрого коллективного доступа и лёгкой модификации. Для написания

вики-страниц используется простой синтаксис, позволяющий легко и быстро структурировать информацию [5].

В команде используется DocuWiki (<http://wiki.splitbrain.org/wiki:dokuwiki>) в качестве основного вики-движка. Для некоторых проектов используется модуль вики, интегрированный в Redmine.

Интегрированная среда разработки

Обычно среда разработки включает в себя текстовый редактор, компилятор и/или интерпретатор, средства автоматизации сборки и отладчик [4].

В команде используется IDE Eclipse (<http://www.eclipse.org>). Eclipse хорошо интегрируется с различными средствами поддержки разработки (Subversion, Trac) при помощи модулей расширения.

Вывод

Процесс перехода на использование СПО в веб-разработке может проходить двумя путями: использование рассмотренных решений в окружении Windows либо осуществление полного перехода на СПО. В последнем случае можно обеспечить полный цикл поддержки разработки веб-приложений с нулевыми затратами на программное обеспечение. В заключении хотелось бы отметить, что команда решает все повседневные и сопутствующие задачи исключительно используя СПО.

Литература

- [1] Гибкая методология разработки.
- [2] Жизненный цикл проекта.
- [3] Система управления версиями.
- [4] Среда разработки программного обеспечения.
- [5] Wiki.
- [6] Амблер С. *Гибкие технологии: экстремальное программирование и унифицированный процесс разработки*. СПб:Питер, 2005.

М. Е. Кривушин

Томск, ЗАО «СОЮЗ Т»

Проект: Сеть ЗАО «СОЮЗ Т» <http://deepwalker.blogspot.com>

Централизованное управление системами на базе GNU/Linux

Введение

Современный уровень развития свободного программного обеспечения (СПО) позволяет использовать системы на базе GNU/Linux в корпоративном секторе. С 2006 года в ЗАО «СОЮЗ Т» применяется СПО, с 2007 года на рабочих местах используется ОС Linux. Сеть ЗАО «СОЮЗ Т» имеет централизованную архитектуру с центром в Томске и множеством филиалов. Задачи, решаемые сетью, — это торговые операции на терминальных серверах под управлением MS Windows и обмен информацией. Сеть должна предоставлять современный уровень безопасности и доступности. Технологии, обеспечивающие работу сети Linux-машин, должны интегрироваться с имеющейся инфраструктурой MS Active Directory, что необходимо для совместной работы в полученной неоднородной среде. Инфраструктура управления сетью базируется на Kerberos, LDAP и Cfengine.

Kerberos, аутентификация

Система Kerberos разработана в Массачусетском технологическом институте для проекта Athena. Kerberos построен на криптографии с симметричным ключом, разрабатывался в основном для клиент-серверной модели, предоставляет взаимную аутентификацию (и сервер, и клиент проверяют друг друга). Инфраструктура требует третьей доверенной стороны — KDC (Key Distribution Center — центр распределения ключей). Kerberos является системой, проверенной временем, обладает хорошим дизайном. Kerberos позволяет аутентифицироваться со своими ключами в дружественных сферах, в частности, пользователь из сферы Kerberos может получить доступ к машинам в домене Active Directory, для чего достаточно настроить дружественные отношения и отображение имён.

Критичными для Kerberos являются синхронизация времени и корректность DNS (обратное разрешение имён). Практически 90 % любых проблем связаны с этими причинами.

В ЗАО «СОЮЗ Т» инфраструктура Kerberos состоит из двух KDC, расположенных в Томске, и их клиентов.

LDAP, хранение информации

LDAP — Lightweight Directory Access Protocol, протокол доступа к каталогу. Каталог LDAP представляет собой базу данных, информация в которой организована в виде дерева. LDAP позволяет искать по каталогу, изменять, добавлять и удалять записи. LDAP хорошо подходит для хранения разнородной информации, записей, у которых не все атрибуты обязательны для заполнения или могут включаться несколько раз. Например, у группы может быть сколько угодно вхождений MemberUid, или dNSDomain может включать в себя сколько угодно записей MXRecord. Как видно из примеров, LDAP — идеальное решение для хранения различной информации, необходимой для правильного функционирования сети. Также плюсом является единая репликация сервисов, чья информация хранится в каталоге LDAP. Операции с LDAP хорошо скриптуются, и конфигурация разных сервисов становится более единообразной — добавление нового пользователя, DNS-записи или учетной записи для IP-телефонии очень похожи.

Конфигурирование

Есть два типа инструментов для конфигурирования нескольких машин: инструменты, посылающие пользовательский ввод одновременно нескольким машинам, и системы с поддержкой конфигурации. К первому типу относятся ClusterSSH и аналоги, ко второму — Cfengine, Puppet, Bcfg2. Системы с поддержкой конфигурации через определённые промежутки времени проверяют все условия, на которые настроены. Условием может быть наличие определенной строки в файле, разрешение на доступ, состояние процесса (запущен или нет), установлены ли необходимые пакеты ПО и т. д. Если условие не соблюдено, то корректируется состояние системы до нужного. В ЗАО «СОЮЗ Т» применяются ClusterSSH, Cfengine и скрипты.

Скрипты предназначены для извлечения конфигурационных файлов из каталога LDAP и контроля состояния пакетов через apt.

Литература

- [1] *Heimdal* Официальный сайт Heimdal <http://www.h51.org/>
- [2] *OpenLDAP* Официальный сайт OpenLDAP <http://www.openldap.org/>
- [3] *Cfengine* Официальный сайт Cfengine <http://cfengine.org/>

А. Я. Чичак

Томск, ООО «С-Плюс»

Миграция коммерческих структур на СПО. Проблемы и способы их решения

В процессе миграции на свободное ПО любой, не только коммерческой организации, возникают проблемы:

1. организационные;
2. психологические;
3. технические.

Организационные проблемы связаны с отношением руководства к необходимости миграции и ожидаемым результатам. Психологические — это в первую очередь восприятие сотрудниками организации нововведений. Самая главная техническая проблема состоит в том, чтобы провести миграцию, не нарушая производственных процессов организации.

В коммерческих организациях проблема непрерывной работы предприятия стоит более остро, но и решение в ряде случаев происходит проще, например, при довольно жёсткой позиции руководства сотрудники осваивают новую систему, не проявляя при этом явного негатива.

Проведение миграции можно разбить на четыре этапа.

Первый этап (переговорный). На этом этапе выясняется круг задач, решаемых в организации при помощи компьютерной техники,

разъясняются возможности СПО и его ограничения. Особо убедительны доводы о малой подверженности систем вирусам, большей возможности контроля прав доступа, например, к сменным носителям, лучшая защита информации. Продолжительность этапа совершенно непредсказуема, может достигать и 6–8 месяцев.

Вторым этапом является составление максимально полного списка ПО, используемого в организации. Наиболее эффективный способ анализа используемого ПО — анкетирование сотрудников, в процессе которого каждый сотрудник заполняет анкету, в которой указывает всё используемое программное обеспечение на рабочем месте, не только участвующее в производственных процессах. Обобщённый список реально необходимого ПО утверждается руководством организации. При правильной организации сбора и анализа информации максимальная продолжительность этапа — 1 неделя.

Следующий этап — составление плана миграции. Возможны два варианта: постепенное замещение компонент системы на аналоги из СПО и одномоментный. Варианты постепенного замещения — установка OpenOffice, работа с сервером в терминальном режиме из Windows, используя NX, тонкий клиент с загрузкой по сети. Довольно быстрый этап, в среднем занимает 2–3 рабочих дня вместе с согласованием плана работ.

И, наконец, собственно внедрение, консультирование персонала по вопросам использования СПО, демонстрация аналогов ранее использовавшихся программ, сопровождение установленных систем. Средняя длительность этого этапа (не считая сопровождения) практически не зависит от количества рабочих станций и составляет около 1 месяца. Как показывает опыт миграций, основные проблемы возникают с унаследованным ПО, предназначенным для предоставления отчётности в налоговую инспекцию и пенсионный фонд, разнообразными системами «банк-клиент». Для решения этих проблем возможно предложить установку компьютера с лицензионной MS Windows и работу с ним в терминальном режиме, но и этот вариант в последнее время становится проблематичным, так как многие системы «банк-клиент» под MS Windows Vista работают хуже, чем под Wine@Etersoft, а MS Windows XP купить не всегда представляется возможным из-за сложившегося дефицита.

Ф. Е. Татарский, А. А. Князева

Томск,
Томский Политехнический Университет

Опыт использования СПО для обучения программистов в ВУЗе

Аннотация

В докладе рассказывается о свободных и открытых программных продуктах, технологиях и методическом обеспечении, которые используются в подготовке инженеров по специальности 230105 «Программное обеспечение вычислительной техники и автоматизированных систем» на кафедре АиКС АВТФ ТПУ. Освещаются вопросы развёртывания рабочей среды (бездисковые терминалы X Windows), обучения различным аспектам программирования, использования технологии DocBook для подготовки методического обеспечения, CVS для хранения исходных кодов работ студентов и для поддержки «сквозного обучения», веб-технологий для организации дистанционного обучения и обратной связи.

Учебные планы для направления 230105 «Программное обеспечение вычислительной техники и автоматизированных систем», подготовка инженеров по которому ведётся на кафедре АиКС АВТФ ТПУ[1], содержат различные специальные дисциплины, такие как «Объектно-ориентированное программирование», «Операционные системы», «Базы данных», «Технология разработки программного обеспечения» и многие другие.

Очевидно, что поддержка этих дисциплин требует использования множества программ. До недавнего времени вопрос с выбором программного обеспечения решался на усмотрение преподавателя. Вопрос с лицензированием ПО просто не стоял. В последнее время ситуация сильно изменилась. На учебных компьютерах устанавливается только то программное обеспечение, на которое в ТПУ имеются лицензии. Лицензии закупаются централизованно, на основании «усреднённых» заявок от обеспечивающих кафедр. Выбор лицензионного ПО при таком подходе сильно ограничен и потребностей учебного процесса не покрывает.

Таким образом, использование свободного, открытого или бесплатного программного обеспечения не является вопросом моды или идеологии. Для нас это насущная необходимость. Находясь в такой ситуации мы приняли решение просто отказаться от требующего лицензирования программного обеспечения и постараться обеспечить учебный процесс, оставаясь в рамках свободного и открытого ПО. Далее перечислено то ПО, которое мы уже используем в учебном процессе.

Операционная система

Так сложилось, что мы не используем Linux. В качестве операционной системы мы выбрали систему FreeBSD[6], так как у нас уже имеется большой положительный опыт её использования для решения различных задач. Однако ту же самую функциональность можно получить и в системе Linux. Более того, когда система уже настроена и работает, разницу между Linux и FreeBSD сможет заметить не всякий пользователь.

На нашей кафедре уже давно используется платная, несвободная операционная система. Написано много методических материалов, ориентированных именно на неё. В таких условиях невозможно разом заменить её на FreeBSD, не разрушив сложившуюся инфраструктуру. Мы нашли другое решение.

Наши рабочие станции могут загружать FreeBSD по сети (по протоколу PXE) и работать в режиме бездисковых терминалов. Этот подход даёт нам возможность запускать FreeBSD на рабочих станциях, не внося никакие изменения на жёсткий диск компьютера.

Такую систему можно использовать и как бездисковую рабочую станцию, когда всё ПО запускается на ней самой, и вместе с сервером терминалов, когда все пользователи запускают программы на одном сервере терминалов, а рабочая станция отображает только графический интерфейс. В качестве графического интерфейса используется известная система XOrg[13]. Оба подхода имеют свои преимущества и недостатки и могут использоваться параллельно.

Студентам-заочникам, которые работают дома, мы предлагаем использовать так называемый «живой диск» (LiveCD). Хорошо зарекомендовала себя сборка Frenzy[7], базирующаяся на дистрибутиве FreeBSD 6.

Используемое в учебном процессе программное обеспечение

В качестве первого и базового языка программирования для наших студентов мы используем C++. Коллекция компиляторов GCC входит в базовую конфигурацию FreeBSD. Кроме C++ в учебном процессе требуются языки Perl[11] и Lisp[3], которые так же имеются и могут быть использованы во FreeBSD. Кроме того, студентам необходима СУБД MySQL[10] и Веб сервер Apache[2].

Программное обеспечение для поддержки учебного процесса

Учебный процесс требует подготовки большого количества текстовых материалов, таких как методические пособия, рабочие программы, лабораторные задания, задания на курсовые работы.

Поскольку с течением времени меняются потребности учебного процесса, используемые технологии да и мы сами набираемся опыта, текстовые материалы часто меняются. Кроме того, у нас имеется необходимость поддерживать две версии одних и тех же документов. Основная форма — тексты в электронном виде на сайте. Эти тексты доступны всем студентам, в том числе и заочникам. На сайте всегда находится свежая, актуальная информация. Вторая форма — печатная. Во многих случаях с печатной формой документа работать удобнее чем с электронной. Кроме того, нам необходимо подготавливать печатные рабочие программы для соблюдения формальностей.

Здесь возникают следующие проблемы:

1. Объём документов большой, часть документов редактируется несколькими авторами параллельно. Необходимо синхронизировать правки и выдерживать единое оформление текстов.
2. Часто изменения в текстах появляются уже в процессе выполнения работ студентами. Необходимо поддерживать несколько версий документов и иметь возможность при необходимости откатываться назад.
3. Часть материалов пересекается. Одна и та же глава может быть включена в лекции по двум разным дисциплинам.
4. Необходимо подготавливать параллельно две версии документов — печатную и электронную.

Для решения всех этих проблем мы с успехом используем систему DocBook[5].

Одной из самых больших проблем в обеспечении учебного процесса является нехватка учебного времени. Мы стараемся оптимизировать его. Для это мы, в частности, работаем над «состыковкой» дисциплин, чтобы результаты работы студентов по одной дисциплине были одновременно и входными данными для следующей.

При таком подходе не обойтись без системы контроля версий. Мы выбрали систему CVS[4] и занимаемся её внедрением в учебный процесс. Эта работа ещё не закончена, но трудности с внедрением CVS обусловлены в основном материальными причинами — нет аппаратуры (сервера) для поддержки репозитория. Однако уже сейчас некоторые наши студенты используют CVS для хранения своих дипломных работ.

Почтовые рассылки оказались удобным способом общения со студентами заочной формы обучения. Очень часто бывает так, что многие студенты задают в течении семестра похожие вопросы. Ответы на них отнимают много времени. При использовании почтовой рассылки эти временные расходы сокращаются.

Кроме того, почтовая рассылка имеет стимулирующий эффект. Когда студенты видят, что кто-то из их одногруппников задает вопросы, а значит работает, они сами начинают работать интенсивнее.

Для поддержки почтовых рассылок мы используем традиционную связку sendmail+mailman[8].

Планы на будущее

Выше было перечислено то ПО, которое мы уже в том или ином виде используем. Однако в наши планы входит расширение этого списка.

В рамках своей деятельности мы занимаемся руководством дипломными работами студентов. У нас имеется потребность в ресурсе, на котором можно публиковать неформальные описания актуальных задач, подходящих в качестве тем дипломных работ. Каждое из таких описаний может быть снабжено комментариями как преподавателя, так и студентов. На наш взгляд, по формату такой ресурс ближе всего к блогу.

Реферат является одной из традиционных форм самостоятельной работы студентов. В то же время, совместная работа студентов над

решением задачи может оказать большой положительный эффект на качество обучения. На наш взгляд, Wiki-система может стать отличной площадкой для совместного творчества студентов, склонных к коллективной работе.

В качестве блоговой платформы мы планируем использовать систему WordPress[12]. В качестве wiki-системы — систему MediaWiki[9].

Литература

- [1] ТПУ. Кафедра АиКС.
- [2] The apache software foundation, 2008.
- [3] Association of lisp users, 2008.
- [4] Cvs - open source version control, 2008.
- [5] Docbook.org, 2008.
- [6] The freebsd project, 2008.
- [7] Frenzy, 2008.
- [8] Mailman, the gnu mailing list manager, 2008.
- [9] Mediawiki, 2008.
- [10] Mysql :: The world's most popular open source database:, 2008.
- [11] The perl directory at perl.org, 2008.
- [12] Wordpress - blog tool and weblog platform, 2008.
- [13] X.org foundation, 2008.

Е. Г. Пьяных

Томск,
Томский государственный педагогический университет

Об опыте педагогического университета по повышению квалификации в области СПО различных категорий работников образования

Аннотация

Доклад посвящён обобщению накопленного Томским государственным педагогическим университетом опыта по повышению квалификации различных категорий работников образования в области СПО и сопровождения процесса внедрения СПО в учреждениях системы образования.

В последнее время реализуется большое количество проектов по внедрению свободного программного обеспечения (СПО) в учреждения бюджетной сферы, в том числе, учреждения системы образования. Это требует, в свою очередь, наличия квалифицированных кадров, обладающих навыками работы в СПО.

Начиная с 2006 года, Томский государственный педагогический университет ведёт работы по переводу учебного и административного процесса на свободное программное обеспечение. В это же время ТГПУ осуществляет повышение квалификации различных категорий работников образования в области СПО. В докладе рассмотрены особенности содержания, методики проведения и способы организации образовательных курсов в области СПО. ТГПУ разработаны и апробированы программы повышения квалификации для профессорско-преподавательского состава Вуза, административных кадров ВУЗа, учителей-предметников, учителей информатики, системных администраторов, административных кадров школы.

Обучение различных категорий слушателей работе с СПО ведется в рамках:

1. внутривузовской программы повышения квалификации;
2. областного проекта по внедрению СПО в учреждения образования;
3. федерального проекта по внедрению СПО в учреждения образования;
4. программы Интел «Обучение для будущего».

В настоящее время в общей сложности обучено более 200 человек.

В докладе отмечено, что программы повышения квалификации различных категорий пользователей разрабатывались с учётом их профессиональных потребностей. Так, учебные программы для учителей направлены не только на изучение СПО (офисные технологии, программирование), но и на методику применения СПО в деятельности учителя. Программа повышения квалификации для административных кадров образовательных учреждений в большей степени ориентирована на изучение офисных технологий и подготовку различной документации с использованием СПО.

Базовой целью всех программ повышения квалификации является не столько доскональное изучение тех или иных программных продуктов, сколько общее знакомство с возможностями СПО. Поэтому программы повышения квалификации включают достаточно большой перечень программных продуктов.

Следует отметить, что программы повышения квалификации ориентированы не только на разные профессиональные категории пользователей, но и позволяют учитывать различный уровень владения информационными технологиями. При работе с начинающими пользователями программы обучения включают темы, связанные с принципами работы операционной системы и офисными технологиями.

При работе с опытными пользователями основным методом обучения является метод сравнения. Основная цель — научить пользователя делать в СПО всё то, что он делал посредством коммерческих программных продуктов.

Параллельно с обучением различных категорий слушателей ведутся работы по подготовке учебного обеспечения к реализуемым программам. Разработаны учебные пособия по пакету OpenOffice.org, по СУБД, по программному обеспечению для вёрстки и подготовки публикаций, по объектно-ориентированному программированию, по работе с образовательными пакетами и др. (всего более 20 наименований).

Следует отметить, что в своей деятельности ТГПУ находит поддержку в Департаменте общего образования Администрации Томской области, взаимодействует с районными ресурсными центрами.

В дальнейшем ТГПУ планирует развивать это направление деятельности. Наибольшие перспективы, с нашей точки зрения, в развитии дистанционного обучения по вопросам применения СПО в профессиональной деятельности.

С. В. Кучин

с. Кафтанчиково,
Томский район, Томская область,
МОУ «Кафтанчиковская СОШ»

Linux глазами сельского учителя и его учеников

Большинство учителей сельских школ являются совместителями (ведут информатику, математику, физику и т. д.) и не в состоянии конфигурировать операционную систему. Для этого у них нет ни времени, ни опыта. Что необходимо от СПО простому сельскому учителю, который не является суперспециалистом в настройке операционных систем? Ответ прост. Для того чтобы начать работу, необходим дистрибутив, при установке которого:

1. Автоматически настраивается сеть, включая Samba-клиент.
2. Обнаруживаются и автоматически монтируются все физические и логические диски.
3. Есть возможность выбора порядка загрузки Windows/Linux.
4. Есть возможность установки и обновления с CD и DVD.

Этим требованиям в конце 2006 года в большей мере соответствовал дистрибутив Mandriva PowerPack, который и был установлен в МОУ «Кафтанчиковская СОШ» как вторая операционная система. Немного позднее захотелось разнообразия. Так появились Kubuntu и ALT Linux.

В течение 2007–2008 гг. на базе кабинета информатики Кафтанчиковской школы производилось тестирование различных дистрибутивов Linux. Тестерами выступали ученики школы. Они высказывали своё личное, скорее всего, субъективное мнение о разных версиях Linux. Возможно, их взгляд отличается от мнений профессионалов, но не надо забывать, что использовать СПО в школах будут именно они.

Перед учениками ставилась задача сравнить между собой предложенные им дистрибутивы. В тестировании участвовало 40 учащихся 9–11 классов. Отзывы были самые разные, но при их обработке проявилась интересная закономерность.

Итоги Исследования:

Место в рейтинге	Объекты исследования
1	Mandriva Linux 2007 Spring Free
2	ALT Linux 4.0 Junior
3	Mandriva PowerPack 2008
4	Mandriva PowerPack 2007
5	Kubuntu 7.4

Вот отзывы учеников, приведённые с сохранением орфографии и пунктуации:

- «Мне больше понравилась Mandriva Linux 2007 Spring, так как видом он лучше, функции у всех одинаковые, но у него как то более упрощено и понятно, чем у других. Второе место ALT Linux из за того, что в нем есть различные учебники по геометрии, алгебре, словари, переводчики.»
- «Все образцы ОС которые мы наблюдали сегодня примерно одинаковы. Mandriva Linux 2007 Spring имеет больше количество различных редакторов, прост в использовании и удобен даже для начинающих пользователей ПК. ALT Linux имеет также большое количество редакторов и обучающих программ, игр.»
- «Linux 2007 Спринк. Он мне понравился потому что у него хорошее оформление и хорош он в ориентировке.»
- «Mandriva PowerPack 2008 — самая удобная версия Linux. Удобное обращение компьютерных устройств. Он более похож на Windows XP. Он более комфортен в работе и понятен. На втором месте я бы поставил ALT Linux, в нем тоже хорошо работать, но у него худшее оформление, чем у PowerPack. На последнем — Кубунта, с ним бы я вообще не хотел работать. Он мне не понравился.»

Конечно, некоторые ученики пытались сравнить Linux и Windows XP, причём на примере прикладных программ. Поэтому следующим логическим шагом была попытка сравнения офисных программ.

Учащимся было предложено выполнить одинаковые по содержанию задания, используя разные текстовые редакторы.

1. OpenOfficeWriter
2. MS Word 2003
3. MS Word 2007

Результат никого не удивил. Опытные пользователи MS Word 2003 успешно справились с заданием в OpenOfficeWriter примерно за то

же время. Для них более сложным оказался MS Word 2007. Ну а у начинающих во всех трёх редакторах были одинаковые проблемы.

Всё вышеизложенное подтверждает следующее:

Пользователи, которые первый раз сели за компьютер, одинаково сложно осваивают как Linux, так и Windows. Им совершенно без разницы, в каком «офисе» работать и в каком формате сохранять данные. Поэтому нельзя говорить о том, что «Винда» интуитивно понятнее, чем «Линукс», здесь можно говорить скорее о привычке, чем о результате.

Мне понравилось то, как сам Линус Торвалдс относится к своему творению. Он искренне считает, что скоро никто не будет даже думать о том, что есть такая вещь, как операционная система — все технологии будут просто обслуживать сферу развлечений и общения между людьми.

В. Б. Смирнов

Томская область, с. Зырянское

Сопровождение сервера в школе

Проблема настройки сервисов специалистом с неглубокими знаниями. Оценив преимущество ручной настройки через текстовые файлы хотелось бы иметь и графический интерфейс. При недостатке времени гораздо эффективнее работать по смешанному методу. Задать характеристики через графический интерфейс, а далее править и доводить систему напрямую. В этом случае, по крайней мере, есть начальная рабочая конфигурация, которая позволяет редактировать настройки в течение 2–3 месяцев.

Практика нашей работы показала, что в средней школе свой сервер жизненно необходим. Как пример, в свое время web- и ftp-серверы, поднятые под ОС Windows, сыграли решающую роль в развитии информационных технологий в ЗСОШ №2. Далее школа «взяла» нацпроект по данному направлению. Это краткое пояснение зачем школам серверные технологии.

Далее проблемы. Сопровождение сервера в школе ложится на плечи учителя информатики. Помимо всего прочего, у этого учителя нагрузка 25–30 часов, семья и какие то личные интересы :). На экспе-

рименты с ПО время ограничено, а службы должны работать, что называется, ещё «вчера».

Разворот Desktop-ных систем СПО в сторону пользователя не может не радовать. Linux-системы обладают достаточно широкими возможностями, и глупо ими не пользоваться. Проблема работы с серверами на базе Linux для меня кроется в длительном изучении работы и настройки программы из-за плохого знания английского и недостатка времени. Задавать вопросы иногда просто стыдно. . . Большую помощь в настройке сервисов оказывают работоспособные блоки файлов конфигурации. В связи с этим очень хочется видеть графический интерфейс настройки интернет-сервисов, внутрисетевых служб, в частности: Web, Ftp, mail, proxi (sqwid), Ldap, samba, установки устройств.

Создав через графическую оболочку первоначальную конфигурацию, новичку легче будет ориентироваться в *.conf. При этом нельзя не согласиться с тем, что «тонкую» настройку лучше проводить в текстовом режиме.

Несколько слов о том, зачем и какие серверы нужны школе:

Web — наш сайт занимает более двух Гбайт, и это не предел; на чужом ресурсе поддерживать его проблематично.

Ftp — хранение объёмных ресурсов сайта, внутрирайонный обмен данными.

Mail — удобно иметь свой сервер, ученики быстро осваиваются в e-mail-пространстве, проверено.

Ldap — компьютеров много, дети работают на разных машинах в разное время, особенно после уроков. Централизованный доступ к своему профилю снимет многие проблемы.

Samba — системы Windows ещё долго будут соседями.

Открытие доступа к папкам — пример: класс компьютеров, 11 штук, у каждого жёсткий диск на 320 Гбайт, хочется их использовать в сети.

LTSP с настройкой подключения определенных машин, настройкой подключения терминальных клиентов, в том числе и из под Windows.

И ещё вопрос к разработчикам: возможно ли сделать распределённую вычислительную структуру на базе компьютерного класса? Из области желаемого (фантастики) — кластер из 4–8 компьютеров, который работает как один терминальный сервер.

Кроме того, на Desktop-ных системах хотелось бы иметь графический интерфейс работы с жёсткими дисками (монтирование, разметка, форматирование, в том числе и дополнительных дисков).

Возвращаясь к теме графическо-текстового редактирования файлов конфигурации серверов, следует добавить, что учитель информатики никогда не будет отлично разбираться в тонкостях настроек. Вряд ли он останется учителем после этого, причём по разным причинам. . .

Л. А. Татарникова

Томск,
ОЦ «Школьный университет» ТУСУРа

Проект: Внедрение и сопровождение свободного программного обеспечения в учреждениях общего образования Томской области
<http://spo.tomsk.ru>

Переход на СПО: методическое обеспечение

В последние годы мы немного (и это мягко сказано) избаловались. Разнообразие прикладных программ (Хочешь изучать графику? Вот тебе Photoshop и Corel! А ты хочешь заниматься черчением? Есть прекрасная программа, AutoCAD называется!), обилие обучающих программ, достаточно надёжная и, главное, привычная система позволяли строить уроки информатики «на любой вкус». Преподаватель определился с программами и учебниками, наработал уроки, выверил презентации, подобрал контролирующие материалы. . . Причём речь идёт не только об учителе информатики — многие предметники активно используют компьютеры на уроке.

Что нас ожидает в следующие два (ну, может, три) года? Новая система будет установлена, сеть настроена, принтеры подключены. Можно работать?

Если бы так. . . Как говорится, «стрижка только начата». Забот много: подобрать аналоги тем программам, которыми пользовался в учебном процессе, и адаптировать к ним свои методические материалы, изучить новые программы и разработать для них методику преподавания. Собственные обучающие программы перенести на новую платформу. Переработать программы и учебные планы, адапти-

ровать, дополнить, изменить контрольные работы. Мы уже не говорим об учебниках и обучающих программах профессиональных фирм.

Образовательный центр «Школьный университет» подобрался к решению этой проблемы почти два года назад. Отчасти из-за того, что некоторые школы начали задавать вопрос: «А когда у вас будут практикумы под Linux?», отчасти из-за того, что «просто интересно» — есть под Linux интереснейшие программы. Ну а события последнего года значительно ускорили процесс перехода к разработке курсов под СПО.

Первое, что хочется отметить, — это открытый электронный практикум «Уроки Linux», разработанный в рамках областного проекта. Что он собой представляет? 46 уроков, позволяющих учителю хоть немного сориентироваться в мире Линукс-программ. Правда, эти уроки создавались под конкретную систему — Kubuntu. Однако, их большую часть можно легко адаптировать и под другие системы.

Уже «вторым изданием» выходит в свет курс компьютерного дизайна, базирующийся на использовании таких программ, как GIMP и Inkscape. Популярная, востребованная тематика — реклама, яркие примеры — весь процесс создания «произведения рекламного искусства», от идеи до её реализации. Теория, подкрепляемая интересными практическими примерами, и практика, позволяющая постичь теоретические основы, — кредо курса.

Примерно год назад среди itdrom-овцев (активных обитателей портала itdrom.com) нашёлся искренний почитатель OpenOffice, студент, который взялся переработать курс офисных технологий под этот уже довольно известный пакет. Что можно сказать об офисных приложениях? Думаю, что если обычному пользователю Word предложить оформить документ во Writer, то процесс адаптации произойдёт за 5–30 минут. Действительно, интерфейсы Word и Writer очень похожи, модель документа практически та же. Однако, «продвинутым» пользователям, использующим достаточно редкие, «глубинные» возможности Word, придётся затратить на осваивание OpenOffice намного больше времени.

Переработанный курс OpenOffice придётся кстати и тем, и другим. Будет полезна и справочная система, позволяющая получить некоторые теоретические и общие сведения, и комплект задач для самостоятельного решения.

Программирование. Особый раздел информатики, который, казалось бы, не зависит от системы. Пузырьковая сортировка работает

одинаково и под Windows, и под Linux. Однако при создании программ мы используем среды разработки и, к сожалению, среды «под Linux» чаще всего не такие, к каким мы привыкли в Windows.

Но и в этой сфере Школьный университет может предложить помощь учителю и ученику. Прежде всего, переработан курс программирования на языке Pascal, который теперь базируется на использовании среды Free Pascal параллельно с Borland Pascal.

Кроме того, ведётся работа над курсом программирования для исполнителя Черепашка, ориентированная на среды Logo Writer, K Turtle и BlackBox. Этот курс ориентирован на использование в среднем звене (7–9 классы).

Вы хотите изучать алгоритмизацию и программирование ещё раньше? Действительно, алгоритмическая грамотность сейчас востребована во многих сферах деятельности, она же помогает детям легче ориентироваться в других дисциплинах, причём не только в математических.

Для учащихся 5–7 классов можно предложить исполнителя Little Wizard. Составление программ для этого исполнителя похоже на складывание мозаики. Играя, учащийся получает как знания из теории алгоритмов, так и навыки составления, редактирования и сохранения программ.

Все курсы Школьного университета предлагают учителю не только учебные пособия и электронные практикумы, позволяющие детям работать самостоятельно, но и множество методических материалов: программу курса, комплекты для контроля и оценки знаний, методические рекомендации к проведению занятий, материалы для проектной деятельности.

Хочется верить, что УМК Школьного университета значительно облегчат учителю переход к свободному программному обеспечению. Во всяком случае, мы делаем для этого всё возможное.

Д. В. Ефанов

Москва, ОАО «ВНИИНС»

Проект: ВС Школьный Сервер

www.school-linux.ru

Организация школьного сервера на базе дистрибутива Debian GNU/Linux

Аннотация

Рассматривается задача организации сервера в школе. Предлагается использование дистрибутива Debian GNU/Linux. Рассматриваются технические решения по взаимодействию школьного сервера с продуктами «1С:ХроноГраф Школа» и «Электронная учительская».

Активное внедрение информационных и коммуникационных технологий в современной средней школе привело к появлению компьютерных классов, представляющих собой локальные вычислительные сети с выходом в глобальную сеть Internet. Существенно изменилась роль компьютера в школе. Если раньше компьютер использовался исключительно на уроках информатики, то сегодня с его помощью изучаются практически все школьные предметы. Компьютер также стал инструментом работы директора, секретаря, библиотекаря, психолога, бухгалтера — всех административно-хозяйственных служб школы.

В результате возросла роль школьного сервера, на который возлагается как решение классических серверных задач (централизованное хранение файлов и печать документов, организация выхода в Internet), так и специфическая задача создания информационного образовательного пространства (ИОП), в значительной степени определяющая требования к программному обеспечению школьного сервера, и в первую очередь к серверной операционной системе.

С точки зрения создания ИОП школьный сервер является образовательным ресурсом специального назначения, состоящим из двух частей: внутренней, обращённой к школьному сообществу (учащиеся, учителя, родители), и внешней, представляющей школу в мировом информационном пространстве.

С одной стороны, школьный сервер должен быть технологической основой внутренней школьной информационно-образовательной и управленческой среды. Он должен предоставлять основную нормативно-справочную и оперативную информацию об участниках образовательного процесса, внутреннюю статистику для учителей и администрации, информацию для родителей, учащихся и учителей индивидуального и (или) конфиденциального характера, доступ к которой

определяется ролью или именем пользователя и паролем, информацией о внутреннем распорядке школы, текущих событиях, текущих и итоговых результатах внутришкольного контроля и др.

С другой стороны, школьный сервер является виртуальным представителем данной школы в Internet. Он демонстрирует и рекламирует её особенности как образовательного учреждения (образовательные программы, особенности обучения, направления специализации, правила приёма), а также представляет результаты деятельности школьного коллектива, формирующие информационные образовательные ресурсы (методические разработки учителей, работы детей и образцы детского творчества).

Кроме того, учитывая современный всплеск социальных сетей и педагогические возможности коммуникаций в современном образовании и воспитании, целесообразно также предусмотреть коммуникационный блок, организующий как сетевое общение учителей и учащихся, посвящённое решению педагогических проблем, дидактических задач и обобщению опыта, так и свободное общение.

Также должны быть решены юридические вопросы применения школьного сервера как программного средства обработки информации, в контексте выполнения закона «О защите персональных данных».

Для комплексного решения представленных задач Национальный фонд подготовки кадров ведёт проект «Информатизация системы образования», целью которого является создание условий для системного внедрения и активного использования информационных и коммуникационных технологий в работе школ. В рамках данного проекта ОАО «ВНИИНС» ведёт разработку серверной операционной системы на базе дистрибутива Debian GNU/Linux, получившей название «ВС Школьный Сервер» и предназначенной для организации школьного сервера.

Выбор Debian в качестве технологической основы обоснован его высоким качеством и существованием вокруг проекта Debian одного из старейших сообществ пользователей и разработчиков Linux. Debian широко используется во всём мире в качестве базы для многих проектов, в т. ч. и в сфере образования. Одним из недавних примеров является начало массового внедрения в бразильские школы дистрибутива Linux Educacional 2.0, основанного на Debian.

Отличительной особенностью «ВС Школьный Сервер» является возможность запуска на клиентских рабочих местах под управлени-

ем Windows программного комплекса «1С:Управление школой», предназначенного для автоматизации администрирования общеобразовательного учреждения и поставленного в российские школы в рамках проекта по обеспечению лицензионной поддержки стандартного базового пакета ПО для общеобразовательных учреждений.

Особое внимание в «ВС Школьный Сервер» было уделено взаимодействию школьного сервера с конфигурацией «1С:ХроноГраф Школа» и с программно-технологическим комплексом «Электронная учительская».

Таким образом, задача организации сервера в школе может быть решена с использованием дистрибутива «ВС Школьный Сервер».

Литература

- [1] Дашниц Н. Л. Роль школьного веб-сервера в образовательном процессе современной школы.
- [2] http://www.ict.edu.ru/vconf/index.php?a=vconf&c=getForm&r=thesisDesc&id_sec=190&id_vconf=33&id_thesis=7148&d=light

В. М. Вымятнин

Томск,
Томский государственный университет

СПО в ВУЗе: необходимое, но не достаточное

Отказ от использования коммерческого ПО (в первую очередь подразумевается ОС семейства Windows) и переход на «свободное» ПО (подразумевается, как правило, один из многочисленных дистрибутивов ОС Linux) в последние годы стали предметом широкого обсуждения не только в среде ИТ-специалистов, но и в СМИ. Этому в немалой степени способствовало и «дело Поносова». Масла в огонь подлило и принятое на правительственном уровне решение поставить через три года систему общего среднего образования перед выбором: покупка Windows (и ПО, работающего на этой платформе) за свой счёт или переход на платформу Linux (и использование свободно распространяемого прикладного ПО).

Конечно, для школы, с её постоянным недофинансированием второй вариант представляется более привлекательным (особенно, если забыть, что кроме стоимости приобретения существует и стоимость владения). Тем более, что для платформы Linux разработано практически всё прикладное ПО, необходимое для обеспечения учебного процесса в школе (и не только в рамках курса информатики). Пользовательский интерфейс современных программных продуктов определяется эргономикой, а не программной платформой, поэтому школьник, освоивший работу в OpenOfficeOrg Writer, без особого труда сможет работать и в Microsoft Word.

Говоря о школе, не следует забывать, что кроме учебного процесса есть ещё и административная деятельность. Она связана с контактами с вышестоящими и сторонними организациями и соответствующим документооборотом. При переходе на электронный документооборот проблема совместимости форматов документов приобретает высокую важность. Поскольку вопрос о переходе всех государственных учреждений на СПО пока не ставится, полное «изгнание торговцев» (коммерческого по с маркой MS) «из храма» (школы) нереально.

Миссия высшего профессионального образования качественно отличается от миссии общего среднего. Задача школы — дать выпускнику основы знаний (в том числе и в области ИТ), необходимых для практической деятельности в различных областях, а также и для продолжения образования. Задача ВУЗа — подготовить специалиста, способного использовать новейшие научные и технологические достижения в своей профессиональной области (в том числе и соответствующее профессионально ориентированное ПО).

В Томском государственном университете СПО в качестве сетевых ОС (сначала FreeBSD, а затем и Linux) используется почти 15 лет. Поскольку серверные продукты (коммерческие!) ведущих производителей (Oracle, IBM и т. д.) имеют дистрибутивы как для Windows, так и для ряда версий UNIX, это не вызывает проблем. В качестве веб-платформы широко используется популярная СПО-тройка «Apache + PHP + MySQL».

Однако использование Linux в качестве настольной ОС создает больше проблем, чем решает: многие профессионально-ориентированные пакеты не имеют Linux-версий, а кроссплатформенное ПО, как правило, не относится к классу СПО, и его стоимость для различных ОС практически совпадают. Кроме того, наличие в одном подраз-

делении компьютеров, работающих под различными ОС, усложняет работу администраторов.

Более перспективным является использование СПО там, где коммерческое ПО не является стандартом de facto. Примером может являться компьютерная графика «для личных нужд», где можно прекрасно использовать Paint.NET или Gimp. Однако в профессиональной среде «компьютерных художников» есть свои корпоративные стандарты, и это необходимо учитывать при подготовке специалистов. Специалист, прекрасно владеющий Gimp'ом и Inkscape, но не знающий тонкостей работы с продукцией Adobe и Corel, может остаться без работы.

Мощные математические, статистические, лингвистические и т. д. пакеты — это многофункциональность и надёжность (а отсюда и высокая цена — за вложенный интеллект). И если фирмы, их производящие, желают использовать только платформу Windows — это их право. А вот соответствующие облегчённые аналоги (не для «крутых» профессионалов), свободно распространяемые и портированные как под Windows, так и под Linux (а может быть, и под другие ОС?), могут побудить многих к отказу от «тяжёлой артиллерии».

Имея необходимый набор свободно распространяемого кроссплатформенного прикладного ПО, можно безболезненно поменять коммерческую ОС на бесплатную. Но не следует забывать и про стоимость владения.

С. И. Иевлев

Москва, ALT Linux

Проект: alterator

<http://wiki.sisyphus.ru/Alterator>

ALT Linux Installer

Аннотация

После нескольких лет разработки в ALT Linux была создана новая современная программа установки операционной системы. Она обладает высокой функциональностью (широкий выбор источников установки, возможность автоустановки), простым интерфейсом, модульностью и широкими возможностями по адаптации к конкретному продукту.

Инсталлятор — неотъемлемая часть любого дистрибутива. Основная задача инсталлятора — корректное развёртывание системы

на оборудовании пользователя. От него же зависит и общее впечатление от системы в целом.

Инсталлятор ALT Linux есть результат объединения следующих продуктов: общая инфраструктура (`installer`), штатный конфигуратор системы (`alterator`) и профиль установки (`installer-desktop`, `installer-server` и т. д.).

`Alterator` — модульная система с поддержкой трёх различных интерфейсов взаимодействия с пользователем: командная строка, традиционный GUI и `http`-интерфейс через `web`-браузер. В инсталляторе и центре управления системой используются одни и те же модули. Благодаря этому достаточно легко отлаживать отдельные компоненты инсталлятора.

`Installer` состоит из трёх стадий. Первая, загрузчик, поддерживает массу способов загрузки основной части инсталлятора: диск, `nfs`-сервер, `cdrom`, `ftp`-сервер (анонимный и с паролем), `http`-сервер. В первых двух случаях возможна работа как с развёрнутым дистрибутивом, так и с `iso`-образом непосредственно. Вторая стадия инсталлятора выполняет основную часть работы — разбивку диска и установку базовой системы. Третья стадия работает уже внутри установленной системы. Благодаря такому разбиению на каждом этапе имеет место минимальный расход оперативной памяти, что позволяет устанавливаться на достаточно слабые машины.

Инсталлятор работает в стиле пошагового выполнения модулей конфигуратора. Шаги могут быть интерактивные и неинтерактивные. Последние оформлены в виде скриптов и разделены на следующие группы: `initinstall.d` — инициализация инсталлятора, `preinstall.d` — перенос настроек из среды инсталлятора в установленную систему (между второй и третьей стадиями инсталлятора), `postinstall.d` — изменения в установленной системе по окончании работы инсталлятора.

Каждый модуль конфигуратора, работающий в инсталляторе, может иметь свой набор `preinstall.d`-скриптов. Таким образом, когда составитель дистрибутива заменяет модуль на эквивалентный, прозрачным образом происходит замена и необходимых для его работы скриптов.

Профиль инсталлятора — содержит описание шагов, описание порядка исполнения шагов, а также набор дополнительных скриптов. Для наиболее распространённых случаев можно воспользоваться готовыми коллекциями скриптов из пакетов серии `installer-feature`. На-

пример, `installer-feature-pxeboot` делает все необходимые настройки в системе, чтобы сразу после установки она могла функционировать как `tftp`-сервер.

Таким образом, инсталлятор может быть очень точно подстроен под требования конкретного выпускаемого продукта, а наличие большого количества готовых компонент на все случаи жизни позволяет свести к минимуму возможные ошибки при составлении профиля.

К сказанному осталось только добавить, что инсталлятор имеет отладочный режим (что позволяет запустить его в случае сбоя автоопределения оборудования) и режим неинтерактивной автоустановки по заранее составленному сценарию.

М. С. Пожидаев

Томск,
Томский государственный университет

Проект: Indocs

Обзор информационной системы для торговли и складского учёта Indocs

Программный пакет Indocs создан для организации процесса торговли и складского учёта. Система разработана на базе СУБД PostgreSQL. Клиентская часть реализована на языке Java (j2se). К основным возможностям системы относятся:

- организация иерархического каталога товаров с неограниченной глубиной вложенности;
- складской учёт с указанием серийных номеров товара;
- построение прайс-листов;
- окружение для выписки счетов клиентам;
- оформление расходных документов;
- хранение информации о клиентах.

Складской учёт ведётся через систему накладных. Приходные накладные работают одновременно в двух валютах — рублях и долларах. Цена товаров рассчитывается автоматически с указанием надбавки и текущего курса иностранной валюты.

Такие характеристики, как многопользовательский сетевой доступ, разделение прав пользователей, поддержка транзакций, полная переносимость клиентской части на любую платформу, являются следствием возможностей базовых компонентов (Postgresql и Java). Печатные формы строятся при помощи библиотеки JasperReports.

Система имеет двухуровневую архитектуру. Первый слой — это ядро операций с данными, второй — пользовательский интерфейс. Операции с данными реализованы только для хранения в реляционных СУБД при поддержке языка SQL. Локальная работа системы может быть реализована при помощи пакета hsqldb. Оконная пользовательская часть выполнена на основе библиотеки Swing.

В настоящий момент система Indocs используется в компьютерной фирме «Интек» в г. Северске. Её исходный код публикуется под лицензией GPL. Главная цель разработки — максимальная независимость от лицензионных ограничений. Кроме этого, данная система позволяет организовать работу предприятия с использованием только открытого программного обеспечения. Исходный код системы хорошо структурирован и допускает возможность модификации для требований, не предусмотренных текущей реализацией.

К недостаткам системы относятся невозможность репликации удалённых пользователей и полная реализация логики работы приложения на стороне клиента (модель «толстый клиент»).

С. В. Подушкин

Томск, ООО «АББРИС»

Проект: ОфисМастер, OfficeMaster

<http://www.abbris.ru>

ОфисМастер — сервер инфраструктуры для «чайников»

Аннотация

ОфисМастер — сервер на основе ОС Линукс, разработанный для использования в организациях, не имеющих штатных ИТ-специалистов, предоставляющий полный набор инфраструктурных служб. Основные задачи администрирования сведены к типовым и для их выполнения используется простой веб-интерфейс. Сервер обеспечивает регистрацию пользователей и групп, лёгкое управление доступом к электронной

почте и онлайн ресурсам, учитывает потребляемый трафик, защищает от вирусов и нежелательных сообщений и многое другое.

ОС Линукс практически с первых своих дней была и остаётся гибкой, надёжной и безопасной основой для серверных решений.

Открытость самой операционной системы и приложений, разрабатываемых для неё, обеспечивает возможность создавать качественно новые решения без необходимости «изобретать колесо» заново. Одним из таких решений и является ОфисМастер.

Комбинируя традиционные корпоративные технологии под удобным и простым интерфейсом, он открывает доступ к основным ИТ-службам с минимальными затратами, позволяя обеспечить высокую эффективность, независимость и безопасность бизнеса.

ОфисМастер — это не просто очередное техническое решение для пополнения списка «полезных инструментов», это решение, комбинирующее технологии и практику эффективного и безопасного их применения в организации. Именно практика применения технологий и является центром внимания фирмы «АББРИС». Это находит отражение в применённом в ОфисМастере интерфейсе «инструкция всегда под рукой».

Что даёт ОфисМастер?

ОфисМастер обеспечивает автоматическое распределение адресного пространства и автоматическую регистрацию имён компьютеров в локальной сети, позволяя просто включать компьютер в локальную сеть и приступать к работе.

ОфисМастер обеспечивает централизованную аутентификацию пользователей, обеспечивая парольную защиту ресурсов.

ОфисМастер обеспечивает соблюдение политики качества паролей и средства смены/сброса паролей через веб-интерфейс.

ОфисМастер позволяет регистрировать пользователей с размещением такой информации, как телефоны, адреса и фотографии. Также обеспечивается регистрация групп пользователей. И пользователи, и группы могут выступать в качестве адресатов при электронной переписке. Информация о пользователях и группах доступна в виде корпоративной адресной книги как через веб-интерфейс сервера, так и в почтовых программах.

ОфисМастер предоставляет файловое хранилище для совместной работы с возможностью разграничения доступа к файлам и папкам как персонально, так и по группам пользователей.

ОфисМастер обеспечивает развитые возможности электронной почты. В состав сервера включён почтовый веб-клиент, позволяющий работать с электронной почтой с любой машины в локальной сети. Средствами управления доступом можно запретить использование пользователем почты, ограничить использование почты рамками организации либо предоставить неограниченный доступ.

Адреса групп могут использоваться как для обычной пересылки электронных сообщений членам группы, так и для «умной рассылки», при которой члены группы всегда получают сообщения от имени группового адреса, что позволяет сохранить переписку в рамках группы. Такой режим даёт возможность всем членам группы быть в курсе всей переписки, в любое время вмешаться в переписку, поддерживать её в случае отсутствия других членов группы. Этот режим полезен для создания временных адресов для рекламных акций, приёмных кампаний в учебных заведениях, официальных адресов и пр.

ОфисМастер обеспечивает доступ к онлайн-ресурсам через прокси-сервер. Такой доступ позволяет как повысить эффективность использования Интернет-соединения за счёт повторного использования ранее полученной информации, так и обеспечить гибкое управление доступом пользователей к ресурсам.

ОфисМастер проверяет всю информацию на наличие вирусов. Электронная почта эффективно защищена от спама.

ОфисМастер ведёт персональный учёт использования пользователями ресурсов в режиме, близком к реальному времени. ОфисМастер предоставляет набор готовых форм отчётов для контроля потребления интернет-служб и, кроме прочего, позволяет выяснить, какие сайты посещают пользователи, кому отправляют и от кого получают электронные письма.

Доступ к функциям ОфисМастера разграничен на основе членства в специальных группах, обеспечивая удобные возможности по распределению и делегированию полномочий в организации, позволяя избежать зависимости от единственного человека (обычная ситуация в малых компаниях с одним администратором).

ОфисМастер предоставляет в распоряжение пользователей «доску объявлений».

ОфисМастер — открытое решение, компоненты которого написаны на популярных языках PHP и Python, а это значит, что оно всегда может быть развито до необходимого заказчику уровня.

ОфисМастер — решение, не имеющее никаких ограничений — ни по числу подключений, ни по числу учётных записей, ни по допустимой нагрузке, что делает его очень выгодным на фоне коммерческих решений.

Д. В. Левин

Москва, ALT Linux

Проект: hasher

<http://ftp.altlinux.org/pub/people/ldv/hasher>

hasher: технология безопасной сборки пакетов

Аннотация

Рассматривается задача безопасной воспроизводимой сборки пакетов репозитория, изучаются требования, накладываемые этой задачей на архитектуру системы сборки, рассматривается архитектура *hasher*'а и существенные моменты её реализации. Приводятся примеры использования *hasher*'а.

Традиционная схема сборки дистрибутивов

Давным-давно, когда дистрибутивы операционных систем помещались на один компакт-диск вместе с исходным кодом, а создавали их узкие группы специалистов, никакой сборочной технологии по существу не было, а отдельные элементы дистрибутива собирались прямо в хост-системе, полученной из полностью установленного дистрибутива.

Со временем инструментальные дистрибутивы выросли в размере, увеличилось число принимающих участие в разработке дистрибутивов, и в результате сборка дистрибутива в хост-системе стала неудобной, ненадёжной и небезопасной.

Сборка в хост-системе: неудобство и ненадёжность

Необходимость установить дистрибутив полностью приводит к неоправданно большому размеру сборочной среды.

В свою очередь, при таком размере среды проявляется несовместимость инструментальных средств, которые не всегда удаётся решить с помощью альтернатив, переключателей и других приёмов.

Поскольку все инструментальные средства установить в систему не удаётся, возникает необходимость прав администратора для установки произвольных требующихся для сборки пакетов в хост-систему.

Кроме того, невозможно обеспечить параллельную сборку пакетов с несовместимыми сборочными зависимостями.

И, наконец, при таком слабоконтролируемом составе сборочной среды, которой является хост-система, возникает неявная, неочевидная и нигде не обозначенная зависимость результата сборки от конкретной сборочной среды, возникают «неприкосновенные сборочные серверы».

Сборка в хост-системе: небезопасность

Небезопасность хост-системы проистекает из-за самой возможности запуска произвольного кода с правами администратора при установке пакетов, требуемых для сборки.

Небезопасность пользователя, занимающегося сборкой, связана с запуском произвольного кода с правами сборщика непосредственно во время сборки.

Небезопасность сборок друг от друга обусловлена двумя факторами: возможностью произвольного изменения сборочного окружения во время сборки и возможностью непосредственного воздействия выполненного кода сборки на последующие сборочные процессы.

Сборка дистрибутива: источники угроз

Почему безопасности сборочной технологии уделяется так много внимания? Потому что сборка дистрибутива является областью повышенного риска.

Внимание blackhat community связано с привлекательностью компрометации дистрибутива, через который можно легко получить контроль над множеством серверов и рабочих станций.

Большое число разработчиков разной квалификации приводит к тому, что некоторые из них могут стать лёгкой добычей для злоумышленников и впоследствии могут быть использованы для атаки на дистрибутив.

Скомпрометированным может оказаться как клиентское ПО, используемое разработчиком, так и ПО, собираемое разработчиком. За последние два года были публичные случаи и того, и другого.

Нельзя исключать и возможность непосредственной атаки на сборочную систему.

Требования к сборочной технологии

Таким образом, технология сборки элементов дистрибутива (пакетов) должна:

- не снижать уровень безопасности хост-системы;
- обеспечивать собственную безопасность от атак со стороны пакетов;
- обеспечивать безопасность сборки одних пакетов от атак со стороны сборки других пакетов;
- гарантировать надёжность (воспроизводимость) результатов сборки;
- обеспечивать приемлемый уровень производительности.

Архитектура *hasher*'а

hasher базируется на принципе создания новой сборочной среды для каждой сборки.

В основе архитектуры *hasher*'а лежит трёхпользовательская модель: вызывающий непривилегированный пользователь (*C*) и два непривилегированных вспомогательных псевдопользователя; первый (*R*) играет роль root в порождаемой сборочной среде (далее псевдоroot), второй (*U*) — обычного пользователя, собирающего программы (далее псевдосборщик).

Переключение между вызывающим и вспомогательными пользователями осуществляется с помощью специальной привилегированной программы, написанной с применением параноидальных мер защиты от непривилегированных пользователей. Кроме того, с помощью этой программы удаляются процессы, запущенные вспомогательными псевдопользователями и не завершившиеся в срок, а также создаются файлы устройств. Наконец, эта программа предоставляет возможность контролировать ресурсы, выделяемые процессам непривилегированных пользователей, для защиты от DoS-атак.

Путь пакета через сборочную систему в общих чертах выглядит следующим образом:

1. Пользователь *C* порождает среду (artbox) для работы с art.

2. Полностью удаляется сборочная среда, возможно оставшаяся от предыдущей сборки. Удаление происходит последовательно в члуте пользователем U , в члуте пользователем R и, наконец, пользователем C .
3. Пользователь C создаёт каркас новой сборочной среды, состоящий из вспомогательных каталогов и вспомогательных статически слинкованных программ (`ash`, `find` и `cpio`). С помощью вспомогательной привилегированной программы создаётся фиксированный набор файлов устройств, достаточный для нормального функционирования сборочной среды и при этом не несущий угрозы `host`-системе.
4. Порождается базовая установочная среда, представляющая собой набор средств, необходимых для штатной установки пакетов в эту среду. Пользователь C с помощью `aptbox` определяет набор пакетов, необходимых для порождения базовой установочной среды. Пользователь R с помощью вспомогательных статически слинкованных программ распаковывает эти пакеты.
5. Порождается базовая сборочная среда, представляющая собой набор средств, необходимых для сборки любого пакета. Пользователь C с помощью `aptbox` определяет набор пакетов, пользователь R устанавливает их.
6. Проверяется исходный пакет.
7. Порождается сборочная среда для данного пакета. Пользователь U извлекает сборочные зависимости пакета, пользователь C с помощью `aptbox` определяет набор пакетов для установки и пользователь R устанавливает их.
8. Пользователь U осуществляет сборку пакета и проверку результатов сборки.

Такая схема призвана исключить атаки вида $U \rightarrow R$, $U \rightarrow C$, $R \rightarrow C$, а также все виды атак на `root`.

Для повышения производительности, особенно важной при сборке большого числа пакетов, применяется кэширование базовой сборочной среды.

С помощью средств `apt` реализована возможность использования собранных ранее пакетов для сборки последующих пакетов.

***beehive*: распределённая сборка с помощью *hasher*'а.**

Благодаря свойству воспроизводимости результатов сборки *hasher* можно использовать для параллельной сборки большого числа паке-

тов на нескольких серверах. Таким образом удаётся достичь разумного времени сборки при средних вычислительных ресурсах. Открывается возможность организовать регулярное тестирование на перебираемость большого репозитория пакетов, что и было сделано на примере Sisyphus с помощью средства параллельной сборки *beehive*.

Литература

- [1] Dmitry V. Levin, «hasher: технология безопасной сборки дистрибутива», тезисы докладов 1-ой международной конференции разработчиков свободных программ на Протве, 2004.
- [2] Dmitry V. Levin, «hasher: технология виртуализации для безопасного выполнения приложений», тезисы докладов 2-ой международной конференции разработчиков свободных программ на Протве, 2005.

А. Г. Боковой

Москва, Samba Team

Проект: Samba <http://www.samba.org>, <http://ctdb.samba.org>,
<http://www.openchange.org>

Новая Samba

С 1992 года, когда Andrew Tridgell написал свою реализацию протокола SMB для Unix-подобных систем, чтобы получить доступ из клиента PathWorks в MS DOS к данным на сервере Sun, прошло уже более 15 лет. В 1996 году Microsoft потребовалась «прорывная технология», чтобы стать Internet-компанией, и такой технологией стал стек протоколов SMB, переименованный в CIFS, Common Internet File System. Расширенный и дополненный, десять лет спустя CIFS уже неотделим в сознании потребителей от Microsoft. Проект Samba традиционно рассматривался как попытка догнать Microsoft, но неожиданно в 2007 на неприметных устройствах хранения фотографий, музыки и видео их разработчики из азиатских стран пишут «совместимо с протоколом Samba», а не CIFS. Тихая революция?

2007 год вообще был богат на события. Команда разработчиков проекта Samba попадает на первые страницы главных деловых газет, таких как Financial Times, за победу над Microsoft в рамках антимонопольного процесса в Европейском Союзе. Затем Microsoft публикует более 40,000 страниц документации под лицензией, дающей возможность использовать её при разработке Samba, а также впервые с 1998 года принимает участие в конференциях, посвящённых разработке Samba. В 2008 году эта документация становится доступной всем желающим. Из самой закрытой компании ИТ-мира Microsoft становится самой открытой?

Но это только вершина айсберга. В 2007 году проект выпускает версию Samba, рассчитанную на высокопроизводительные системы хранения. Кластерная Samba в составе Scale-out File Services компании IBM позволяет горизонтальное масштабирование систем хранения с доступом на основе CIFS и демонстрирует производительность минимум в четыре раза быстрее любых коммерческих аналогов, включая Microsoft, фактического «законодателя мод» в области CIFS.

Небольшая группа внутри Samba Team за пять лет сформировала основу для замены Microsoft Active Directory. Samba 4.0, ещё не выпущенная и вряд ли готовая увидеть свет до конца 2008 года, тем не менее уже стала площадкой для атаки на другие бастионы «технологий Microsoft» — Microsoft Exchange. В проекте OpenChange уже реализована практически вся функциональность клиентской части Microsoft Exchange, что позволяет, например, GNOME Evolution работать в сетях Exchange наравне с клиентами от Microsoft, а MosaBox наступать на позиции Microsoft Unified Communications.

В то же время пользователи заблудились в трёх соснах. Samba 3.0, 3.2 и 4.0 развиваются одновременно и с разным функционалом, выглядят вкусными «стогами сена», дразнят и манят уже не первый год. Однако до сих пор проект не говорит пользователям, для чего планируется каждая версия и каков режим её выпуска. В докладе автор попытается объяснить позиционирование, новую модель разработки и выпуска разных версий Samba, их новинки и возможности. Также отдельное внимание будет уделено взаимодействию с разработчиками и компаниями, включая создателей Windows Vista и протокола SMB2.

О. Л. Садов

Санкт-Петербург, ОАО «ЛИНУКС ИНК»

Проект: Scientific Linux/НауЛинукс

<http://www.scientificlinux.org>, <http://www.naulinux.ru>

Международная инициатива Scientific Linux и проект НауЛинукс для школьного образования

Аннотация

Дано описание международного проекта Scientific Linux. Проанализированы возможности использования репозитория проекта (включая российский) для создания специализированных дистрибутивов.

Scientific Linux

Дистрибутив Scientific Linux (SL) был создан в 2004 Fermi National Accelerator Laboratory в сотрудничестве с рядом других лабораторий, занимающихся проблемами физики высоких энергий. Данный дистрибутив был создан на базе использовавшегося уже в течение нескольких лет дистрибутива «Fermi Linux». Изначально он носил название HEPL (High Energy Physics Linux). Ключевой особенностью данного дистрибутива являлась возможность его адаптации к конкретным нуждам отдельных лабораторий и университетов путём создания собственных вариантов дистрибутива — т. н. «сайтов». В том же году, после интенсивного обсуждения на конференции HEPIX, к проекту присоединился CERN.

Именно SL был выбран базовым дистрибутивом для GRID-системы обработки результатов экспериментов, проводимых на строящемся в CERN Адронном Суперколлайдере (LCG[1]). Его дальнейшим развитием стал общеевропейский (а теперь получивший и общемировой статус) проект EGEE[2]. На данный момент Grid-инфраструктуры, построенные на базе SL, объединяют сотни вычислительных центров. Эти центры расположены в десятках стран в разных частях мира и обладают совокупной вычислительной мощностью в десятки тысяч CPU. Число одновременно выполняющихся заданий достигает сотни тысяч, а объёмы обрабатываемых данных — петабайтов.

Уникальной является модель построения проекта SL. В ней удачно сочетаются деятельность сообществ разработчиков (самого SL, проектов CentOS и Fedora), крупного коммерческого разработчика (Red Hat) и структур, финансируемых государством (FermiLab, CERN). Такая схема серьёзно снижает традиционные для IT-проектов риски, связанные с использованием разработок по всем трём указанным выше моделям, и даёт высокую гарантию стабильного хода проекта.

SL строится на основе исходных пакетов Red Hat Enterprise Linux (RHEL) и позволяет использовать в нём бинарные пакеты и драйверы устройств, созданных для RHEL. SL объединяет наборы пакетов, входящие в серверные и десктопные варианты RHEL, включает в себя компоненты кластеризации, виртуализации и ряд дополнительных программных компонентов, добавленных разработчиками дистрибутива (например, поддержка OpenAFS).

Текущая версия SL 5.1 поддерживает архитектуры i386 и x86_64. В состав дистрибутива входит более 2.5 тысяч программных пакетов, позволяющих использовать его для рабочей среды пользователя с возможностью выбора комфортной графической среды (GNOME, KDE, IceWM), создания серверной и сетевой инфраструктуры, ведения разработки при помощи различных языков программирования и программных сред, построения кластерных конфигураций для проведения массовых вычислений, балансировки нагрузки, построения отказоустойчивых и даже катастрофоустойчивых решений.

Для повышения функциональности могут быть использованы программные репозитории[3] проекта, например: Dag Weers, AtRPMS, Dries Verachtert, Linux Ink, с совокупным объёмом почти в 9 тыс. программных пакетов. Также могут быть использованы пакеты из репозиториях других программных проектов (например, Fedora, EPEL и др.).

Коллектив разработчиков ОАО «ЛИНУКС ИНК» принимает участие в данном международном проекте, поддерживая программный репозиторий, ориентированный на SL, локализованный вариант Scientific Linux Cyrillic Edition и документации по нему[4], ведёт русскую версию сайта проекта[5], участвует в выпуске учебно-справочной литературы, посвящённой вопросам внедрения свободного ПО в целом[6] и данному дистрибутиву в частности[7].

НауЛинукс

В рамках проекта НауЛинукс решалась проблема адаптация SL для целей школьного образования. Освоение возможностей, предоставляемых данным программным решением на самом раннем этапе обучения, даёт сегодняшним школьникам возможность приобрести навыки работы в любом сколь угодно полном объёме в дистрибутиве, занимающем на данный момент ключевую роль в самых передовых научно-исследовательских проектах и по сути дела являющемся промышленным стандартом.

В данный момент компания ОАО «ЛИНУКС ИНК» ведёт ряд проектов на базе SL, имеющих отношение к учебному и научному процессу, результаты которых могут быть широко применены и в сфере школьного образования.

В рамках проекта «Разработка и апробация в пилотных субъектах Российской Федерации пакета свободного программного обеспечения для использования в общеобразовательных учреждениях Российской Федерации в 2007–2008 годах» ведётся разработка дистрибутива для рабочих станций. Было добавлено более 300 программных пакетов, адаптирована процедура установки, добавлена поддержка процессоров класса Pentium I, продемонстрирована возможность локальной пересборки. Осуществлена помощь коллегам из FermiLab в адаптации процесса генерации дистрибутива в виде набора CD. Был выявлен ряд программных ошибок, о путях исправления которых были оповещены участники проекта SL, компании производителя базового дистрибутива и разработчики соответствующих проектов. Ведётся работа по добавлению поддержки национальных языков (татарского) и системы, упрощающей развёртывание дистрибутива.

В рамках проекта ELSP/A2/C/071 НФПК «Разработка серверной операционной системы и комплекта серверных приложений для общеобразовательных учреждений на базе свободного программного обеспечения» ведётся работа по созданию серверного решения школьного уровня, интегрированного с существующими системами автоматизации школьной деятельности.

Кроме указанных выше проектов, компания ОАО «ЛИНУКС ИНК» принимала и принимает участие в ряде разработок, которые могут оказаться полезными на этапе создания образовательных ресурсов на базе свободных программных платформ. Примерами таких разработок являются порталные системы управления контен-

том с встроенными механизмами контроля исполняемых регламентов (workflow), привязка этого контента к картографическим данным[8], библиотечно-архивные системы[9] и др. Данные решения представляют собой WEB-ориентированные прикладные системы, допускающие кроссплатформенное использование, что на начальном этапе внедрения решений на базе свободного ПО в школьной деятельности будет иметь определяющее значение.

Литература

- [1] LHC Computing GRID <http://lcg.web.cern.ch/LCG/>
- [2] Enabling Grids for E-sciencE <http://public.eu-egee.org>
- [3] SL RPM Repositories <https://www.scientificlinux.org/community/repo>
- [4] Русская документация по SL <http://www.linux-ink.ru/static/>
- [5] Русская версия сайта проекта SL <http://www.scientificlinux.ru>
- [6] Руководство по миграции на Linux для региональных администраций http://www.ibm.com/developerworks/ru/library/linux_migr/contents.html
- [7] Red Hat Enterprise Linux/Scientific Linux. Полное руководство пользователя <http://www.bhv.ru/books/book.php?id=178797>
- [8] Прототип системы публикации картографических данных <http://ecology.linux-ink.ru>
- [9] Прототип электронно-библиотечной системы <http://elbi.linux-ink.ru>

М. А. Шигорин

Киев, ООО Медиа Мэджик

Проект: ALT Linux Team

<http://sisyphus.ru>

Linux Terminal Server Project

Аннотация

Преимущества использования терминальных технологий для целей миграции на свободное ПО и повторного использования морально устаревшего «железа». Участие в проектах ALT Linux и LTSP: порознь и вместе. ALTSP как уникальный сплав «лучшего из двух миров» — LTSP4/5.

Доистория

В 1999 году уже существовало как движение свободного ПО, так и намерение разработчиков ядра Linux вплотную заняться настольными системами. Этим воспользовались в одной детройтской фирме, занимавшейся обслуживанием госпиталей; так появилась первая версия LTSP.

События развивались довольно быстро, и в 2004 был выпущен LTSP4 с очень неплохими эксплуатационными характеристиками: работа на 12М RAM, загрузка в полминуты, поддержка балансировки нагрузки и локально запускаемых на клиенте приложений (например, для интернет-телефонии с доступом к микрофону).

LTSP5

В 2005 году начались работы над фреймворком по составлению терминальных решений из существующих дистрибутивов, а не над специализированным мини-дистрибутивом. Эта работа выполнялась в тесном сотрудничестве с проектом Ubuntu.

При том, что технологически переработка представляла собой прорыв, на практике многие ценные свойства были утрачены или заменены на другие. Например, существенно возросли аппаратные требования к терминалам, сильно возросло время их загрузки, исчезла поддержка LocalApps и кластеризации. В отличие от того, что задумывалось, ситуация с безопасностью также ухудшилась.

Всё это было оправдано происшедшим заложением фундамента для дальнейшей мегадистрибутивной разработки: теперь LTSP предоставляет методологию по поддержке загрузки и обеспечения работы клиентов и компоненты, готовые к использованию для интеграции такого режима в существующие дистрибутивы.

В 2007 году к разработке LTSP5 присоединились участники проектов Gentoo, openSUSE, Fedora, Slackware, принося с собой кусочки специфического для их дистрибутивов кода, исправляя и улучшая общий.

ALTSP5

В том же 2007 году был доведён до бета-состояния и использован в деле форк LTSP5 на базе ALT Linux, сочетавший в себе как наработки по использованию дистрибутива из пятой ветки, так и зрелые компоненты из четвёртой; наш офис[2] успешно переехал на технологию ALTSP5 в конце весны.

К концу года был наработан дистрибутив, который обеспечил на стенде загрузку и нормальную работу Pentium 166/32M как одного из тонких клиентов, подключённых к более мощной машине — даже при использовании KDE, OpenOffice.org и Eclipse.

Почему форк?

Рассматривались разные варианты. В итоге остановились на LTSP5 в качестве основы, но были «выкинуты» те части LTSP 5.0, которые были сочтены незрелыми, и втянуты элементы LTSP 4.2, которые доказали свою лучшую пригодность (NFS root и использование XDMCP).

Upstream merge

Результат вскоре был работоспособен, но сильно отличался от основного направления разработки LTSP5. После ряда обсуждений было решено всё-таки попробовать объединить ветки весной 2008, что позволило сократить объём различий примерно вчетверо.

Текущее состояние

ALTSP5 остался таким же «гибридным» вариантом: используется специальная сборка ядра с патчами для надёжного свопа по сети; потребность в памяти терминала получилось свести к сопоставимому с требуемым для LTSP4.2 объёму — 16М.

Доступны снапшоты терминального решения на основе ALT Linux 4.0 Desktop и школьного дистрибутива Линукс Терминал, которые позволяют воспользоваться терминальным сервером непосредственно после установки (при возможности загрузиться по PXE).

Написана документация[1] и система управления клиентами — обе скорее в минимальном объёме: документации чем меньше *приходится* читать — тем лучше, а вот улучшение модуля Alterator — один из главных участков дальнейшей работы.

Планы

Существует намерение разрешить возникший «конфликт поколений» LTSP (в первую очередь по части требований к ресурсам) за счёт реализации поддержки множественных транспортов в стиле LTSP4/5 и расширяя их — для:

- загрузки терминалов (NFS/NBD);
- регистрации в системе (XDMCP/LDM/RDP/...);
- работы с приложениями (X11/X11+ssh/RDP/...);
- работы с локальными устройствами (ltsdfs/RDP/...).

Дистрибутивами по факту всю пользуются и в виде снапшотов, причём они весьма близки к состоянию релиза; работа в направлении выпусков также ведётся.

Контакты

Загрузить бета-версии (и будущие релизы) можно с серверов [ftp.linux.kiev.ua](ftp://linux.kiev.ua) и beta.altlinux.org; подписаться на рассылку, где идёт обсуждение разработки и применения — по адресу <https://lists.altlinux.org/mailman/listinfo/ltsp-server>.

Докладчика можно найти как mike@altlinux.org (email/jabber).

Литература

- [1] Alexander Chumachenko. Ltp5 в altlinux, 2007.
- [2] Media Magic Ltd. Терминал-сервер, 2007.

А. Варламов, С. Забелин, С. М. Мишура

Новосибирск,
Apache Harmony

Проект: Apache Harmony

<http://harmony.apache.org>

Apache Harmony: разработка Java-платформы как свободного ПО

Apache Harmony — это проект с открытым кодом, основной целью которого является разработка полной и совместимой реализации стандартной версии Java-платформы (J2SE) под лицензией Apache (v. 2). Не менее важной целью проекта является разработка модульной архитектуры для библиотек классов и виртуальной машины, которая позволит независимым разработчиками переиспользовать компоненты и откроет путь к инновациям в разработке отдельных компонент.

Проект Harmony стал возможен благодаря изменениям в Java Community Process (JCP), где одним из ключевых инициаторов изменений являлось Apache Software Foundation (ASF). Эти изменения сделали возможным открытую реализацию спецификаций JCP и предоставили возможность бесплатной сертификации для некоммерческих, учебных организаций и частных лиц. К сожалению, переговоры по получению лицензии на сертификационный пакет (Technology Compatibility Kit-ТСК) для самого проекта Harmony пока не завершены.

Началом проекта считается май 2005, когда он был принят ASF для апробации. В октябре 2006 проект был переведён в разряд официальных ASF-проектов. На текущее время разработка библиотеки классов для спецификации J2SE v. 5 почти завершена, также проект имеет виртуальную машину, близкую по функциональности и производительности к коммерческим. Большое внимание уделяется открытости разработки, это включает в себя доступность исходного кода,

прозрачность принятия технических и организационных решений — проект имеет репозиторий кода и базу ошибок, открытые для всех; все дискуссии проходят на общедоступных списках рассылки. Также важным аспектом разработки является следование clean-room процедуре. Эта процедура предназначена для удостоверения, что код свободен от интеллектуальной собственности (IP) третьих лиц, так как в свободном доступе находится большое количество кода для J2SE, в том числе включённого в различные коммерческие реализации.

Огромное значение в проекте уделяется тестированию. Проект имеет хорошо организованную тестовую базу и инфраструктуру, которая помогает разработчикам в кратчайшие сроки получать результаты тестирования и отслеживать стабильность вносимых изменений практически в режиме online. Тестовая база состоит из несколько специализированных тестовых сьюит, которые дополнены тестовыми сьюитами других проектов и автоматизированными сценариями работы различных приложений.

А. П. Щербаков

Томск,
Институт Оптики Атмосферы СО РАН

Разработка макета автономного робота на основе платформы GNU/Linux

Аннотация

Рассказывается о реализованном макете системы управления автономного мобильного робота на основе библиотек GNU/Linux. Робот снабжён системой технического зрения, включающую две видеокамеры и программное обеспечение. Система управления использует как ряд существующих свободно распространяемых, так и разработанных автором математических методов анализа распознавания зрительных образов, и методы трехмерного моделирования зрительных сцен. Робот предназначен для развития и тестирования методов автоматической навигации и ориентации в пространстве на основе видеоинформации.

Макет был создан для отработки математических методов автоматического управления, навигации и ориентации в пространстве.

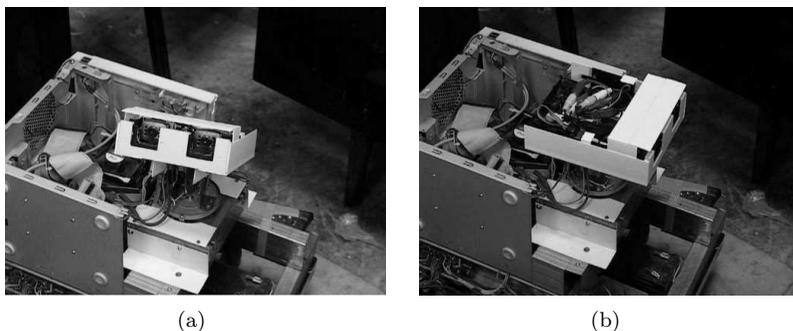


Рис. 1:

Робот имеет подвижное шасси, собранное по трёхколёсной схеме, обычно применяемой в исследовательских вариантах робототехники. На борту установлена подвижная видеосистема, состоящая из двух видеокамер. Видеосистема может поворачиваться в горизонтальной и вертикальной плоскости (Рис 1 а., б.).

Аппаратная часть системы управления построена на базе процессора P4-Celeron-D 2.2МГц под управлением ОС Linux. Процессорный блок установлен на борту робота и питается от аккумулятора через автомобильный инвертор напряжения. Программная часть состоит из блоков, изображённых на *схеме 1*.

Lisp-интерпретатор, реализованный на языке C++, осуществляет постановку и выполнение текущей задачи, и манипуляции с базами знаний и образов.

Поскольку интерпретатор не может обеспечить высокую скорость, процедуры, отвечающие за восприятие и анализ информации от внешней среды, реализованы как встроенные функции интерпретатора.

В макете применены следующие компоненты свободного ПО:

1. Операционная система Alt Linux Compact 3.
2. Компилятор GNU C/C++.
3. Qt — многоплатформенная библиотека для построения пользовательского интерфейса прикладных программ.
4. V4L2 — библиотека для работы с устройствами ввода изображений.

Макет применялся для отработки методов автоматической идентификации объектов [1–3], трёхмерной реконструкции зрительных

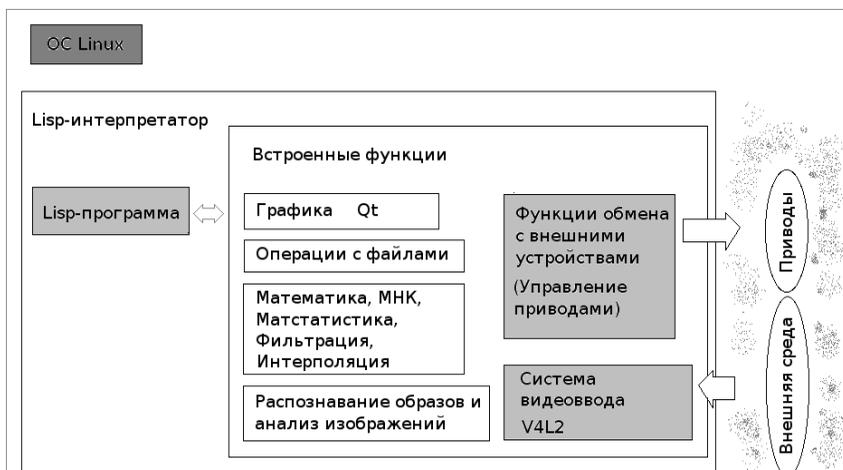


Рис. 2: Схема 1

сцен, обнаружении препятствий и определения расстояния до них, поиска свободного пути проезда.

Литература

- [1] А. П. Щербаков. Быстродействующий алгоритм сегментации изображений. //Автометрия, 2005, №2, стр. 59–67.
- [2] А. П. Щербаков. Быстродействующий алгоритм вычисления положения размера и угла наклона объекта на двумерной зрительной сцене. //Цифровая обработка сигналов, 2004, №4, стр. 15–20.
- [3] А. P. Shcherbakov. Fast identification of two-dimensional patterns in arbitrary position. Space of features. //Pattern Recognition and Image Analysis, Volume 17, Number 4, декабрь 2007, pp. 450–456.

Г. Э. Куликов

Томск,
Институт Оптики Атмосферы СО РАН

Создание автоматизированной системы сбора и обработки данных физического эксперимента на основе платформы GNU/Linux

Аннотация

В работе предлагается структура системы сбора и обработки данных экспериментальных физических установок. В основе предлагаемой схемы лежит жёстко иерархическая модель, использующая простую абстрактную модель измерений, не зависящую от конкретной аппаратуры и методики эксперимента, что позволяет использовать её в качестве основы для построения систем регистрации и обработки данных различных физических установок. Система реализовывалась на платформе GNU/Linux исключительно средствами свободного программного обеспечения.

Современные физические установки (**ЭФУ**) характеризуются большим числом каналов регистрации и высокими скоростями потока данных. Уже к началу 80-х гг. исследования как на больших физических установках, так и на компактных экспериментальных стендах достигли такого уровня, при котором наличие *автоматизированной информационно-управляющей системы сбора и обработки данных (ССОД)* становится необходимым условием функционирования всего комплекса, а сама **ССОД** оказывается одной из наиболее сложных частей экспериментальной установки. Существенно, что любая **ССОД** всегда работает в реальном времени эксперимента и осуществляет стыковку регистрирующей аппаратуры с общей базой данных и системой обработки измерений, причём аппаратное обеспечение может представлять собой распределённую сеть микроконтроллеров или компьютеров [1].

В настоящее время в практику экспериментаторов активно внедряются платформы **ССОД**, самым известным (и практически монопольным) представителем которых является система LabView фирмы National Instruments [2]. Известно, что, используя LabView, можно очень быстро построить качественное ПО экспериментальной установки, и тем не менее, исследователи продолжают создавать собственные реализации. Это связано как со сложностями повторного исполь-

зования кода, неразрывно связанного в случае LabView с представлением графического пользовательского интерфейса, так и с закрытым характером системы, не позволяющим детально проанализировать происходящие процессы. В ряде случаев попытки использования LabView оказываются неудачными именно из-за отсутствия у разработчика внятного представления о работе G-программы LabView.

Целью настоящей работы была попытка создания простой масштабируемой **ССОД**, независимой от вида используемой аппаратуры и специфики **ЭФУ**, допускающей жёсткое разделение частей **ПО** на отвечающие за работу по сбору данных и управление **ЭФУ** с одной стороны, и за интерфейс с пользователем — с другой, и построенной как открытое **ПО** на основе свободных компонент.

Исходя из анализа требований к **ССОД** и общей практики, в качестве платформы для реализации был выбран Линукс (с перспективой использования расширений реального времени). **ССОД** строится согласно модели, в которой задача решается множеством отдельных модулей — процессов, являющихся генераторами и приёмниками данных, взаимодействующими между собой через среду-посредника, доставляющую модулям сигналы и данные. Модули объединяются в цепочки, образующие *схему* **ССОД**, причём «схема» собирается динамически (позднее связывание). Код большинства модулей выполняется асинхронно в отдельных потоках исполнения (тредах). Важно, что посредник может распределять события и данные между узлами сети. Существенные отличия от модели, принятой в LabView, — 1) сознательное ограничение на число и тип передаваемых между модулями событий и данных; 2) отказ от модели «распространения данных» в чистом виде и активная эксплуатация синхронных сигналов. Это позволило резко упростить систему и программирование в ней, отказаться от написания специализированного компилятора: «схема» эксперимента непосредственно представляется в виде совокупности взаимодействующих объектов. Компактность «схемы» эксперимента + обслуживающее ядро позволяют организовывать **ПО** конкретного эксперимента в виде нескольких отдельных процессов, что облегчает программирование и увеличивает надёжность: каждый сепаратный процесс почти всегда может при необходимости быть перезапущен без остановки эксперимента в целом.

Каждая «схема» может общаться со внешним миром строго стандартизированным образом и только посредством специально описанных «портов» (физически представляющих собой блоки разделяемой

памяти и связанные с ними наборы семафоров). Это позволяет с одной стороны, легко стыковать между собой «схемы», разработанные в разное время и для различных целей, а с другой — использовать одну и ту же реализацию графического интерфейса с совершенно разными «схемами» различных экспериментов.

Для облегчения как процесса разработки, так и эксплуатации, система разрабатывалась как межъязыковое приложение, центральную роль в котором играет простая и эффективная система программирования Python.

Литература

- [1] *Олссон Д., Пиани Л.* Цифровые системы автоматизации и управления, в 2-х томах. — СПб.: Невский Диалект, 2001. — С. 557.
- [2] *Тревис Дж.* LabView для всех / пер. с англ. Клушин Н.А. — М.: ДМК Пресс, ПриборКомплект, 2004. — С. 544.

А. И. Привезенцев

Томск,
Институт Оптики Атмосферы СО РАН

Проект: Информационная система «Молекулярная спектроскопия»
<http://wadis.saga.iao.ru/saga2/>

Разработка модуля «Мета+» информационной системы «Молекулярная спектроскопия» для формирования онтологического описания

Аннотация

Рассматривается программная реализация модуля «Мета+» научной Web-ориентированной системы «Молекулярная спектроскопия», позволяющего автоматизировать формирование метаданных в виде онтологического описания в рамках проекта Semantic Web.

Техническое развитие научных сообществ позволило накапливать большие массивы данных. Обработка этих массивов различными специализированными программными инструментами делает возможным получение ещё большего количества новых данных из существующих. Постоянные манипулирования с данными требуют семантиче-

ского описания данных, описания связей с первоначальными данными, сравнения с другими массивами, поиск данных для научных расчётов. Эти массивы хранятся и обрабатываются для формирования научного знания. Для этого в информационной системе (ИС) «Молекулярная спектроскопия» осуществляется создание семантического описания ресурсов в виде онтологий предметной области в рамках проекта Semantic Web[2].

Для формирования, обработки и визуализации онтологического описания информационных ресурсов разрабатывается специальный модуль «Мета+». Так, программная реализация этого модуля осуществлена на серверном скриптовом языке PHP 5. Этот язык был выбран, исходя из следующих посылок:

- ядро существующей информационной системы реализовано на PHP;
- поддержка объектно-ориентированного программирования;
- наличие существующих API для работы с RDF/OWL[3].

Разработка на PHP ведётся в свободной открытой интегрированной среде разработки (IDE) PHP Development Tools, распространяемой под Eclipse Public License и построенной на платформе фреймворка Eclipse. Для программирования на JavaScript используется свободное открытое IDE Aptana, распространяемое под лицензией GPL и также построенное на платформе фреймворка Eclipse. Использование в основе этих двух IDE одной платформы Eclipse позволяет интегрировать их возможности в одну среду разработки.

Модуль «Мета+» использует API для работы с RDF и OWL, распространяемые свободно в виде ПО с открытым кодом. Для работы с RDF используется RDF API for PHP (RAP) [1], распространяемого под лицензией LGPL. RAP — инструмент для PHP-разработчиков, стартовавший как проект с открытым исходным кодом в Берлинском Университете Фрайа в 2002 году и затем расширенный другими разработчиками. В RAP в качестве абстрактного слоя по работе с реляционными базами данных используется API ADODB, который позволяет абстрагироваться от конкретной базы данных.

Для работы с OWL используется API rOWL [4], распространяемого под лицензией GPL и построенного на API RAP. rOWL — Web-ориентированный RDFS/OWL публикатор, браузер, редактор. Проект поддерживается группой немецких учреждений, среди которых, например, — Университет Лейпцига.

Варианты программных библиотек `rOWL` и `RAP` были частично модифицированы в соответствии с требованиями разрабатываемой функциональности.

Существующая система управления базами данных (СУБД) `MySQL` в ИС «Молекулярная спектроскопия» используется для хранения массивов данных и используется с модулем «Мета+» для формирования метаданных при алгоритмическом обчёте этих массивов. Наличие существующей СУБД явилось основой для использования оной в качестве хранилища онтологической базы знаний.

Кроме языка `PHP` для реализации приложений использовался клиентский язык `JavaScript`. Чтобы не создавать собственные типовые функции для работы в асинхронном режиме, использовалась свободная открытая библиотека `jQuery`.

Основным языком для разметки информации использовался язык `XML`. На этом языке создавались конфигурационные файлы, шаблоны для генерации онтологических экземпляров. Сами онтологии создавались используя `XML`-синтаксис `OWL/RDF` моделей.

Разработка модуля «Мета+» велась с использованием тестов для проверки работоспособности функциональности прикладных классов. Каркасом для тестирования программного обеспечения была выбрана сторонняя библиотека `PHPUnit`, представляющая собой свободный фреймворк с открытым исходным кодом для разработки с применением тестирования.

Архитектура модуля «Мета+» тесно связана с существующим ПО научной информационной системы «Молекулярная спектроскопия». Реализация модуля произведена с использованием современных подходов к созданию `Web`-ориентированных приложений, используя существующее свободное открытое ПО.

Литература

- [1] Freie Universität Berlin. `Rap - rdf api for php` home page, 2008.
- [2] World Wide Web Consortium. Semantic web activity home page, 2001.
- [3] F. van Harmelen edited by D. L. McGuinness. Owl web ontology language overview, 2004. W3C Recommendation.
- [4] Universität Leipzig. `powl - semantic web development platform` home page, 2006.

А. Ю. Ахлёстин

Томск, ИОА СО РАН

Проект: ATMOS: научный портал для атмосферных наук

<http://atmos.iao.ru>

Промежуточное программное обеспечение для создания научных информационных систем

Атмосферные науки, науки об окружающей среде, космическая погода, физика, химия, астрономия, инженерные науки, медицина, биология представляют собой источник процедурных и декларативных знаний. Создание автоматизированных информационно-вычислительных систем (ИВС) в этих науках связано с рядом задач, которые должны учитывать междисциплинарный характер исследований (механизм интеграции приложений, обмен структурированными данными и знаниями), коллективные формы работы (публикация результатов исследований в Интернете) и возможность автоматизированной обработки знаний. Перечисленные задачи могут быть решены, если существуют инструментальные средства, доступные разработчикам. В настоящее время существует ряд теоретических подходов для решения задач, но отсутствует программное обеспечение для их реализации.

Современное состояние работ по созданию инструментария характеризуется двумя смежными подходами, использующими в качестве языков этого инструментария PHP или Java. Первая версия, в которой сформированы компоненты инфраструктуры на уровне данных, реализована средствами, наиболее часто используемыми в практике (http-сервер Apache, СУБД MySQL, PHP) [1]. Вторая версия, носящая ниже название Osaka, отличается от первой версии поддержкой информационного уровня и уровня знаний и основана на использовании языка Java, сервлет-контейнера Tomcat, интегрированных сред JSF и STRUTS и СУБД, поддерживающих JDBC, описывается в нашей работе.

Проблемы создания программного обеспечения для научных информационных систем во многом определяются рядом проектов, осуществляемых за рубежом. Эти проекты определяют рамки, в которых должно развиваться программное обеспечение, ориентированное

на создание научных информационно-вычислительных систем.

Среди зарубежных проектов в первую очередь стоит отметить проекты Национального Научного Фонда США в русле создания киберинфраструктуры [7], европейский проект INFRASTR-3 [4] и английский проект e-Science [2]. Именно эти проекты определяют магистральное мышление в постановке и решении задач по построению научных ИВС.

Инструментальная часть, предоставляемая для создателей веб-ИВС, ограничена набором веб-серверов (Apache, IIS, Sun iPlanet), языками программирования (PHP, Python, Java-script, Java) и набором СУБД (Oracle, MySQL, Postgres). Развиваемые в последние пять лет средства под общим названием GRID (Globus) ориентированы на организацию вычислений, а не на создание информационных ресурсов. Ключевую роль за минувшее десятилетие играют рекомендации, принимаемые W3C, среди которых надо выделить предназначенные на создание семантического веба. Отметим монографию В. Оберле раскрывающую некоторые идеи представленные в нашей работе [8].

Типовые решения по созданию инфраструктуры ИС в большей части связаны с системами управления контентом (CMS) (например, [6][3][5]). Нам неизвестны некоммерческие программные продукты, ориентированные на создание инфраструктуры научных ИВС.

Литература

- [1] Атмосфера и окружающая среда. Промежуточное программное обеспечение, 2006.
- [2] e-science initiative.
- [3] Infoglue.
- [4] Infrastr-3.
- [5] Jboss portal.
- [6] Opencms.
- [7] Revolutionizing science and engineering through cyberinfrastructure, report of the national science foundation, 2003.
- [8] D.Oberle. *Semantic management of middleware*. Springer, 2006.

М. С. Пожидаев

Томск,
Томский государственный университет

Проект: Sisyphus

<http://sisyphus.ru>

Обзор компонентов и возможностей средств для работы без зрительного контроля в дистрибутивах Alt Linux

Основной средой для работы в дистрибутивах ALT Linux без зрительного контроля является emacspeak. В настоящее время emacspeak позволяет выполнять следующие операции:

- манипуляции с файлами;
- чтение электронной почты;
- обзор WWW-страниц (без поддержки Java Script);
- запуск консольных приложений (без поддержки ncurses);
- удобное редактирование исходных текстов программ;
- быстрый вызов компилятора и нахождение ошибок;
- ведение календаря и записных книжек;
- прослушивание книг из текстовых файлов.

Некоторые операции хорошо доступны в силу того, что они могут быть выполнены при помощи консольных инструментов. Например:

- ведение издательской деятельности и вёрстка физико-математических текстов (tetex-latex);
- вёрстка нотных текстов (lilypond);
- подготовка музыкальных записей на основе нотного текста (lilypond + timidity);
- подготовка и запись CD/DVD-дисков (mkisofs, cdrecord, growisofs) и др.

К достоинствам системы emacspeak относится высокая скорость работы незрячего человека, практически недостижимая при использовании ведущих коммерческих пакетов для Microsoft Windows.

Основной пакет для вывода речи — речевой сервер VoiceMan. Такой компонент нужен в силу:

- необходимости организации одновременной работы нескольких приложений с недопущением одновременного чтения разного текста;
- необходимости работы с несколькими языками;
- отделения программ-чтецов от особенностей разных речевых синтезаторов.

Речевой сервер VoiceMan способен обрабатывать и удалённые подключения по сети.

Основные синтезаторы, которые сейчас доступны, — это mbrola для английского языка и ru_tts — для русского. Оба синтезатора не имеют опубликованных исходных текстов, хотя их лицензии допускают размещение в публичных репозиториях и использование в некоммерческих целях. Также подготовлены к работе синтезаторы flite, festival, espeak. Синтезаторы festival и espeak имеют частичную реализацию поддержки русского языка, но недостаточную для их непосредственного использования в работе.

В ближайших планах — подготовка работы пакета ogss и реализация удобных средств конфигурирования речевого вывода при помощи платформы «Альтератор».

Е. В. Лапин, Д. Н. Бараксанов, Д. А. Лихачёв, А. Лёвин
Томск, ТУСУР

Опыт внедрения открытой CMS

Аннотация

Типовые задачи создания и сопровождения информационных сайтов и способы их решения. Положительный опыт внедрения OpenCms в администрации Томской области и ТУСУРе в качестве платформы управления содержимым сайтов.

Постановка задачи

Информационные сайты, как правило, обладают типовой структурой и функциональностью. Часто встречаемые задачи:

- управление динамической структурой сайта;
- редактирование страниц сайта с использованием визуального редактора;

- управление фотогалереями;
- регулярная публикация новостей;
- размещение анонсов мероприятий (анонс теряет актуальность после наступления события и должен быть удалён);
- организация обратной связи;
- полнотекстовый поиск по сайту;

Существует большое количество открытых систем управления содержанием, позволяющих решить поставленные задачи.

Немного об OpenCms

После анализа свободных программных продуктов для решения задачи была выбрана система управления содержанием OpenCms (<http://www.opencms.org>).

Наиболее важными особенностями OpenCms являются:

- удобный веб-интерфейс управления содержанием и администрирования;
- возможность создания новых типов структурированного контента (новости, анонсы) на основе XSD-схем с автоматическим построением административных веб-форм;
- применение шаблонов отображения в зависимости как от типа контента, так и от адреса страницы;
- редактирование неструктурированного контента с использованием визуального WYSIWYG редактора;
- возможность функционирования нескольких сайтов на одной инсталляции системы;
- модульная архитектура, позволяющая легко расширять функциональность;
- представления каждого сайта в виде отдельного модуля, который может быть легко перенесён с одной инсталляции системы на другую;
- поддержка любого количества языковых версий сайта;
- хранение нескольких версий контента;
- гибкое разграничение прав доступа;
- возможность использования одного контента в разных частях сайта и даже на разных сайтах (в пределах одной инсталляции);
- полнотекстовый поиск;
- улучшение производительности за счёт статического экспорта страниц, кэширования динамических частей страниц и кластеризации.

Положительный опыт внедрения

Положительный опыт внедрения OpenCms имеется как в органах государственной власти, образовательных учреждениях, так и в коммерческих организациях.

В администрации Томской области функционируют:

- Интернет-портал Администрации Томской области (<http://tomsk.gov.ru>);
- Сайт департамента по культуре (<http://depculture.tomsk.gov.ru>);
- Здравоохранение в Томской области (<http://zdrav.tomsk.ru>);
- Интернет-справочник «Государственные услуги населению Томской области» (<http://gus.tomsk.ru>);
- Интернет-портал «Мультиязычный инвестиционный паспорт Томской области» (<http://investintomsk.com>);
- Сайт «Равнение на Победу» (<http://pobeda.tomsk.ru>).

В ТУСУРе:

- Официальный информационный портал ТУСУРа (<http://www.tusur.ru>);
- Информационный центр абитуриентов (<http://ica.tusur.ru>);
- Сайт заочного и вечернего факультета (<http://zvf.tusur.ru>);
- Сайт кафедры АОИ (<http://aoi.tusur.ru>);
- Электронная версия журнала «Радиоэлектроник» (<http://re.tusur.ru>);
- Сайт областного центра новых информационных технологий (<http://toznit.tusur.ru>);
- Сайт инновационной образовательной программы (<http://innovation.tusur.ru>).

Другие сайты:

- Сайт кафе Томичка (<http://tomichka.tomsk.ru>);
- Сайт Центра коммуникаций (<http://centercom.tomsk.ru>);
- Сайт мастера татуировки (<http://tattoo-art.tomsk.ru>).

Выводы

Основываясь на опыте использования OpenCms, можно сделать вывод, что система очень хорошо подходит для создания и сопровождения типовых информационных сайтов. Однако реализации специфических задач, таких как персонафикация, интерактивные сервисы, сложная бизнес-логика, может вызвать затруднения.

С. В. Кратов

Новосибирск, ИВМиМГ СО РАН

Организация взаимодействия пользователей распределённых систем на базе свободного ПО

Аннотация

В Сибирском Отделении РАН создаётся информационно-вычислительная система поддержки инновационной деятельности [1]. В отличие от существующих, предлагаемая система должна обеспечить активную информационную поддержку: организацию и ведение инновационного проекта. Дополнительную информацию о системе можно получить на сайтах проекта: <http://portal.sccc.ru> и <http://sinin.nsc.ru>. Целью данного доклада является описание реализации способов коммуникации пользователей в рамках подобной системы. Взаимодействие пользователей обеспечивается одним из модулей системы — коммуникатором, основанным на базе свободного ПО и создающим единую среду общения. Общение включает полный спектр способов коммуникации — от оперативных до асинхронных — и базируется на таких сервисах, как Jabber-сервер, IRC-сервер с Web IRC-клиентом, форум, почтовый Web-клиент и т. д.

Одной из частей коммуникатора нашей сетевой системы поддержки инновационной деятельности является IRC-сервер — сервисная система, при помощи которой пользователи портала смогут общаться через сеть Интернет в режиме реального времени. В качестве IRC-сервера в нашей системе используется UnrealIRCd. Выбранный IRC-сервер обладает следующими преимуществами:

¹Работа выполнена при поддержке ИП РАН 15.2.

- поддержка модулей — возможно создание модулей с новыми командами, режимами пользователей и каналов;
- поддержка SSL соединений;
- поддержка IPv6;
- поддержка сжатых соединений между серверами;
- возможности защиты от flood'a;
- наличие спамфилтра;
- поддержка CGI:IRC.

Подключиться к нашему IRC-серверу можно с использованием любого IRC-клиента, используя следующие настройки: имя сервера — `portalx.sssc.ru`, порт — `6667`. Также на нашем сервере работает Web IRC-клиент — CGI:IRC, доступный по следующему адресу: <http://portalx.sssc.ru/cgi-bin/cgiirc-0.5.9/irc.cgi>.

Ещё одним компонентом коммуникатора является Jabber-сервер. Jabber — это открытый протокол для быстрого обмена сообщениями между любыми двумя абонентами в Интернет. Jabber имеет ряд преимуществ по сравнению с коммерческими системами IM, таких как открытость, расширяемость, децентрализованность, безопасность и др. В качестве Jabber-сервера в нашей системе используется Wildfire. Данный сервер обладает следующими основными функциями:

- полное администрирование через Web-консоль;
- кроссплатформенность (Java);
- интеграция с LDAP и возможность легкой интеграции с любой системой идентификации пользователей;
- перевод на множество языков;
- поддержка плагинов.

Подключиться к нашему Jabber-серверу можно с использованием любого Jabber-клиента, используя следующие настройки клиента: имя сервера — `portalx.sssc.ru`, порты — `5269` или `5222`, порт для ssl-подключения — `5223`.

Также перед нами стояла задача выбора форума для нашего проекта. Для тестирования были отобраны несколько форумов, наиболее распространённых в данный момент в русскоязычном сегменте сети Интернет, — `vBulletin`, `Invision Power Board`, `UBB`, `phpBB`, `Ikonboard` и др. В процессе тестирования скрипты оценивались по следующим параметрам:

- бесплатность форума, его разработка на основе открытых исходных кодов;

- возможность самостоятельного добавления в форум необходимых модификаций и наличие большой базы уже созданных пользователями модификаций;
- высокая скорость и стабильность работы;
- возможность работы с несколькими языками.

По совокупности параметров было решено остановиться на форуме phpBB.

В состав коммуникатора входит и почтовый Web-клиент. С его помощью пользователи системы посредством браузера могут получить доступ по POP3 и IMAP протоколам к любому своему удаленному почтовому серверу. В качестве Web-клиента в системе используется скрипт PoMaMail. Выбранный Web-клиент обладает следующими основными преимуществами:

- обладает полной функциональностью по сравнению с обычными локальными почтовыми клиентами;
- не требует для работы СУБД, но в случае необходимости может использовать MySQL для повышения производительности;
- в процессе работы потребляет мало вычислительных ресурсов сервера;
- имеет собственную встроенную IMAP/POP3 библиотеку;
- осуществляет фильтрацию спама;
- присутствует возможность работы с несколькими языками;
- допускает расширение функций путем подключения дополнительных модулей;
- поддерживает внешние SMTP-серверы.

Используемый почтовый Web-клиент доступен по следующему адресу: <http://portalx.sssc.ru/webmail/source/index.php>.

Все использованные в коммуникаторе компоненты является бесплатными, многие разрабатывается на основе открытых исходных кодов.

Литература

- [1] Алексеев А. С., Кратов С. В., Остапкевич М. Б., Пискунов С. В. Сибирская сетевая система поддержки инновационной деятельности // «Научный сервис в сети Интернет: многоядерный компьютерный мир. 15 лет РФФИ». Труды Всероссийской научной конференции (24–29 сентября 2007 г., г. Новороссийск). — М.: Изд-во МГУ, 2007. — С. 352–354.

В. В. Сергеев, С. Ю. Коростелев, С. Г. Псахье

Томск,

Институт физики прочности и материаловедения, СО РАН

Проект: Разработка программного обеспечения для моделирования методом частиц

Приложение «Analyzer 3D» для обработки и анализа результатов моделирования методом частиц

Аннотация

В данной работе описывается приложение «Analyzer 3D», предназначенное для графической обработки результатов моделирования. Оно позволяет проанализировать структуру модели и такие характеристики частиц, как скорости движения, приложенные силы, углы вращения, скорости вращения, смещения и траектории движения. Также представлены основные возможности для работы с трёхмерными объектами.

Важным этапом при моделировании физических процессов является обработка полученных результатов. От качества их обработки зависит правильность принятия последующих решений. Лучше всего человек воспринимает информацию, представленную визуально. Следовательно, для улучшения восприятия результатов неотъемлемой частью является их графическое представление.

Большинство систем моделирования физических процессов представляют свои результаты в виде набора файлов. В каждом файле описывается состояние модели на определённом этапе эволюции. Другими словами, отдельно взятый файл соответствует определённому шагу процесса моделирования.

Приложение «Analyzer 3D» позволяет графически отобразить файлы с результатами и визуально проанализировать эволюцию модели.

Просмотр файлов (шагов) моделирования осуществляется по списку. Для его создания, редактирования и сортировки в составе приложения имеются все необходимые средства.

Работа в трёхмерном пространстве накладывает определённые трудности на анализ модели. Для удобства просмотра трёхмерных

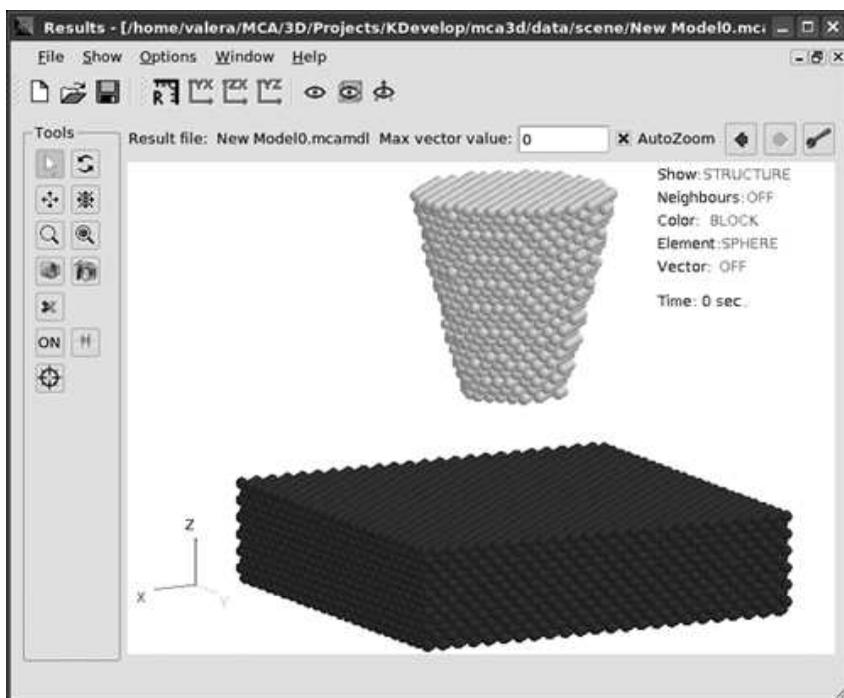


Рис. 1: Рабочая область приложения

объектов реализованы специальные утилиты. Их управление осуществляется с помощью различных менеджеров, позволяющих задавать различные параметры. Одной из утилит является позиционирование «центра сцены». «Центром сцены» является точка в пространстве, относительно которой осуществляются всевозможные трансформации (поворот, масштабирования и т. д.). При загрузке модели «центр сцены» по умолчанию помещается в центр трёхмерного объекта.

Другой утилитой является установка набора секущих плоскостей. При работе с 3D-телами часто бывают ситуации, когда интересующая нас часть объекта недоступна для просмотра (находится внутри объекта или закрыта другой его частью). Менеджер сечений позволяет вывести для просмотра только необходимую часть объекта. При этом

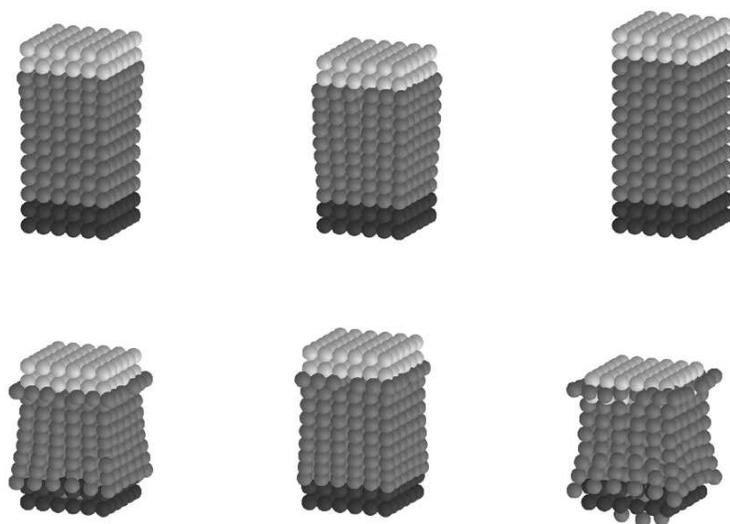


Рис. 2: Моделирование процесса сжатия образца

возможно использование утилиты позиционирования «центра сцены» для установки его в центре выводимой части объекта.

При использовании утилиты «режим линеек» можно проецировать объект на двухмерные плоскости, что позволяет измерять расстояние между любыми двумя точками плоскости, а также определять координаты и размеры какой-либо частицы или части объекта.

Все параметры вида и список просматриваемых файлов можно загрузить или сохранить в специальном файле с расширением «.mcarlt».

«Analyzer 3D» позволяет увидеть структуру модели и различные характеристики частиц, такие как скорости движения, приложенные силы, углы вращения, скорости вращения, смещения и траектории движения.

Частицы могут быть изображены сферами или точками. Радиус устанавливается при загрузке модели программой или задается пользователем.

При отображении векторных полей (скорости, силы и т. д.) применяется автоматическое масштабирование векторов по максимальному значению в загружаемом файле. Данный коэффициент масштаба

можно зафиксировать для сравнения изменений в различных файлах. Также применяется различная цветовая раскраска частиц и векторов: по материалам, по принадлежности к блокам, по значению (в случае векторной величины). При наличии информации о связях между соседними частицами (например, в результатах системы моделирования методом подвижных клеточных автоматов «МСА 3D») существует возможность просмотра сетки связанных и несвязанных частиц. Эти средства позволяют детально проанализировать эволюцию модели. Сейчас «Analyzer 3D» работает с форматами данных «lammps» и «МСА 3D», но его можно использовать для просмотра любых трёхмерных моделей, состоящих из набора частиц.

Интерфейс использует свободно распространяемые библиотеки Qt и OpenGL. Они являются кросс-платформенными, что позволяет использовать «Analyzer 3D» в операционных системах Linux и Windows. Благодаря OpenGL на данный момент приложение способно работать с моделями, содержащими большое число частиц (более миллиона), на персональном компьютере средней производительности. Данное приложения широко применяется при обработке результатов систем моделирования методами подвижных клеточных автоматов («МСА 3D») и молекулярной динамики («Monster MD»).

Ю. А. Поплавский, А. П. Щербаков, Г. Э. Куликов

Томск,
ИАО СО РАН

Использование распределённой сетевой архитектуры и платформы GNU/Linux в непрерывных спектрофотометрических анализаторах

Аннотация

К обсуждению предлагается архитектура программного обеспечения спектрофотометрических анализаторов, работающих в непрерывном режиме. Описываемое решение реализовано на платформе GNU/Linux в настольном и встраиваемом (ARM) вариантах. В различных режимах были проведены испытания дифракционных спектроанализаторов при измерениях параметров растворов и газов по спектрам поглощения.

Существует достаточно обширный класс приборов, в которых информация о параметрах и свойствах вещества извлекается из их спектров. Применение спектроскопии в сочетании с математической обработкой данных позволяет решить широкий круг прикладных задач, связанных с определением различных характеристик окружающей среды и технологических процессов. С развитием вычислительной техники, сетевых технологий, интенсивной компьютеризацией исследований, возникает ситуация, когда использование приборов, в том числе действующих по принципам «старой» классической аппаратуры [2], позволяет принципиально улучшить их параметры и значительно расширить области применения. В настоящее время всё более значимым становится возможность работы спектральных приборов в непрерывном автоматическом режиме. В таких задачах программные и аппаратные средства автоматизации сбора данных и управления начинают играть определяющую роль. Очевидно, что выбор общедоступного свободного программного обеспечения (ПО) позволяет удешевить и повысить надёжность разработки научного ПО.

В общем случае при использовании спектрофотометрических анализаторов для решения конкретной прикладной задачи необходимо выполнение определенных действий, связанных с разработкой алгоритмов и программ:

1. Определение и уточнение особенностей используемого спектрофотометра и физических алгоритмов получения информации из регистрируемых спектров;
2. Разработка алгоритмов считывания, распределения и первичной обработки данных, в том числе при работе в сетевом окружении, а также выбор и программирование аппаратных средств и оборудования для считывания спектральной информации;
3. Разработка алгоритмов приведения спектров к стандартному виду по частоте и интенсивности;
4. Создание приборных баз данных по измеряемым параметрам и поиск функциональных или калибровочных зависимостей [3].

Из требования обеспечения непрерывности измерений и работы со многими приборами в сетевом окружении следуют задачи:

1. Обеспечения непрерывной передачи измеренных величин в информационную базу данных;

2. Обеспечение пользователя сетевым доступом к информации о параметрах окружающей среды и технологических процессов, полученных по спектрам поглощения растворов и газов [1].

В работе описывается вариант архитектуры непрерывных анализаторов, отличающийся:

1. Формированием программно-аппаратного комплекса в виде взаимодействующих по определённому протоколу программ;
2. Вынесением кода, непосредственно работающего с аппаратурой, в отдельную службу так, что программы более высокого уровня оказываются нечувствительны к типу используемых измерителей;
3. Использованием для связи всех компонент комплекса протокола дейтаграмм, что позволяет организовать прозрачную работу как в локальной, так и в сетевой среде;
4. Использованием отдельных служб для связи с сервером баз данных и реализации WEB-интерфейса.

В разрабатываемых приборах было использовано следующее свободное ПО:

1. Операционные системы AltLinuxMaster 2.4 и Compact 3.0;
2. Компилятор GNU C/C++ и интерпретатор Python 2.4;
3. Многоплатформенный фреймворк WxWidgets.

На текущий момент в качестве аппаратной части системы сбора данных с приборов испытан контроллер семейства ARM9 под управлением ОС Linux с ядром серии 2.6. Разработка произведена с использованием пакета Arm-linux-gcc-4.1.1. Этот же пакет использован для программирования микроконвертера ADuc7022 от фирмы Analog Devices, имеющего ядро ARM7-TDMI. Микроконвертер использовался в качестве прецизионного скоростного АЦП, подключенного к контроллеру ARM9.

В заключение следует отметить, что существующее свободное ПО оказывается достаточным для решения широкого круга научно-практических задач и позволяет заметно облегчить создание сложного программного обеспечения, в том числе спектроскопических приложений.

Литература

- [1] <http://asd.iao.ru/pya/far/>
- [2] Поплавский Ю.А., <http://asd.iao.ru/pya/HTML/R1.html>
- [3] Поплавский Ю.А. Синица Л.Н. Щербаков А.П. Поплавский Ю. А., Сердюков В.И. Патент РФ №2002123665/28. 2008.

Н. А. Лаврентьев

Томск,

Институт Оптики Атмосферы СО РАН

Проект: Информационно-вычислительная система «Климат»
<http://climate.atmos.iao.ru>

Климатические модели в Интернет-доступной ИВС

Аннотация

В данной работе описывается информационно-вычислительная система «Климат», доступная в сети Интернет. ИВС ориентирована на выполнение вычислений с помощью региональных климатических моделей MM5 и WRF, а также глобальной модели IBM РАН. Разработанная ИВС использует четырёхуровневую архитектуру «клиент-сервер», где запрос клиента выполняется на веб-сервере, вычисления производятся на высокопроизводительном кластере, а входные и выходные данные хранятся на выделенном сервере баз данных.

В настоящее время при исследовании и прогнозе локальных атмосферных процессов, протекающих над ограниченной территорией, широко используются различные метеорологические модели, компьютерная реализация которых связана с применением нетривиальных вычислительных алгоритмов и высокопроизводительных вычислительных ресурсов.

Использование этих моделей в исследовательской и прикладной работе ограничено, с одной стороны, недостаточным распространением необходимых вычислительных ресурсов, а с другой стороны, — отсутствием необходимого числа специалистов. Поэтому актуальным является создание информационно-вычислительной системы (ИВС) доступной в сети Интернет.

Информационно-вычислительная система «Климат» должна, с одной стороны, продемонстрировать возможности математического моделирования для решения отдельных климатических задач, а с другой стороны, — на основе технологий глобальной сети и средств графической визуализации осуществлять взаимодействие с потребителями информации о результатах решения таких задач.

ИВС построена с использованием средств портала ATMOS, использующего в качестве основы инфраструктуру веб-сервера LAMP, все составляющие которой (Linux, Apache, MySQL, PHP) представляют из себя свободное программное обеспечение.

В области региональных климатических моделей одна из вычислительных компонент системы представляет собой мезомасштабную модель MM5 [4] (<http://climate.atmos.iao.ru/star/mm5>), разработанную в Национальном центре атмосферных исследований США, а другая — модель WRF [3] (<http://climate.atmos.iao.ru/star/wrf>), поддерживаемую в текущее время мезомасштабными метеорологическими подразделениями NCAR. Обе эти модели имеют реализации, рассчитанные на работу в высокопроизводительных вычислительных системах параллельной архитектуры. ИВС также включает в себя глобальную климатическую модель, разработанную в Институте вычислительной математики Российской академии наук (<http://climate.atmos.iao.ru/clim>).

Для мезомасштабных моделей MM5 и WRF ввод пользователем данных происходит стандартно в четыре этапа:

1. Инициализация рельефа, категорий землепользования и растительности;
2. Задание стационарных географических параметров выбранных областей;
3. Подготовка данных по временному периоду и уровням давления;
4. Инициализация физических параметров модели.

На основе данных, введённых на этих этапах, создаётся набор конфигурационных файлов модели, после чего параллельная версия модели ставится в очередь на выполнение на вычислительном кластере Института оптики атмосферы. Постановка задач в очередь и управление очередью осуществляется с помощью менеджера Torque[2].

Так как вычислительные модули программы зависят от количества вложенных областей, выбранных пользователем для исследования, перед решением каждой задачи мезомасштабные модели при-

ходится перекомпилировать. В зависимости от настройки, проводимой администратором кластера, в вычислениях могут использоваться от 2 до 20 процессоров (на текущий момент — 10, пять двухпроцессорных узлов).

После выполнения задачи и получения выходных файлов модели, пользователю предоставляется возможность просмотра полученных результатов по каждой вычисленной величине и в любой момент расчётного интервала в графическом виде, используя для этого систему анализа и отображения данных GrADS[1]. Также возможно составление коллекций графиков, позволяющих отслеживать изменение выбранных пользователем переменных по времени и сигма-уровням высоты.

Литература

- [1] Grid analysis and display system (grads).
- [2] Torque resource manager 2.0.
- [3] The weather research and forecasting model.
- [4] Dudhia J. Mesoscale model 5. *Monthly Weather Review*, 121:1493–1513, 1993.

А. А. Вичугова, В. Вичугов

Томск,

Томский политехнический университет

Проект: Лицензионное программное обеспечение для образования,
свободное и бесплатное

<http://soft.mod-site.net>

Опыт использования СПО в учебном процессе технического ВУЗа

Аннотация

На сегодняшний день перед многими предприятиями и учебными заведениями остро стоит проблема лицензирования программного обеспечения (ПО). Высокая стоимость лицензии на использование программных продуктов вынуждает обратиться к поиску «альтернативного софта» или так называемого «свободного ПО (СПО)». В России

это понятие становится всё более популярным, поскольку из-за многозначности английского слова «free» под данным термином иногда понимают бесплатность программ. Но лицензии бесплатных и свободных программ принципиально отличаются друг от друга: свободное ПО подразумевает лишь доступность программ для всеобщего использования, критики и улучшения, но не гарантирует бесплатность.

Для снижения расходов на программную составляющую ИТ-сектора современного предприятия или образовательного учреждения эту проблему можно решить путём замены дорогостоящих лицензионных продуктов на их бесплатные аналоги, которые также являются лицензионными, но не предусматривают платы за их использование. В докладе приведён опыт применения СПО в учебный процесс технического ВУЗа, а также представлена таблица соответствия платных систем и их свободных и бесплатных аналогов.

Как известно, программным ядром компьютера является его операционная система (ОС), которая определяет параметры остальных приложений. В соответствии с концепциями свободного ПО оптимальным и единственно верным является переход с популярной коммерческой ОС Windows на бесплатные Unix-подобные системы: различные версии Linux, BSD и т. д. с удобным и интуитивно понятным графическим интерфейсом, ориентированным на пользователя, который привык работать с продуктами семейства Windows. Как правило, подобные дистрибутивы содержат не только саму ОС, но и базовый набор офисных программ — текстовые, табличные и графические редакторы, архиваторы, файловые менеджеры, калькуляторы, утилиты для работы с периферийными устройствами и различные драйверы. Более того, широкий выбор дополнительного ПО для этих систем является также свободным и бесплатным.

Но иногда в силу специфики какой-либо области, например, привязки к конкретному программному продукту и т. д., такой радикальный шаг невозможен. В этом случае, оставаясь на Windows-платформе, следует найти альтернативы дорогим системам, не уступающие им по качеству, но использование которых не требует взлома и ни в коей мере не нарушает чьи-либо авторские права, а является абсолютно лицензионным и правомочным. Практически для всех видов платного ПО — от операционных систем до различных специальных прикладных программ, включая офисные, графические, математические и САД-пакеты, а также инструменты администрирования, просмотра и обработки мультимедийной информации и т. д., — существуют лицензионные бесплатные аналоги.

Для предприятий важным пунктом в решении задачи администрирования пользовательских машин является наличие графического инструмента удалённого управления а также антивирусная защита. Конкретно в нашем случае для обеспечения учебного процесса в техническом ВУЗе при высокой плотности рабочих станций на базе ОС Windows для их удалённого администрирования была выбрана программа UltraVNC под лицензией GNU GPL, позволяющая распределять права по удалённому доступу согласно записям в Active Directory Windows-сервера. Офисные задачи успешно решаются известным пакетом от «Инфра ресурс» — OpenOffice.org. Широкое разнообразие свободных и бесплатных средств разработки, таких как NetBeans, продукты Borland Turbo Explorer, также облегчает переход на лицензионное ПО. В качестве замены «закрытым» коммерческим продуктам достойно показали себя электронный словарь JaLingo, пакет инженерных расчётов Octave, графические редакторы: Gimp, Inkscape, Blender.

Однако вследствие отсутствия цельности во взглядах российского общества на вопросы незаконного использования лицензионного ПО и недостаточно высокого уровня правовой культуры в сфере IT возникают некоторые парадоксы. Например, студенты пишут отчёт по лабораторной работе в OO Writer, но сохраняют его в закрытом стандарте от Microsoft и распечатывают его с помощью MS Office, поскольку на компьютере в пункте печати присутствует только этот редактор.

Кроме бесплатности свободных программ, именно их свобода, т. е. возможность анализа и модификации исходных кодов, является неоспоримым достоинством для использования их в учебном процессе, поскольку участки исходников реально работающих приложений и сервисов предлагаются студентам в виде задач в курсах дисциплин по программированию. Это делается преподавателями для того, чтобы студент — будущий IT-специалист — по окончании обучения стал именно специалистом, т. к. образование должно быть практико-ориентированным.

Проанализировав основные запросы на функциональность ПО для домашнего и учебного использования, авторы произвели поиск подобных альтернатив платному ПО для операционных систем (ОС) семейства Windows, представленные далее в таблице 1. В результате этого были найдены, апробированы и внедрены в учебный процесс многие программные продукты, а также создан сайт <http://soft.mod->

Таблица 1: Бесплатные аналоги наиболее популярных приложений
(для ОС семейства Windows)

Назначение	Платные программы	Бесплатные аналоги
Удаленное администрирование	Radmin	UltraVNC
Чистка реестра	PowerTools (RegCleaner)	CCleaner
Антивирусы	Nod32, Kaspersky Anti-virus, Norton Anti-virus	ClamWin Free Antivirus, RemoveIT
Архиваторы	WinRar	7-Zip, IZArc
СУБД	MS SQL	MySQL + MySQL-Front/ HeidiSQL / SQLyog / Aqua Data Studio
	MS Access	OpenOffice.org Base
Растровая графика	Adobe Photoshop	Gimp
Векторная графика	Corel Draw	Inkscape
Трехмерная графика и анимация	3D Studio Max	Blender
Офисные программы	MS Office	OpenOffice.org
Распознавание текста	ABBY FineReader	OCR CuneiForm
Переводчики	ABBY Lingvo	JaLingo, Dicto
Проектирование печатных плат	P-CAD	DipTrace*
Моделирование систем	Matlab	Octave, VisSim*
Просмотр графики	ACDSee	FastStone MaxView
Мультимедиа-проигрыватели	NVIDIA, Winamp	VideoLAN — VLCMediaPlayer
	Turbo Pascal	Free Pascal, Pascal ABC
	Borland Developer Studio	Borland Turbo Explorer
Среды разработки	Idea	C++ and Delphi
	Turbo Prolog	NetBeans, Eclipse
Текстовый редактор	Aditor	SWI Prolog
Файловый менеджер	Total Comander	PsPAD, Notepad++
Запись CD-DVD	Nero	Far
Эмулятор CD-DVD	Alcohol	Small CD-Writer
		DAEMON Tools

* — программа бесплатна для домашнего и учебного использования.

site.net, посвящённый данному направлению. На сайте представлены не только описания и скриншоты программ, но также дистрибутивы для скачивания и советы по установке и настройке. При создании этого сайта были использованы только лицензионные, свободные и бесплатные программные продукты.

Надеемся, представленная информация поможет Вам решить часть проблем, связанных с переходом на лицензионное ПО.

Литература

- [1] Википедия — электронная энциклопедия [Электронный ресурс] — Режим доступа: <http://ru.wikipedia.org>
- [2] Открытые системы [Электронный ресурс] — Режим доступа: <http://www.osp.ru/os/2007/01/39992>
- [3] Корпорация Microsoft. Лицензирование [Электронный ресурс] — Режим доступа: http://www.microsoft.com/rus/licensing/general/examples/winxp_pro_sp2.msp

Д. Н. Бараксанов, Е. В. Лапин, Д. А. Лихачёв, А. Лёвин
Томск, ТУСУР

Свободные решения для построения веб-пространства университета

Аннотация

В докладе представлена общая концепция построения веб-пространства университета, а также конкретные решения создания веб-пространства на основе использования СПО.

Общие положения

Веб-пространство университета состоит из нескольких сайтов, направленных на удовлетворение информационных потребностей пользователей (студенты, аспиранты, преподаватели, научные работники, сотрудники, школьники и их родители, абитуриенты и др.).

Создание веб-пространства ТУСУРа предполагало решение трёх основных групп задач:

1. Создание информационных сайтов университета, факультетов и кафедр, подразделений.
2. Создание образовательного портала, предоставляющего доступ пользователям ТУСУРа к базам данных нормативно-правовой документации и электронных ресурсов.
3. Обеспечение механизма единого поиска.

Информационные сайты

Для создания информационных сайтов используется система управления содержимым OpenCms (<http://www.opencms.org>).

OpenCms распространяется свободно, с открытым исходным кодом, и не требует никаких лицензионных отчислений. Система построена на хорошо зарекомендовавших себя Java-технологиях и имеет большое количество внедрений.

На одной инсталляции OpenCms возможна реализация нескольких обособленных сайтов. Так, например, в настоящий момент на одной инсталляции OpenCms функционируют официальный информационный портал университета, информационный центр абитуриентов, а также некоторые сайты факультетов, кафедр, структурных подразделений университета и тематические сайты. Каждому сайту соответствует модуль OpenCms, который содержит шаблоны пользовательского интерфейса, а также всё информационное наполнение, что даёт возможность лёгкого переноса сайта с одного сервера OpenCms на другой. Кроме того, OpenCms позволяет обеспечить информационную интеграцию сайтов, функционирующих на одной инсталляции. Например, информация в разделе факультета ЗиВФ на официальном информационном портале университета (<http://www.tusur.ru/ru/faculties/zvf/>) полностью берётся с официального сайта ЗиВФ (<http://zvf.tusur.ru>).

Образовательный портал

Основной задачей создания образовательного портала является обеспечение доступа в режиме «единого окна» ко множеству независимых веб-приложений. Для решения этой задачи была выбрана портал-портлетная технология. В качестве спецификации портал-портлетной технологии выбрана JSR168. Выбранная спецификация определяет такие понятия, как портал, портлет, портлет-контейнер, а также описывает механизмы взаимодействия между ними [1].

Основываясь на выбранной спецификации и анализе решений, удовлетворяющих этой спецификации, в качестве интеграционной платформы был выбран JBoss Portal (<http://www.jboss.com/products/platforms/portals/>). JBoss Portal предоставляет открытое, основанное на стандартах, порталное решение.



Рис. 1: Структура каталогов образовательного портала

Основной функционал, доступный пользователям, реализуется отдельными веб-приложениями, реализованными в виде портлетов.

JBoss Portal включает в себя портлет-контейнер и поставляется в стандартной конфигурации с портлетами, позволяющими регистрировать и авторизовывать пользователей, управлять информационным содержанием портала, администрировать портал. JBoss Portal позволяет создавать собственные темы и разметки оформления портала.

Структура каталогов образовательного портала представлена на рисунке 1.

Для внедрения автоматической проверки статуса студентов в общеузовской базе потребовалось доработать стандартный портлет регистрации и авторизации пользователей.

Для удобства получения необходимой информации в каждом каталоге реализован механизм полнотекстового поиска.

Системы администрирования каталогов реализованы в виде отдельных веб-приложений и недоступны на портале.

Образовательный портал ТУСУРа доступен по адресу <http://portal.tusur.ru>.

Система единого поиска

Система единого поиска реализована на базе системы Nutch (<http://lucene.apache.org/nutch/>). Apache Nutch представляет собой программный комплекс для организации поисковых серверов. Nutch базируется на библиотеке для построения движков полнотекстового поиска Apache Lucene (<http://lucene.apache.org/>), добавляя при этом дополнительные веб-ориентированные возможности, такие как веб-краулер (поисковый робот), парсер HTML и других типов документов. Nutch построен на основе модульной архитектуры, что позволяет настроить его для решения практически любых поисковых задач [2].

В настоящее время поисковые индексы обновляются один раз в сутки (обрабатываются 9 сайтов).

Система единого поиска доступна по адресу <http://search.tusur.ru>.

Литература

[1] Jsr 168: Portlet specification.

[2] Nutch.

И. А. Хахаев

Санкт-Петербург, ГОУ ВПО СПбТЭИ

Опыт эксплуатации ALT Linux Desktop Edu

Аннотация

Описываются особенности специализированного дистрибутива ALT Linux Desktop Edu, разработанного для поддержки учебного процесса в экономическом вузе. Обсуждаются достоинства и недостатки данного решения по сравнению с универсальными дистрибутивами GNU/Linux. Оценивается опыт эксплуатации дистрибутива в течение 2007/2008 учебного года.

Государственное образовательное учреждение высшего профессионального образования «Санкт-Петербургский торгово-экономический институт» (ГОУ ВПО СПбТЭИ) проводит подготовку специалистов по следующим специальностям:

- 80 502 — «Экономика и управление на предприятии (в торговле, общественном питании)»
- 80 109 — «Бухгалтерский учёт, анализ и аудит»
- 80 104 — «Товароведение и экспертиза товаров»
- 260 501 — «Технология продуктов общественного питания».

Свободное ПО (в том числе различные дистрибутивы GNU/Linux) применяется в учебном процессе с 2003/2004 учебного года для проведения практических занятий по дисциплинам цикла «Информатика» федеральной компоненты ГОС 2-го поколения для всех указанных специальностей (дисциплины «Информатика», «Информационные технологии в экономике» и «Информационные системы в экономике»).

Ни один из опробованных дистрибутивов GNU/Linux не обеспечивал желаемого сочетания простоты развёртывания, функциональности, поддержки оборудования и возможностей обновления (опробовано более 10 дистрибутивов). Поэтому был сформулирован набор критериев (см. <http://heap.altlinux.org/engine/Edu/Soft/AltLinuxDesktopEdu>) и проведён поиск разработчиков, готовых создать дистрибутив в соответствии с этими критериями. Одним из важнейших требований было наличие установочного комплекта на компакт-дисках (CD), а не на DVD. Результатом явился ALT Linux Desktop Edu в комплекте из 2-х CD, основными особенностями которого являются

- ограниченный (в соответствии с ТЗ) набор поддерживаемого оборудования;
- ограниченный (требуемый) набор прикладных программ;
- наличие LiveCD для тестирования аппаратных конфигураций.

Первый диск обеспечивает развёртывание системы с набором программ, необходимым для курса «Информатика», а второй диск может работать как LiveCD или как источник дополнительных пакетов. Такое решение существенно ускоряет процесс подготовки компьютерных классов по сравнению с многодисковыми дистрибутивами, когда для получения нужной функциональности рабочего места задействуются почти все CD дистрибутива (современные однодисковые дистрибутивы вообще не обеспечивают необходимой функциональности).

Недостатком данного решения является ограниченный набор поддерживаемого оборудования и прикладных программ, но при использовании репозитория branch после установки базовой системы при

необходимости можно расширять функциональность конкретного рабочего места.

Нужно отметить, что передача данного дистрибутива и репозитория branch документально оформлена, что существенно снижает риски претензий со стороны контролирующих органов.

В процессе эксплуатации дистрибутива на рабочие места устанавливались различные варианты дистрибутива (условно «августовская», «октябрьская» и «декабрьская» сборки). Они появлялись вследствие обновления репозитория branch и при идентичности в целом демонстрировали мелкие отличия, связанные с настройками системы «по умолчанию». Почему-то особенно сильно в разных сборках отличались настройки для среды разработки FreePascal.

Возникали также некоторые трудности при подборе видеорежимов для сочетаний «новая видеокарта» (i865) + «старый монитор» (LG StudioWorks 520Si), однако эти трудности оказались достаточно легко преодолимы.

В целом, опыт эксплуатации дистрибутива ALT Linux Desktop Edu можно оценить как положительный. Он используется во всех учебных компьютерных классах на 120 рабочих местах, а также в подразделениях института на компьютерах, не имеющих DVD-привода (на компьютерах с DVD-приводом используется ALT Linux Desktop). Использование дистрибутива будет продолжено и в следующем учебном году.

А. Н. Гороховский

Донецк,

Донецкий национальный технический университет

Проект: «Assistant» — использование СПО в образовании

<http://peooc.donntu.edu.ua/predmets/>

Internet проект «Assistant» — самопроверка расчётно-технических работ студентов

Аннотация

Рассматриваются основные возможности проекта «Assistant», который используется в учебном процессе для дистанционной самопроверки студентами своих расчётно-технических работ — контрольных, курсовых и пр.

Разработка выполнена и применяется в течение 2002–2008 гг. на базе ОС ALT Linux с использованием пакетов свободно-распространяемых программ: Web-сервера *Apache*, языка *Perl* и СУБД *MySQL*.

Общемировая тенденция развития системы образования характеризуется всё возрастающей ролью дистанционного обучения и активизации этого процесса в полной мере способствует развитие web-технологий и Internet в целом [1, 2, 3].

Для тех, кто работает в системе высшего образования, не является секретом то, что во время сессий, как правило, значительно возрастает учебная нагрузка как на студента, так и на преподавателя, связанная с выполнением, проверкой и сдачей курсовых, контрольных и других работ, сопровождающих учебный процесс, особенно, если эти работы носят расчётно-технический характер. В условиях ограниченных временных рамок снижается уровень контроля расчётной части работ и, как следствие, страдает качество их выполнения.

С применением современных Интернет-технологий с точки зрения эффективности проведения учебного процесса появляется возможность некоторой оптимизации пространственно-временных взаимоотношений между преподавателями и студентами.

Эти соображения послужили основой для разработки и включения в учебный процесс специализированной Web-системы для самопроверки расчётно-технических работ студентов.

Концептуальная схема работы на стороне студента

Предполагается, что вся работа происходит на компьютере, подключённом к сети Internet (или локальной сети университета), который находится под управлением надёжной ОС (например, ALT Linux), а доступ к серверу происходит посредством браузера с графическим интерфейсом. Итак, студент:

1. Через html-форму формирует запрос¹ на проверку своей работы по требуемым: *дисциплине, виду работы* (контрольная, курсовая и т. д.), *группе, номеру варианта и зачётки*²;

¹ Адрес в нашем случае: <http://peooc.donntu.edu.ua/predmets/>

² Требование указывать в запросе номер варианта и зачётки вызвано желанием ограничить преднамеренное вмешательство в самопроверку посторонних лиц.

2. В ответ получает html-форму с полями для заполнения их промежуточными (*ключевыми*) результатами расчётов³ и окончательными ответами;
3. Заполняет поля и отправляет html-форму на самопроверку;
4. В ответ получает результаты самопроверки, анализирует их и исправляет ошибки в расчётах. В случае необходимости повторяет самопроверку⁴.

Схема работы Perl-модуля самопроверки на сервере

1. Получение запроса на формирование html-формы с полями для указанной *дисциплины, вида работы, группы, номера варианта и зачётки*.
2. Выбор из БД группы и проверка номера варианта, зачётки и т. д. Проверка даты последнего доступа. Чтение результатов предыдущих проверок и БД с «эталонными» ответами.
3. В случае успешного выполнения процедур предыдущего этапа, формирование и отправка html-формы с ранее зачтёнными ответами (из БД «эталонных» ответов) и полями для заполнения их недостающими результатами.
4. Получение html-формы с ответами и их проверка с помощью БД «эталонных» ответов:
 - разделение ответов на числовые (D) и выбираемые из списков (S);
 - фильтрация входящих данных на допустимые значения и возвращение html-формы на дозаполнение в случае пустых (D) или невыбранных (S) полей;
 - сравнение S-ответов с «эталонными» на *полное* совпадение;
 - сравнение D-ответов с «эталонными» на совпадение в «коридоре» *относительной ошибки* $\pm\Delta$ (например, $\Delta \approx 1,5\%$);
 - в случае несовпадения (как S- так и D-ответов) прекращение проверки только тех последующих ответов, которые зависят от них по цепочке расчётов.

³Впоследствии, при анализе проявившихся ошибок, *ключевые* результаты расчётов помогут выявить их источник.

⁴Для предотвращения «слепого» или автоматического подбора ответов, количество самопроверок ограничено двумя на день.

5. Сохранение журнала результатов проверки работы в БД с целью последующего анализа и на случай разрешения неясных и спорных вопросов во время консультации с преподавателем.
6. Возвращение результатов проверки:
 - в случае обнаруженных ошибок в расчётах — информирование о местах и причине их появления, с указанием возможного пути их исправления;
 - в случае успешного выполнения всех расчётов — отображение отчёта самопроверки с указанием уникального номера выполненной работы.

Литература

- [1] Научно-практический журнал «Открытое образование» <http://www.lib.ru/CTOTOR/BRUKS/mithsoftware.txt>
- [2] Журнал «Вопросы Интернет образования» http://center.fio.ru/vio/vio_07/cd_site/Articles/archive.htm
- [3] Журнал «Компьютерные инструменты в образовании» <http://www.ipo.spb.ru/journal/>

С. В. Косаченко Томская область, с.Каргасок,
МОУ Каргасокская средняя общеобразовательная школа №2

Опыт использования СПО в школьном образовании.

Аннотация

Опыт перехода сельской школы на СПО и использование СПО в дистанционном обучении школьников.

Необходимость перехода на СПО

Громкое «Пермское дело» над учителем информатики из села Сепыч Пермского края заставило пересмотреть «беспечное отношение» к программному обеспечению (далее ПО) внутри многих российских

Наименование	Цена на 1 ПК	На скольких ПК требуется	Сумма
Windows XP Prof Russian + MS Office XP 2003 Pro	4800.48	14	67 206,72
<i>Антивирусное ПО (программа + подписка на 1 год)</i>			
Nod32	1060 + 1591	45	119 295
Антивирус Касперского 6.0	961 + 673	45	73 530
<i>Графические пакеты</i>			
ADOBE CREATIVE SUITE STANDARD RU UPSELL from Photoshop	20 099,14	22	442 181.08
ИТОГО:582 917,80 рублей			

организаций, в том числе и в образовательных учреждениях. Потребовалось провести инвентаризацию имеющихся лицензий проприетарного ПО, пересмотреть необходимость использования тех или иных программных средств для решения повседневных задач, подсчитать необходимые суммы денежных средств для легализации, начать поиск альтернативного ПО.

Инвентаризация ПО в нашей школе показала, что некоторые компьютеры, поставленные нам по областным программам, из районного бюджета, а также приобретённые школой самостоятельно, поставлялись без лицензий на ПО[2].

Подсчитали суммы на установленное и часто используемое ПО, требующее лицензирования, с учётом скидок, предоставляемых на образовательное учреждение.

Данная сумма заставляет задуматься не только об источниках финансирования на приобретение в школу ПО, но и том, а так ли необходимо проприетарное ПО в школе, когда есть возможность обучать приёмам обработки информации на бесплатных аналогах указанных в таблице программ?

Как производители проприетарного ПО хоть как-то заинтересовали учителя склониться в своем выборе к их продукту? Microsoft по Академической лицензии разрешает учителю устанавливать на своём домашнем ПК операционную систему — это, конечно, широкий и щедрый жест, но для меня как заместителя директора школы, курирующего вопросы внедрения ИКТ в образовательный процесс, этого мало, т. к., например, бесплатный Linux со всеми прилагаемыми

программными пакетами я могу установить не только себе на свой рабочий и домашний ПК, но вполне законно бесплатно раздать всем своим ученикам и коллегам, и друзьям для установки на их домашние ПК!

По итогам полученных данных, обсуждений, консультаций со специалистами разного уровня, мы пришли к решению о переходе на бесплатное СПО.

При переходе особое внимание было уделено подбору такого СПО, которое позволило бы продолжать обучение школьников на курсах ИДО ТГУ.

Форматы, используемые при ДО

Рассмотрим основные форматы, используемые при дистанционном обучении ИДО ТГУ [1]. В основном, это гипертекст в разметке HTML со встроенными скриптами на JavaScript, документы в формате RTF и DOC, XLS, PPT, видео в формате RealNetworks, форматы архивных файлов ZIP и RAR.

Подбор СПО для поддержки ДО с его апробацией занял не более двух недель.

В качестве операционной системы была выбрана система Linux с графической средой KDE как наиболее подходящей для миграции с Windows. При выборе дистрибутива Linux особое внимание отводилось:

- бесплатности и лицензионной чистоте;
- руссифицированности;
- укомплектованного программного обеспечения;
- простоте миграции пользователей с Windows;
- удобству работы, т. н. «юзабилити»;
- поддержке форматов, используемых в ДО;
- поддержке пользователей посредством Интернет.

Было рассмотрено и опробовано несколько дистрибутивов, и окончательный выбор пал на Linux Mandriva FREE 2007 и Linux Kubuntu 7.04.

Linux Mandriva FREE 2007 отличается хорошей настроенностью для конечного пользователя «установил из коробки и работай», что мы очень оценили в самом начале миграции с Windows на Linux. Все драйверы и форматы (кроме архивов RAR — потребовалась установка

дополнительной утилиты), используемые в системе ДО, этим дистрибутивом поддерживаются прямо «из коробки», чем не может похвастать Windows.

Linux Kubuntu 7.04 отличается отличной и доброжелательной поддержкой пользователей через Интернет в форме форумов сетевого сообщества и большого репозитория программных пакетов (более 14 Гигабайт), установку и обновление которых можно легко производить прямо из Интернет. Однако данный дистрибутив после инсталляции пришлось «донастраивать» для использования в учебном процессе и устанавливать необходимое ПО.

Использование СПО

В настоящее время в нашей школе используется исключительно лицензионное программное обеспечение. Как мы этого добились?

Часть компьютеров, поступивших в школу по президентским поставкам, были укомплектованы лицензионным программным обеспечением. Однако часть компьютерной техники школа закупила на собственные средства. Чтобы не нарушать действующее законодательство и установить на новые компьютеры лицензионно чистые продукты, мы не стали приобретать дорогостоящее проприетарное программное обеспечение — такое, как, например, Windows или Microsoft Office, а установили свободное программное обеспечение, бесплатно распространяемое по лицензии GNU/GPL — такое, как Linux Mandriva 2007 (на 17 ПК), Linux Kubuntu 7.10 (на 12 ПК) и OpenOffice (на 40 ПК).

OpenOffice поддерживает все форматы офисного пакета MS Office, поэтому проблем при работе с документами почти нет.

Для просмотра видеолекций мы применяем RealPlayer 9 для Linux.

Для изучения программирования используем FreePascal — аналог TurboPascal.

Для изучения графики применяем мощный графический редактор Gimp.

Для решения дополнительных прикладных задач подбираем и используем СПО и FreeWare.

Интерактивная доска SmartBoard также работает под управлением Linux — фирменное ПО прилагается в комплекте с ней на CD и мало чем отличается от windows-версии.

Школьники просто перешли на СПО и применяют его во время школьных занятий и на курсах ДО ИДО ТГУ.

Литература

- [1] С.В. Косаченко С.П. Анисимова. Дистанционное обучение в сельской школе на основе информационных технологий: анализ опыта. С. 397–401, 2005.
- [2] С.В. Косаченко. Миграция с windows на linux. *Альманах «Вопросы информатизации образования»*, 2007.