Системный администратор

№4(77) апрель 2009
подписной индекс 20780
www.samag.ru

ISSN 1813-5579

Выбираем антивирус для небольшой сети

Windows 7: что новенького?

Основные изменения в WAIK для Windows Server 2008 R2/7

Синхронизируем данные между компьютерами

PowerShell: определяем имя текущего домена

Оборудование Cisco для «самых маленьких»

Тестируем устройство Dr.Web Office Shield

JavaFX – Reach Internet Application
Прощай, унылый Swing?

Приёмы минификации в веб-приложениях

РЕПОРТАЖ

2 Виртуализация от Citrix – новый взгляд на привычные вещи
Итоги Citrix Virtualization Conference 09.
Алексей Бережной

8 В Москве прошла конференция системных администраторов RootConf 2009
Интересные подробности для тех, кто пропустил мероприятие.
Дмитрий Шурупов

11 ТЕНДЕНЦИИ

АДМИНИСТРИРОВАНИЕ

12 Windows 7: что новенького?
Еще не отшумели страсти по Windows Vista, а Microsoft уже выпустила бета-версию новой ОС Windows 7.
Андрей Бирюков

16 Основные изменения в WAIK для Windows Server 2008 R2/7
Одновременно с выходом бета-версий Windows 7 и Windows Server 2008 R2 были представлены и обновленные инструменты, среди которых – Windows AIK.
Сергей Яремчук

20 Автоматическая установка Adobe Creative Suite 3
Уделяем основное внимание выборочной установке пакета.
Иван Коробко

23 «10-Страйк»
Незаменимые программы для системных администраторов.
Дмитрий Степанов

24 Делегируем права на перемещение учетных записей пользователей в Active Directory. Часть 2
На примере создания надстройки нестандартного делегирования административных полномочий для Windows Server 2003 будут рассмотрены методики разработки классов COM на языках сценариев.
Вадим Андросов

31 Синхронизируем данные между компьютерами с помощью сервисов сетевого хранения
С появлением недорогих лаптопов и ультрадешевых неттопов эти «создания» начали плодиться во всех местах моего обитания – дома, в офисе, на даче, диване. И тогда начались проблемы с синхронизацией данных.
Виталий Банковский

34 PowerShell. Определяем имя текущего домена
При подключении к каталогу Active Directory первый шаг – определение имени текущего домена. Для этого используется виртуальный объект RootDSE.
Иван Коробко

36 PowerShell. Поиск объектов в каталоге Active Directory
Получение информации из каталога Active Directory в девяти случаях из десяти сводится к поиску объектов, которые удовлетворяют заданным критериям.
Иван Коробко

40 Оборудование Cisco для «самых маленьких»
Как подключиться? Как настраивать? Что означают те или иные команды?
Сергей Крутилин

46 Почтовый клиент Alpine
Краткий обзор.
Игорь Штомпель

БЕЗОПАСНОСТЬ

48 Выбираем антивирус для небольшой сети
На чем остановить свой выбор?
Сергей Яремчук

59 Тотальная защита локальных сетей
Тестируем устройство Dr.Web Office Shield.
Вячеслав Медведев

АДМИНИСТРИРОВАНИЕ «1С»

60 Обновление конфигурации
Как избавить себя от лишних проблем.
Андрей Луконькин

ЧЕЛОВЕК НОМЕРА

62 Жизнь в стиле хокку
Интервью с Максимом Акимовым, главой представительства Kerio Technologies в России и странах СНГ.
Оксана Родионова

ВЕБ-ПРОГРАММИРОВАНИЕ

67 JavaFX – Reach Internet Application от Sun Прощай, унылый Swing?
В декабре 2008 года компания Sun Microsystems представила финальную версию JavaFX – свою платформу для создания Rich Internet Application, ставшую достойным ответом конкурентам.
Кирилл Сухов

74 Приёмы минификации в веб-приложениях
Любой веб-мастер желает, чтобы его сайт загружался быстрее, чем сайт конкурента, поэтому оптимизация - важный аспект разработки любого веб-проекта.
Антон Гришан

РЕТРОСПЕКТИВА

80 Мал, да удал: мини-комьютер PDP-8
Этот компьютер интересен оригинальными решениями, воплощенными при его проектировании, которые могут оказаться поучительными и полезными для современных разработчиков.
Алексей Вторников

ТВОРЧЕСТВО АДМИНА

88 Контейнер
Рассказ.
Станислав Шпак

91 СИСАДМИН ТОЖЕ ЧЕЛОВЕК

92 КНИЖНАЯ ПОЛКА

39, 45, 90 BUGTRAQ

Виртуализация от Citrix – новый взгляд на привычные вещи

Всевозможные конференции с презентациями каких-либо продуктов от разных компаний проходят постоянно. Большинство посетителей привыкли к устоявшемуся формату подобных мероприятий: «приветственный кофе», затем вступительное слово, презентации, кофе-брейк, снова презентации, потом обед, еще презентации, второй кофе-брейк, презентации и наконец прощальное слово, иногда фуршет и розыгрыш призов. Обычно все проходит просто, предсказуемо и даже обыденно. Но только не в этот раз...

Особенности проведения

19 марта 2009 года в гостинице «Рэдиссон САС Славянская» состоялась конференция, посвященная технологиям виртуализации – Citrix Virtualization Conference 09.

Как известно, при проведении любых публичных мероприятий наиболее ценный ресурс – это время. Обычно не успели еще обсудить все интересующие вопросы, а уже расходится пора... В этот раз компания Citrix отошла от вышеупомянутых традиционных форм проведения конференции.

Во-первых, после вступительной пленарной сессии основная программа мероприятия была разделена на несколько направлений (треков): «Технический трек», касающийся технических деталей, «Бизнес-трек», выделивший в себе часть, напрямую касающуюся бизнеса, и «Инновационный трек», посвященный новым направлениям и технологиям. Надо заметить, что данное разделение было весьма условным.

Например, Александр Светлаков делал доклад на тему «Интеграция продуктов Citrix на платформу HP Blade System» в рамках «Бизнес-трека», а в рамках технического трека выступили представители от Microsoft Алексей Мурзов и Алексей Кибкало с докладом на тему: «Microsoft и Citrix: Конкуренция или сотрудничество». Но это частности. Главное, что участники получили возможность выбора просмотреть и прослушать именно тот материал, который наиболее интересен.

Во-вторых, параллельно с пленарной сессией проводились лабораторные работы по продуктам Citrix XenDesktop и Citrix EdgeSight для XenApp 5. И те участники, которых больше интересовала практическая часть, получили возможность пощупать новые технологии своими руками.

Ну и в-третьих, во время конференции можно было сдать экзамены на сертификат Citrix Certified Administrator (CCA). Данная акция была организована «Сетевой Академей Ланит». Предварительно зарегистрироваться на экзамен можно было, отправив соответствующую заполненную форму администратору тестового центра.

В целом подобное разделение не только внесло новшества в стандартный подход к проведению конференций, но еще и позволило участникам наиболее полно использовать время и ресурсы конференции, выбрав только то, что действительно необходимо.

Помимо основной программы конференции параллельно работала выставка спонсорских компаний, так или иначе связанных с основной темой конференции.

Доклады

С самого начала конференции был задан очень высокий темп. Я расскажу о самых ярких понравившихся выступлениях.

Первое выступление и собственно открытие выставки провел Карл-Хайнц Варум (Karl-Heinz Warum, Regional vice-president DACh & CE Citrix System). Выступление было предельно кратким и довольно энергичным.

По характеру и стилю оформления пленарная сессия походила скорее на шоу, нежели на набор презентаций. Участники выходили на сцену под громкую музыку. Практически все выступления были интересны, но что касается иностранных коллег, то их речь была прямо-таки до предела насыщена эмоциями.

Докладчик кратко напомнил присутствующим об экономическом кризисе и необходимости снизить затраты и одновременно модернизировать ИТ-структуру, используя продукцию компании. Честно говоря, на конференции, наверное, не было ни одного выступления, в котором хотя бы вскользь не упомянули о нынешнем экономическом кризисе. Судя по все-

му, обсуждение мирового экономического кризиса стало модной темой, используемой при презентациях абсолютно любой продукции.

Наверное, стоит упомянуть несколько основных тезисов данного доклада. По словам Карла, в 2012 году в мире будет насчитываться порядка 650 миллионов виртуальных десктопов. И компания Citrix делает все возможное, чтобы эта перспектива получила свое воплощение.

Одним из основных преимуществ виртуализации является независимость от используемой платформы. Так, если у пользователя дома компьютер Apple Macintosh, а в компании в качестве корпоративного стандарта принята платформа Microsoft, то не существует особых проблем транслировать рабочий стол с его виртуального PC прямо на экран домашнего Макинтоша.

Докладчик также упомянул о технологии Citrix Cloud Computing, позволяющей проводить распределенные вычисления. По его словам, эта новая технология позволит осуществить качественный прорыв в области применения информационных технологий и осуществить переход на новый уровень, в том числе и в применении технологии виртуализации. Несмотря на то что доклад касался общих вопросов и носил скорее вступительный характер, речь докладчика произвела завораживающее впечатление.

Следом подошла очередь выступления Сергея Кузнецова (Country manager Citrix) и далее – Сергея Халяпина, (SE Team Lead).

Сергей Кузнецов представил доклад на тему «Стратегия виртуализации Citrix 2009: Новые решения в новых условиях». Он с энтузиазмом отметил выход новых продуктов, таких как:

- XenServer 5.0;
- XenApp 5.0;
- XenDesktop 3.0;
- Repeater (Wanscaler) 5.0.

«Но новое не значит сырое», – сказал Сергей Кузнецов, отметив тот факт, что номера версий представленных продуктов больше «1.0» (единицы) и базируются на технологиях, проверенных временем и большим количеством пользователей по всему миру. Поэтому сейчас, во время экономического кризиса, когда руководство большинства компаний задается вопросом: «А что будет, если новое не заработает», при покупке продуктов Citrix можно с уверенностью заявить, что такой проблемы не возникнет. Всего в мире на продуктах Citrix работает более миллиона серверов, которые используют более ста миллионов пользователей.

По мнению г-на Кузнецова, в недалеком будущем затраты на обслуживание офисных компьютеризированных рабочих мест не будут превышать затраты на закупку кофе для офиса. И все это благодаря системам виртуализации, и в частности таким продуктам, как Citrix XenDesktop.

Упомянул Сергей и о специальных антикризисных мерах компаний, в частности о переходе новой версии XenServer на бесплатную основу. С 7 апреля 2009 года данный продукт будет доступен для бесплатного скачивания на сайте компании Citrix.

В целом доклад произвел довольно приятное впечатление. Энергичный темп совместно с живым изложением и материалом на актуальную тему мало кого оставил равнодушным.

Далее было выступление Сергея Халяпина, который на примере некой компании Citron представил путь возможной экономии средств.

Основываясь на том факте, что 80% средств, выделяемых на ИТ, уходит на поддержку существующей инфраструктуры и только 20% идет на развитие, было предложено за счет технологии виртуализации и внедрения терминальных решений сократить расходы именно на поддержку, оставив инновационные затраты без изменений.

Среди возможных путей экономии средств были отмечены следующие:

- Централизация ресурсов. При внедрении системы виртуализации десктопов XenDesktop достаточно иметь считанное количество образов виртуальных машин для трансляции виртуальных машин на рабочее место пользователя.
- За счет виртуализации и терминализации возможно значительно сократить расходы на приобретение, модернизацию, обслуживание и утилизацию офисных компьютеров.
- Также возможно значительно снизить расходы на электроэнергию за счет внедрения более экономичных тонких клиентов взамен «прожорливых» ПК.
- Вместо расширения WAN-каналов предлагается использовать Repeater (Wanscaler) и Netscaler, что позволяет использовать существующие каналы с максимальной отдачей.

Таким образом, по словам докладчика, общая эконо-

Дмитрий Прокопенко, исполнительный директор компании «Виат», на фоне стенда своей компании

Участники конференции обсуждают новые технологии

на достичь отметки 25% в плане затрат на поддержание ИТ-инфраструктуры, что само по себе не может не радовать.

В общем была нарисована довольно радужная картина сокращения расходов при внедрении новых технологий и повышения качества сервисов, предоставляемых ИТ-структурой.

После этого Сергеем Халяпиным, Денисом Гундаревым и Николаем Шадриным были представлены несколько демонстраций: миграция виртуального офисного десктопа с Windows XP на Windows Vista, использование iPhone в качестве тонкого клиента.

Сам дух, стиль проведения конференции, энтузиазм участников способствовали удивительно живой атмосфере общения. Информация воспринималась довольно легко, вызывая положительные эмоции.

Не обошлось и без курьезов. При демонстрации миграции виртуального офисного десктопа с Windows XP на Windows Vista произошел разрыв сессии. Желая поддержать ход демонстрации, Сергей Халяпин спросил (далее идет почти дословное цитирование):

– Денис, что у нас сейчас происходит на экране?

– А это, видимо, кто-то в другом месте подключился к серверу с этой учетной записью..., – не растерявшись, ответил Денис.

Позже, когда закончилась к демонстрации использования iPhone, Денис сказал:

– Кстати, я знаю, почему произошел разрыв сессии. Это у менеджера был настроен клиент на iPhone, соответственно, он нажимает на иконку и происходит установка соединения.

– Сможем ли мы теперь показать нашу демонстрацию? – вновь спросил Сергей Халяпин.

– Сможем, если наш менеджер перестанет бегать по залу и нажимать разные кнопки, – как ни в чем не бывало ответил Денис, к всеобщему удовольствию зала.

Что касается демонстрации возможностей использования iPhone для работы с продуктами Citrix, это выглядело весьма привлекательно. Зрителей не могли не впечатлить такие вещи, как экранная клавиатура, увеличивающаяся в размерах, масштабирование рабочего стола.

Вообще надо отметить, что несмотря на некоторые технические накладки демонстрации, проводимые Денисом, вызвали неподдельный интерес и явились, по сути, ярким захватывающим событием в программе конференции.

Надо также отметить, что участие технических специалистов: Дениса Гундарева и Николая Шадрина придало конференции неповторимый оттенок живого человеческого общения. Вокруг них практически постоянно собирались участники, чтобы задать интересующий вопрос, получить необходимую консультацию или просто обменяться контактами. Что касается выступлений и демонстраций, проводимых этими замечательными ребятами, то они проходили легко и непринужденно, оставляя после себя прекрасное впечатление.

После непродолжительного кофебрейка было представлено несколько очень коротких выступлений, среди которых запомнился доклад директора по маркетингу BCC Company Александра Герасимова, который рассказал об особенностях использования продук-

Идут лабораторные работы

ции Citrix в качестве платформы для интеграции. В частности, он отметил некоторые особенности работы компаний в условиях экономического кризиса, в том числе:

- резкое и непредсказуемое изменение бизнес-среды;
- необходимость «ручного управления» (то есть моментальной реакции на создавшуюся ситуацию).

В данных условиях обычная репликация баз данных уже может не соответствовать молниеносно изменяющимся условиям рынка и требуется более быстродейственный подход, такой как терминальные решения.

Также хотелось отметить доклад Сергея Члека, (Product manager HP Blade System) который представил решения HP в плане блейд-серверов для виртуализации на базе продуктов Citrix. Несмотря на краткость (около 10 минут), выступление Сергея Члека было весьма информативным. В частности докладчик отметил интеграцию системы виртуализации с ILO (Integrated Lights-Out) и тот факт, что для управления достаточно обычного интернет-браузера. В конце доклада Сергей пригласил посетить стенд компании Hewlett Packard для более близкого знакомства с решениями на базе HP Blade System.

Компания «МТС» в лице Владимира Шмелева представила краткий доклад о возможностях защищенного удаленного доступа к ресурсам компании через каналы пакетной передачи данных (GPRS/EDGE/3G). Суть выступления заключалась в современных возможностях, которыми обладают сотовые интеграторы при организации беспроводных сетей. В качестве примера был рассмотрен вариант с банкоматом, который находится вдали от коммуникаций. Также был рассмотрен случай организации «ручного управления» для топ-менеджера, находящегося вне офиса. В своем докладе Владимир посоветовал не рассматривать крупного сотового оператора как нечто монументальное, абсолютно негибкое и плохо приспосабливаемое к нуждам отдельной компании. Наоборот, возможности крупных сотовых операторов (и «МТС» в том числе) позволяют максимально полно и адаптированно помочь современному бизнесу в организации его работы.

После этого, под громкую музыку, с улыбкой на лице вышел Фабиан Кинли (Business developer Citrix), который провел подряд два длительных выступления: «XenApp 5 FP: Новые возможности проверенного решения» и «Виртуализация рабочих станций с Citrix XenDesktop». (Во втором выступлении также принимал участие Манфред Майерхофер, представитель компании WYSE.)

Это была длительная, очень эмоциональная речь на английском языке, сопровождающаяся англоязычными слайдами презентации. Чтобы понять выступающего, значительная часть участников конференции прибегала к помощи устройства синхронного перевода. Честно говоря, эти доклады оставили несколько неоднозначное впечатление. Данное выступление больше походило на упражнение в ораторском искусстве, нежели на деловую презентацию. Если бы это выступление продолжалось в течение 10 минут, оно воспринималось бы как призыв, как побуждение к действию. Но слушать плотный поток эмоциональных возгласов на иностранном языке в течение полутора часов, пусть даже временами прибегая к услугам устройства синхронного перевода, лично мне показалось несколько утомительным. Большая часть выступления была продублирована российскими коллегами, представившими очень интересные доклады, поэтому некоторые участники конференции предпочли в это время осмотреть стенды компании Citrix и ее партнеров, представленные на выставке.

Денис Гундаров, системный инженер Citrix

После завершения пленарной сессии и обеда передо мной встала проблема выбора. Как я уже упоминал, вторая часть конференции была поделена на три трека: «Бизнес», «Инновационный» и «Технический», которые проходили одновременно. Поскольку одновременно присутствовать в трех различных местах мне показалось весьма затруднительным, я выбрал «Инновационный» трек. И не пожалел. Программа этого цикла оказалась очень интересной и познавательной.

Во-первых, был представлен крайне интересный доклад Николая Шадрина на тему Cloud Computing – новый подход к высоко нагруженным системам. В частности, особенный интерес вызвал Citrix Netscaler. Как известно, WAN-каналы в России обладают, мягко говоря, весьма скромными возможностями. При этом стоят услуги интернет-провайдеров весьма недешево. И появление нового решения в виде устройства, позволяющего не только обеспечить более качественную связь с регионами, но и оптимизировать трафик, снизив затраты вызвали значительное оживление. Во время этого доклада Николая прямо-таки забросали вопросами. Сам по себе доклад был весьма живым и впечатляющим.

Во-вторых, представителем компании ВСС Борисом Королевым были представлены крупицы бесценного опыта в докладе «Опыт построения защищенной инфраструктуры доступа для ОАО «Сибирская Угольная Энергетическая компания». Речь шла о практике использования Citrix Access Gateway.

В-третьих, Дмитрий Бессонов («Группа компаний «Сквирел») представил доклад «Создание виртуальных рабочих мест в компании «Сквирел». Как в предыдущем случае, на опыте конкретной компании было продемонстрировано, как современные технологии могут помочь в сокращении расходов на ИТ-структуру. Особенно впечатлило исследование на тему экономии электроэнергии. По словам докладчика, одна из главных проблем работы современного офиса – отсутствие возможности подвести большую электрическую мощность. Терминализация способна решить эту проблему. Энергопотребление среднего офисного системного блока ~150 Ватт. Энергопотребление тонкого клиента WYSE даже в случае самой большой нагрузки (например, запуск приложений мультимедиа) не превышает 15 Ватт. При наличии сети примерно в 100 компьютеров набегают приличные цифры. Даже с учетом повышения потребления мощности при вводе в эксплуатацию дополнительных серверов получается минимум двукратная экономия. Добавим к этому снижение затрат на ремонт, обслуживание, утилизацию и т. д., и получается весьма внушительная экономия.

В-четвертых, Анатолий Бочков, представитель компании «ОЛЛИ», представил Citrix Branch Repeater. Его доклад был замечателен тем, что помимо презентации нового продукта содержал интересные детали об особенностях передачи данных по TCP-протоколу и методах оптимизации WAN-каналов, которые используются при передаче данных. В частности были рассмотрены такие процессы, как TCP Slow Start и TCP Congestion Control. Также было полезно узнать о различных методах компрессии трафика. Особенно впечатлило сообщение о возможности сжимать зашифрованный трафик по ICA-протоколу, появившейся в новой версии Citrix Branch Repeater (WANscaler).

Несмотря на очень высокий темп проведения конференции, все же ощущался дефицит времени. Поэтому две следующие презентации были очень краткими.

Владимир Высоцкий, представитель компании Softkey, и Олег Зыков, представитель компании «Аскон», подготовили совместное выступление на тему «CAD-online от идеи до рабочего стола», в котором рассказали о возможностях публикации CAD-системы. Участники в реалии увидели опубликованное посредством Citrix XenApp-приложение «Компас 3D». В окне обычного браузера, в данном случае Internet Explorer, были продемонстрированы возможности использования CAD-приложения. Продемонстрированная схема работает на одном физическом сервере, на котором развернуты 4 виртуальных машины Microsoft Windows Server 2003. В качестве терминальных серверов использовались серверы XenApp на базе виртуальных машин, доступ осуществлялся через Citrix Access Gateway. Дополнительно был использован сервер Active Directory. Вся система виртуальных машин работает под управлением Citrix XenServer.

«XenServer – платформа облачных вычислений» – последнее выступление, которое провел Николай Романовский, представитель компании «ОЛЛИ». Он рассказал о возможностях продукта Citrix Cloud Center C3 и преимуществах использования Citrix XenServer для виртуализации серверов. Среди указанных преимуществ:

- Производительность на уровне аппаратной платформы, использует технологии аппаратной виртуализации.
- XenServer Tools для улучшения I/O.
- XenServer изначально 64-битный.
- Возможность при помощи XenAPI создать настраиваемые решения.

В целом доклад произвел очень положительное впечатление, живая речь сопровождалась хорошо подготовленной презентацией. К сожалению, из-за слишком малого количества оставшегося времени доклад получился очень сжатым. Сразу после доклада к Николаю потянулись участники конференции с просьбой записать или выслать презентацию, а также задать интересующие вопросы.

Заключение

Конференция оказалась весьма замечательным мероприятием, не только интересным и познавательным, но и крайне полезным. Информация, продемонстрированная на презентациях, возможность пощупать новые технологии во время лабораторных работ позволили не только расширить свой кругозор, но и получить ответы на интересующие вопросы. Несколько участников конференции из числа моих знакомых сообщили мне, что собираются если не полностью перевести свою систему виртуализации на продукты Citrix, то уж точно намерены скачать и протестировать предложенное ПО для сравнения с продуктами VMware. В конце концов, всегда полезно иметь альтернативу, а продукты компании Citrix и ее партнеров способны сделать эту альтернативу поистине блестящим решением.

Алексей Бережной,
фото автора

Сергей Чиж рядом со своим «подопечными» – полкой с блейд-серверами

Mandriva Linux

Сертифицированная ФСТЭК версия

Дружественный и удобный интерфейс, Простота работы и настройки, Большой спектр поддерживаемого оборудования, Гарантия безопасности: дистрибутивы сертифицированы ФСТЭК.*

Офисная рабочая станция

Mandriva Powerpack 2008 Spring — надежное решение для рабочей станции. Включает в себя офисный пакет OpenOffice.org: текстовый редактор, электронные таблицы, редактор презентаций, конструктор баз данных, почтовый клиент, браузер, другие интернет-приложения, графические редакторы, приложения для работы со звуком и видео, другое ПО для офисного компьютера.

Мобильное рабочее место

Mandriva Flash — защищенное рабочее место для мобильных сотрудников. Mandriva Flash загружается и работает прямо с USB-накопителя. Mandriva Flash содержит необходимые офисные приложения и достаточно места для хранения ваших настроек и данных. Все, что нужно для загрузки защищенного рабочего места — это любой компьютер, поддерживающий загрузку с USB-носителя.

Надежный сервер

Mandriva Corporate Server 4 Update 3 — надежное решение для сервера. На базе Mandriva Corporate Server можно создать: интернет-сервер, почтовый сервер, сервер баз данных, сервер приложений, сервер печати, и т.д.

* Сертификат ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ.

Сертифицированные ФСТЭК продукты рекомендуются к использованию в государственных организациях и организациях, обрабатывающих персональные данные граждан.

Приобрести сертифицированные ФСТЭК продукты вы можете в ГНУ/Линуксцентре.

www.linuxcenter.ru | Телефон в Москве: (499)271-49-55 | Телефон в Санкт-Петербурге: 8(812) 309-06-86

В Москве прошла конференция системных администраторов RootConf 2009

13 и 14 апреля в московском конференц-центре «Инфопространство» прошла профессиональная конференция системных администраторов RootConf 2009 (www.rootconf.ru). В течение двух дней на мероприятии было прочитано около 50 докладов, проведено 2 мастер-класса и 1 «круглый стол».

В целом

Организаторам удалось собрать представителей многих известных компаний, ряд заметных деятелей российской ИТ-индустрии и приятную аудиторию заинтересованных слушателей. Доклады отбирались довольно тщательно – благо, по заявлению организаторов, на каждый слот программы RootConf в среднем претендовало по четыре выступающих. Что не менее важно, широкий спектр тем конференции позволил затронуть все актуальные проблемы и тренды, повседневно встречающиеся системным администраторам разного уровня.

Помимо нескольких залов для выступлений в просторном холле «Инфопространства» разместились стенды компаний, сотрудники которых с удовольствием рассказывали о своих технологиях и отвечали на вопросы посетителей. Если добавить сюда различные конкурсы для всех желающих и качественную организацию мероприятия, получается и вовсе идиллия. Единственным условным недостат-

ком, который, судя по отзывам в Сети, помешал прийти на конференцию многим желающим, стала стоимость участия. Впрочем, несмотря на это, были зарегистрированы почти 500 участников, так что в дни RootConf 2009 не пустовал ни один из задействованных залов.

День первый

Сразу оговорюсь, что поскольку докладов было много, читались они параллельно. Если в первый день для этого было отведено два зала, то во второй – уже три. Поскольку посетить их все одному человеку не представлялось сколь-нибудь возможным, в обзоре я затрону лишь те, на которых я смог присутствовать и которые мне показались наиболее интересными.

Итак, выступления в первом зале первого же дня конференции открывал Игорь Сысоев из компании «Рамблер», всем известный как автор легковесного веб-сервера nginx. Его доклад был посвящен настройке операционной системы FreeBSD 7 при ее использовании для обслуживания нагруженного веб-сервера. Игорь уже делал выступление на схожую тему два года назад на РИТ-2007 (см. «Конференция РИТ-2007 собрала веб-разработчиков России» в журнале «Системный администратор» №5 за 2007 г.), но теперь, с появлением новой стабильной ветки FreeBSD 7, потребовались определенные адаптации прошлых рекомендаций. Как объяснил автор, к подобным выступлениям его побудил тот факт, что если в Сети и появляются статьи на тему оптимизации FreeBSD, то обычно они сводятся к перечислению различных sysctl без особых комментариев. Более того, зачастую приводимые sysctl противоречат друг другу, а невнимательные читатели могут этого даже не заметить. Посему Игорь постарался в своем докладе не только приводить какие-то конкретные рекомендации, но и объяснять их смысл. В конце презентации автор пообещал оформить советы в виде статьи, которую в дальнейшем опубликует на сайте (http://sysoev.ru).

Сергей Присяжный из ATMNIS рассказал о сетевой подсистеме OpenBSD, затронув такие темы, как сокеты, mbuf, сетевой стек и маршрутизация. К сожалению, докладчик начал слишком детально рассказывать об общих вещах с демонстрацией на схемах, как IP-пакеты проходят в стеке от интерфейсного уровня до транспортного и обратно, из-за чего до сетевых реализаций в OpenBSD дело так и не дошло.

Андрей Пантюхин из проекта FreeBSD.org поделился своими взглядами на современное состояние управления программным обеспечением в UNIX-подобных системах, впрочем, акцент на UNIX во многом был формален (и просто более близок автору), поскольку ряд тезисов можно в значительной мере обобщить на другие платформы. Автор говорил как о том, что уже есть, развивается и станет популярным в ближайшем будущем (например, p2p-репозитории пакетов), так и о вещах, которых принципиально не хватает существующим средствам. По его мнению, например, было бы неплохо интегрировать в пакетный менеджер некоторые возможности управления ресурсами (скажем, задавать системные ограничения конкретным приложениям при их установке/конфигурации), обеспечить многоверсионные установки (чтобы появлялась возможность простого тестирования новой версии ПО на рабочем сервере до ее производственного применения). Предвидя вопросы о том, почему в этом не помогут современные виртуализационные технологии, Андрей довольно категорично заявил, что виртуализация – это не решение проблем, а попытка обойти их.

Александр Ильичев из Microsoft, занимающийся продвижением Windows 7, постарался рассказать о том, почему новая операционная система софтверного гиганта стала работать быстрее. Впрочем, публике он скорее запомнился благодаря одному забавному комментарию, вызвавшему бурную реакцию всей аудитории. По его словам, когда в Microsoft приступили к работе над Windows Vista (2001 год), в компании ориентировались на существовавший в то время прогноз, что в 2006 году средний пользовательский десктоп будет оснащен 10-гигагерцовым процессором и 8 Гб оперативной памяти. Как все дружно заметили, это многое объясняет... Из нововведений в Windows 7, которые позволили снизить ее системные требования (официально они остались такими же, как у Vista, но со слов Александра, в действительности – стали ниже), можно выделить существенную работу над так называемой фоновой активностью. Если говорить вкратце, то теперь благодаря так называемым сервисам Trigger-Start все службы запускаются только тогда, когда они действительно нужны и будут использоваться, и останавливаются аналогичным образом.

Последним «большим» докладом этого дня стало выступление Дмитрия Завалишина про загадочную операционную систему «Фантом» (Phantom OS, http://dz.ru/solutions/phantom). Несмотря на во многом провокационную разработку и понятный скептицизм всех, кто про нее впервые слышит, выступление прошло на одном дыхании, а резкой критики со стороны слушателей замечено не было. Идея проекта Phantom OS – в создании принципиально новой ОС, не отягощенной многолетним наследием существующих платформ и одаренной нестандартной архитектурой. «Фантом», откуда и происходит ее название, не должна быть видна программистам, а процессы смогут обмениваться данными между собой напрямую, находясь в едином пространстве памяти и не делая лишних операций вроде обращений к записи/чтению данных на жесткий диск. Кроме того, все процессы системы будут постоянно находиться в оперативной памяти, благодаря чему, например, система сможет мгновенно загружаться после включения компьютера. Докладчик поделился проблемами, с которыми столкнулся при проектировании и создании новой ОС, а также путями их решения. Стоит заметить, что законченного продукта еще нет, но некоторые наработки у проекта уже присутствуют. На вопрос о лицензировании Дмитрий ответил, что вопрос лицензирования ядра «Фантома» пока обсуждается, а все остальное авторы обещают выкладывать под Open Source-лицензиями вроде GPL. Проблему полного отсутствия ПО для новой ОС разработчики намерены решить написанием бинарного транслятора байт-кода Java в свой байт-код и, возможно, созданием POSIX-слоя для совместимости с приложениями для UNIX-подобных систем.

Первый день завершился циклом блиц-докладов протяженностью по 5 минут. В них можно было ознакомиться, например, с Open Source-приложениями, используемыми в LiveJournal, простым и очевидным способом борьбы с вирусами с помощью средства виртуализации VirtualBox (куда установлена Windows) и снапшотов файловой системы ZFS в исполнении Филиппа Торчинского из Sun, видением рабочей среды системного администратора UNIX глазами уже знакомого посетителям Андрея Пантюхина.

День второй

Второй день RootConf 2009 оказался для меня не таким продуктивным по ряду причин, главная из которых – менее интересные темы докладов. Тем не менее некоторые доклады стоит выделить:

- Андрей Никишин из «Лаборатории Касперского» поделился своим опытом в области работы со спамом. Особое внимание автор уделил такой современной угрозе, как SMS-маркетинг через почтовый спам. Была приведена краткая классификация существующих решений борьбы со спамом: программные, программно-аппаратные, hosted service (на домене заказчика изменяется MX-запись, после чего все заботы о фильтрации спама берет на себя удаленный сервер).
- Филипп Кулин рассказал о проектировании и реализации небольшого почтового сервера, основываясь на своем опыте работы в Peterhost.
- Андрей Чеков из «АстелНет» поведал о разработанном в его компании дистрибутиве Asterroid, применяемом для организации цифровой АТС. В его основе – CentOS Linux 5.3 с Asterisk/Callweaver и коммерческий интерфейс от Thirdlane PBX. Что интересно, после внедрения непосредственно IP-АТС заказчику предлагается возможность использовать этот сервер и для других целей – с помощью доступных в CentOS стандартных серверных приложений вроде ipcad (для учета трафика) и duplicity (для создания резервных копий).
- Кирилл Колышкин из Parallels в подробностях рассказал о возможностях управления ресурсами, предлагаемыми популярным виртуализационным средством для Linux – OpenVZ.
- Константин Осипов, участвующий в разработке СУБД MySQL, представил обзор Open Source-набора утилит Maatkit, значительно упрощающих работу с репликацией MySQL-серверов.

Подводя итоги

Как уже и было отмечено в начале репортажа, на мой взгляд, конференция RootConf 2009 однозначно удалась. Это и неудивительно, когда соблюдаются все ключевые компоненты подобных мероприятий: хорошая организация, интересные докладчики и широкий охват тем.

Дмитрий Шурупов,
фото предоставлено организаторами конференции

Intel передала Linux Foundation управление проектом Moblin

2 апреля стало известно, что компания Intel передала некоммерческой организации Linux Foundation право руководить разработкой своего Linux-дистрибутива Moblin для портативных устройств.

Moblin (Mobile & Internet Linux Project) – это Open Source-инициатива Intel, запущенная в 2007 году и нацеленная на создание Linux-дистрибутива, ориентированного на использование в мобильных устройствах вроде нетбуков (разумеется, с процессором Intel Atom). До сих пор координацией проекта занимались сотрудники Intel, но теперь в компании решили, что развитие будет лучше продолжить в рамках мирового сообщества, доверив его консорциуму Linux Foundation. И судя по всему, в Linux Foundation настроены серьезно: «С технической поддержкой уважаемых разработчиков Linux-ядра и независимым, сторонним, руководством проект Moblin нацелен на то, чтобы стать самой продвинутой и открытой мобильной Linux-платформой».

Инициативу поддержали представители ряда Linux-компаний, в числе которых оказались Canonical, Linpus, Mandriva, MontaVista, Novell, Red Flag и Wind River.

Примерно через неделю на саммите Linux Collaboration Summit выступил директор центра Open Source-технологий Intel, который рассказал о планах по совершенствованию Moblin. Была озвучена довольно интересная информация: тесты последнего альфа-релиза Moblin 2 показали, что все крупные компоненты платформы (включая графическую систему) могут загружаться всего за несколько секунд. В связи с этим в Intel намерены вновь сократить общее время загрузки системы – теперь уже до двух секунд. Стремление к такой цели объясняется новыми открывающимися возможностями Moblin: например, с такой скоростью загрузки систему можно будет использовать на устройствах вроде встраиваемых автомобильных компьютеров.

Другим важным аспектом в совершенствовании Moblin станет пользовательский интерфейс. В прошлом году Intel купила компанию OpenedHand, стоящую за разработкой Open Source-фреймворка Clutter. Ожидается, что теперь с его помощью будут создаваться динамические пользовательские интерфейсы, использующие анимацию и графическое ускорение.

Microsoft готовит рынок веб-приложений, открывает код ASP.NET MVC

На конференции MIX09 стало известно, что корпорация Microsoft занялась созданием рынка веб-приложений с открытым исходным кодом, работающих под управлением операционной системы Windows. Помимо самих приложений через него будут также распространяться сопутствующие сервисы и техническая поддержка.

Новый рынок веб-приложений Microsoft был представлен Лорен Куни (Lauren Cooney), менеджером по продуктам подразделения Microsoft Web Platform and Standards. Примечательно, что раньше Лорен работала в IBM и BEA Systems над проектами, связанными с Open Source-технологиями, веб-приложениями и языком программирования Java. Теперь под девизом «Make Web not war» она возглавила направление в Microsoft, которое станет новым витком развития Microsoft Web Platform – набора серверов, фреймворков и приложений, формирующих платформу, нацеленную на взаимодействие с популярными и широко распространенными в веб-сообществе Open Source-приложениями. Веб-приложения для Microsoft Web Platform поставляются со специальным инсталлятором Web Platform Installer, задача которого – упростить процессы их установки и последующего обновления.

Кроме того, Microsoft объявила о запуске галереи Windows Web Application Gallery, которая наглядно демонстрирует далеко идущие планы корпорации: в новом решении предусмотрена интеграция с такими веб-приложениями, как Acquia Drupal, DotNetNuke и WordPress.

Вскоре после этого в блоге другого представителя Microsoft, Скотта Гатри (Scott Guthrie), появилось объявление о публикации исходного кода веб-фреймворка ASP.NET MVC 1.0 под Open Source-лицензией Microsoft Public License (MS-PL). Последняя была одобрена в конце 2007 года группой OSI как лицензия, удовлетворяющая определению Open Source Definition (OSD).

ASP.NET MVC – это новый фреймворк для языка программирования ASP.NET, во многом схожий, например, с популярным Ruby on Rails для Ruby. Он обеспечивает полный контроль за HTML-разметкой, структурой URL-адресов, упрощает модульное тестирование и способствует использованию модели разработки TDD (test driven development).

Rackable Systems поглотила компанию SGI

1 апреля на сайте SGI (Silicon Graphics Inc.) появился пресс-релиз, согласно которому компанию поглощает Rackable Systems всего за 25 миллионов USD. Некоторые могли подумать, что это шутка, но ситуация далека от таковой. SGI хорошо известна всей ИТ-индустрии своими суперкомпьютерами (в свое время она поглотила Cray), операционной системой IRIX и журналируемой файловой системой XFS, графической библиотекой OpenGL. Финансовые проблемы у компании стали заметны еще в начале 2006 года, когда SGI открыто объявила о том, что опасается банкротства и одной из видимых причин тогда назвала операционную систему GNU/Linux: снижение доходов SGI стало результатом перехода от собственных технологий к рабочим станциям на базе процессоров Intel Itanium с Linux. Уже в мае 2006 года компания и ее американские подразделения подали заявление на защиту от банкротства.

Теперь SGI стала частью Rackable Systems, а сумма сделки составила всего 25 миллионов USD – правда, это не совсем точные цифры, поскольку новый владелец наследия SGI пообещал еще и выплатить все ее долги. В пресс-релизе заявляется, что «объединенный бизнес предоставит потребителям лидирующие на рынке программно-аппаратные решения». К их основным зонам интереса относят кластерные вычисления на базе x86, Интернет, облачные вычисления, масштабируемые центры хранения данных, платформы визуализации.

Подготовил Дмитрий Шурупов по материалам www.nixp.ru

Windows 7: что новенького?

Андрей Бирюков

Еще не отшумели страсти по Windows Vista и в интернет-сообществах идут нескончаемые споры о ее преимуществах и недостатках, а корпорация Microsoft уже выпустила бета-версию новой ОС Windows 7.

Сам факт столь раннего появления следующей после Windows Vista операционной системы многие истолковали как признание Microsoft провала ОС Vista. Ведь практически сразу после официального выхода Vista появились сообщения о разработке новой версии операционной системы. Да и быстрое появление бета-версии Windows 7 подтверждает данные предположения. Однако исполнительный директор Microsoft Стив Балмер опроверг факт провала, заявив при этом, «что в Vista есть над чем работать» [1]. По заявлениям того же Балмера, официальный релиз Windows 7 может выйти уже в конце текущего года. Несмотря на это, думаю, изучать особенности и нововведения новой операционной системы можно уже сейчас.

Прежде всего, следует отметить что это тестовая (beta) версия операционной системы, в связи с чем категорически не рекомендуется ставить ее в качестве рабочей ОС на «боевые» машины. По заявлениям Microsoft, данная версия прекращает функционировать 1 августа 2009 года. На момент написания данной статьи дистрибутив Windows 7 уже не был доступен для бесплатного получения с сайта Microsoft.

Я расскажу об основных нововведениях, а потом продемонстрирую на практике работу некоторых из них.

Немного об установке

Я не буду описывать процесс установки операционной системы, так как он в целом схож с установкой Windows Vista, так же минимизировано количество задаваемых перед началом вопросов.

Однако я приведу некоторые свои наблюдения относительно самого процесса установки и нагрузки, создаваемой на систему. Данный процесс занимает намного меньше времени. Так, установка Windows Vista на VMWare Workstation с выделенными 756 Мб оперативной памяти заняла у меня около часа, при этом работать в основной операционной системе было практически невозможно, так

как более 90% аппаратных ресурсов уходило на обслуживание виртуальной машины. Для Windows 7 я установил 512 Мб оперативной памяти, и установка при этом заняла немногим более получаса. При этом основная операционная система была вполне работоспособна, по крайней мере, я смог начать писать эту статью.

Интерфейс

После завершения установки и перезагрузки (правда, только одной) мы попадаем в интерфейс, очень похожий на стандартное рабочее окно Windows Vista. При попытке настроить систему появляется уже знакомое предупреждение User Access Control, технологии, позволяющей контролировать использование пользователями административных привилегий.

В качестве интерфейса в Windows 7 применяется модернизированный вариант Vista Aero. Данный интерфейс поддерживает новую «Панель задач/Taskbar», которая сочетает кнопки запуска приложений, наиболее часто используемых, с иконками приложений, уже запущенных, так что вам не придётся переключаться между панелью задач, меню «Пуск/Start» и панелью «Быстрого запуска/Quick Launch». Также здесь можно переносить иконки в меню «Пуск/Start» из него, и выбирать желаемый порядок иконок, не зависящий от порядка, в котором запускаются приложения. Располагать окна стало легче: просто перетащите окно на верхнюю часть экрана, и оно максимально раскроется; переместите окно на край экрана, и оно закроется наполовину, чтобы облегчить процесс копирования или вставки.

Говоря об интерфейсе, следует отметить тот факт, что изначально в своих рекламных сообщениях Microsoft активно распространяла слухи об использовании на экране так называемого «кольца команд», позволяющего прокручивать логотипы команд на экране, выбирая нужную (по аналогии с iPhone), однако в текущей версии Windows 7 данный интерфейс не представлен, возможно, именно потому, что для него требуется слишком много аппаратных ресурсов.

Новые возможности визуализации, представленные еще в Windows Vista, получили свое развитие в Windows 7.

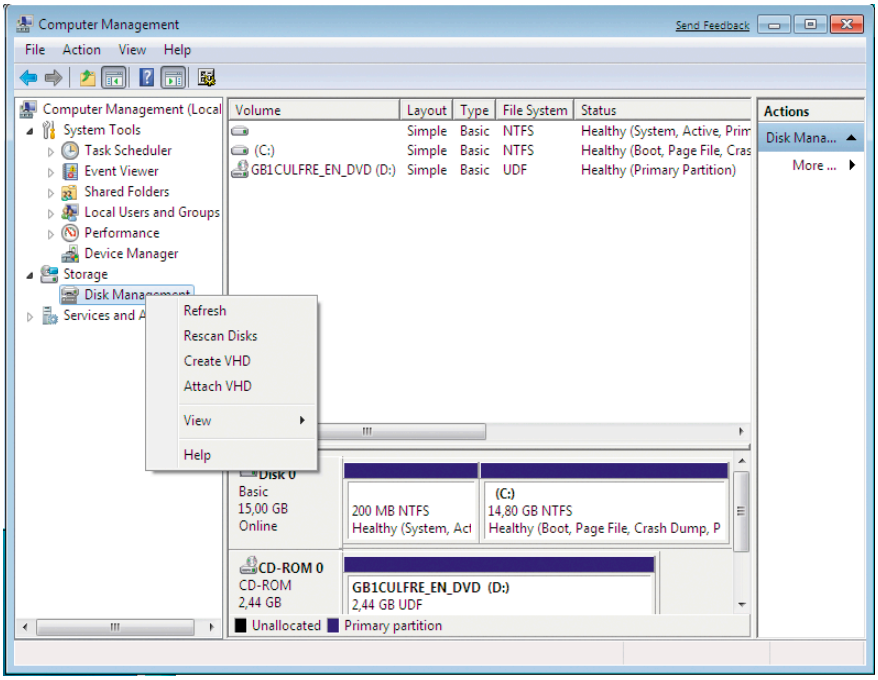


Рисунок 1. Создание VHD-диска

Например, появившийся в Vista быстрый просмотр приложений в панели задач, который выводит уменьшенный экран работающего приложения, стоит только навести на него курсор мыши. В результате такого быстрого просмотра вы получаете только приблизительное представление о том, что в данный момент делает программа. Все уменьшенные экраны будут отображаться одинакового размера.

В добавление к этому в Windows 7 приложения получили собственные места на панели задач, и теперь вы можете навести курсор мыши на уменьшенный экран, чтобы просмотреть его в полном размере.

Если окно вам больше не требуется, вы можете закрыть его из предварительного просмотра, без переключения в само приложение. Вкладки в Internet Explorer работают по аналогичному принципу – каждая вкладка выводится отдельно.

Однако помимо визуальных элементов в новой версии операционной системы также появились и технические нововведения, о которых я расскажу далее.

Работа в сети

Работа с сетями в Windows 7 также претерпела некоторые изменения. С помощью так называемых Jump lists (списков перехода) можно воспользоваться новой функцией «Просмотр доступных сетей/View Available Network». Теперь не нужно проходить через многие диалоги подключения, чтобы подсоединиться к беспроводной сети, теперь можно быстро выбрать нужную беспроводную сеть из значка беспроводных подключений в трее. Всё, что вам требуется – выбрать беспроводную сеть, которую вы хотите использовать, после чего нажать клавишу «Подключиться/Connect». Если нужно ввести сетевой ключ защиты, то будет предложен соответствующий диалог, а если нужно войти через вебстраницу, то на ней вы и окажетесь. Тот же самый список позволяет подключаться к мобильным широкополосным сетям через 3G-модем или к корпоративным VPN-сетям.

Новшества в сфере безопасности

Средства безопасности также претерпели некоторые изменения по сравнению с Windows Vista. В Vista появился целый ряд новшеств: фильтрация, как входящих, так и исходящих соединений, динамические профили, интеграция с IPSec и поддержка IPv6. Однако активным при этом мог быть только один профиль, и при нескольких подключениях применялся наиболее ограничивающий из них. В Windows 7 это упущение было исправлено – теперь несколько профилей могут быть активными, что позволяет более гиб-

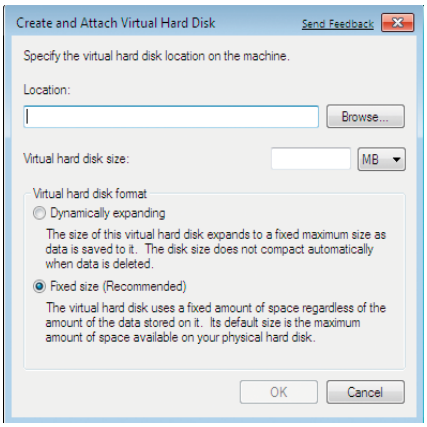


Рисунок 2. Настройки VHD-диска

ко настраивать правила для межсетевого экрана.

Также в Windows 7 теперь можно шифровать содержимое съемных носителей с помощью утилиты BitLocker. Теперь BitLocker поддерживает файловые системы FAT32, ExFAT, NTFS, а также доступ по паролю или смарт-карте.

«Легкий медиаплеер»

Операционную систему семейства Windows невозможно представить без Media Player, входившего в ее состав на протяжении долгих лет. Работа с ним под Vista и XP была примерно одинаковой: запускалось «тяжёлое» приложение, которое даже для того, чтобы проиграть несколько простых мелодий, интенсивно нагружало процессор, занимало много места на экране и в памяти, что не всегда устраивало пользователей, и они, как правило, использовали альтернативные, более «легкие» проигрыватели. К тому же был изменен интерфейс проигрывателя в лучшую сторону, добавлены функции для перезаписи списков воспроизведения.

Создание виртуальных дисков

Одним из принципиально новых средств работы с дисковыми ресурсами стала возможность создания и подключения виртуальных дисков – VHD-файлов. Причем теперь эти диски можно использовать в качестве обычных накопителей.

Использование виртуальных дисков дает целый ряд преимуществ:

- при мультизагрузке операционная система остается неизменной;
- не требуется отдельный раздел для каждого экземпляра операционной системы;
- появляется возможность протестировать совместимость оборудования и периферии с новой ОС;
- удобство резервного копирования (весь образ операционной системы сохраняется в один файл);
- различные образы vhd можно использовать при обучении и на публичных компьютерах.

Раньше VHD-файлы использовались только виртуальными машинами Virtual PC, однако теперь их можно использовать и в рабочей среде, без использования сторонних программ.

Посмотрим, как можно работать с виртуальными дисками [2]. Для начала создадим виртуальный диск. Для этого необходимо открыть консоль управления дисковыми ресурсами, которая находится, как и в предыдущих версиях Windows, в разделе Computer Management, далее Disk Management. Здесь нужно нажать правую кнопку мыши и выбрать Create VHD (см. **рис. 1**).

В открывшемся окне необходимо задать параметры создаваемого диска. Как видно, этот интерфейс аналогичен меню создания виртуальных машин в Microsoft Virual PC. Необходимо указать размещение VHD-файла, его максимальный размер, а также способ выделения дискового пространства: динамическое или фиксированное. Несмотря на то что здесь рекомендуется использовать фиксированное выделение пространства (VHD-файл сразу имеет заданный максимальный размер), я обычно использую динамическое выделение, так как это помогает сэкономить свободное место на диске (см. **рис. 2**).

Через некоторое время VHD-диск будет создан. Теперь, нажав на нем правой кнопкой мыши, мы можем выполнить над ним различные действия. (см. **рис. 3**).

Выберем Detach. В презентации [2] докладчик продемонстрировал, как можно работать с VHD-дисками с помощью консольных команд. Думаю, данный опыт будет весьма полезен, к примеру, в задачах автоматизации резервного копирования, поэтому разберем пример подробнее.

Итак, откроем командную строку, для этого в Windows 7 необходимо в разделе Accessories открыть Command Promt. Далее запускаем утилиту diskpart. Для того чтобы ознакомиться со всеми доступными командами утилиты, можно указать «?».

Теперь подключим созданный нами виртуальный диск. Для этого необходимо использовать следующую команду:

```
select vdisk file=путь_к_VHD_файлу
```

В ответ мы должны получить сообщение об успешном открытии фай-

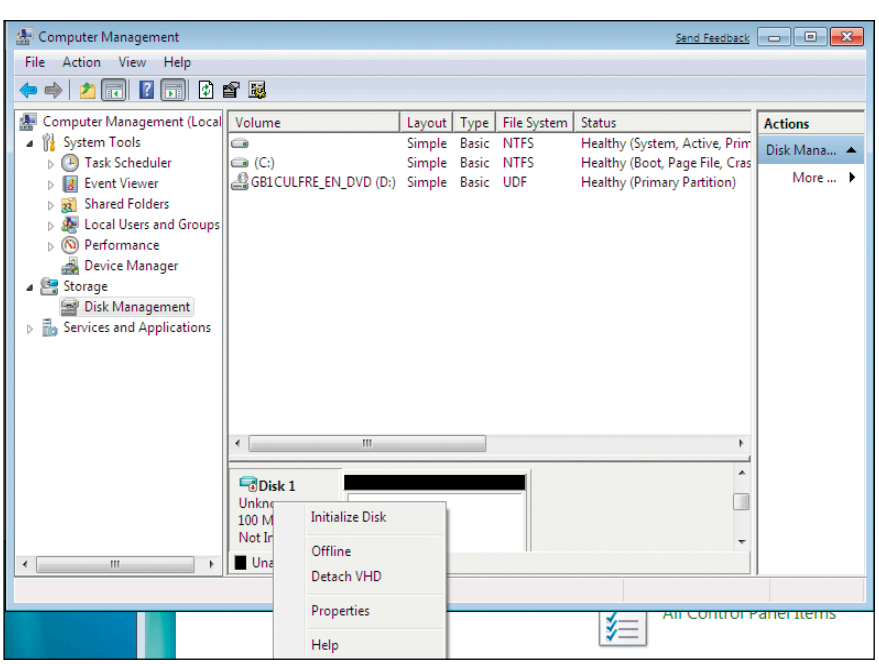


Рисунок 3. Действия над созданным VHD-диском

ла виртуального диска. Затем вводим команду:

```
attach vdisk
```

После этого следует сообщение об успешном подключении диска.

Теперь можно работать с нашим виртуальным диском, как с обычным ресурсом. Например, с помощью сценария сделать резервную копию какого-либо бизнес-приложения. Для отключения виртуального диска в diskpart необходимо ввести команду:

```
detach vdisk
```

Продолжая тему работы с дисками VHD, попробуем развернуть на виртуальном диске операционную систему. Это может оказаться полезным, когда необходимо иметь готовый образ типовой операционной системы, используемой на рабочих станциях пользователей или для тестирования нового оборудования. Виртуальный VHD-диск аналогичен Live CD, загружаемым дистрибутивам, которые не вносят изменений в установленную операционную систему.

Но, прежде чем приступить к созданию образа операционной системы, необходимо обратить внимание на следующие моменты.

Не сохраняйте файлы VHD в каталог Windows, не используйте слово Windows в названии диска. Далее имейте в виду, что для дисков VHD не поддерживается компрессия. К сожалению, также не поддерживается загрузка с VHD на съемном носителе. Также есть несколько предостережений при работе с расширяемыми VHD-дисками (вспомните настройки на **рис. 2**). Свободное место на диске необходимо резервировать заранее под максимальный размер, так как при загрузке диск будет увеличен до данного размера. Скорость работы с VHD фиксированного размера будет выше, так как он менее подвержен фрагментации. Рекомендуемый минимальный размер диска – 7024 Мб.

Некоторые из приведенных ограничений показались мне несколько странными (например, невозможность загрузки со съемного носителя). Хочется верить, что в финальной версии Microsoft исправит недостатки.

Итак, приступим к созданию загружаемого VHD-диска с операционной системой. Сначала необходимо создать VHD-диск с помощью действий, приведенных выше. Для этого нам потребуется сначала загрузиться с диска с дистрибутивом Windows 7.

В первом же окне установки необходимо запустить командную строку с помощью комбинации клавиш <Shift>+<F10>. В открывшемся окне необходимо запустить утилиту diskpart. Далее:

```
create vdisk file="путь к файлу" ↵
      type=fixed maximum=размер диска
```

Затем:

```
select vdisk file="путь к файлу"
attach vdisk
```

После этого необходимо вернуться в меню установки и продолжить ее. На этапе выбора диска для установки системы нужно выбрать созданный VHD-диск. Предупреждение о том, что система не может быть установлена на данный диск, можно проигнорировать, установка все равно пройдет успешно. После установки на этапе загрузки вы можете выбрать новую операционную систему, установленную на VHD-диске.

Более подробно о работе с VHD-дисками вы можете узнать из доклада на сайте [2].

Заключение

В целом, впечатление от новой операционной системы сложилось положительное. Windows Vista, наделавшая несколько лет назад много шума, на деле оказалась довольно громоздким решением, требовательным к аппаратным ресурсам. В Windows 7 эти недостатки смогли исправить, улучшив работоспособность и функционал.

Однако разработчикам еще есть над чем поработать, так что будем ждать выхода новых релизов операционной системы.

1. http://www.windxp.com.ru/win7/winfag.htm – Материал по Windows Vista.
2. http://www.techdays.ru/videos/1237.html – презентации по VHD в Windows 7.

Основные изменения в WAIK для Windows Server 2008 R2/7

Сергей Яремчук

Одновременно с выходом бета-версий операционных систем Windows 7 и Windows Server 2008 R2 были представлены и обновленные инструменты, среди которых – Windows AIK, с особенностями которого мы и познакомимся в статье.

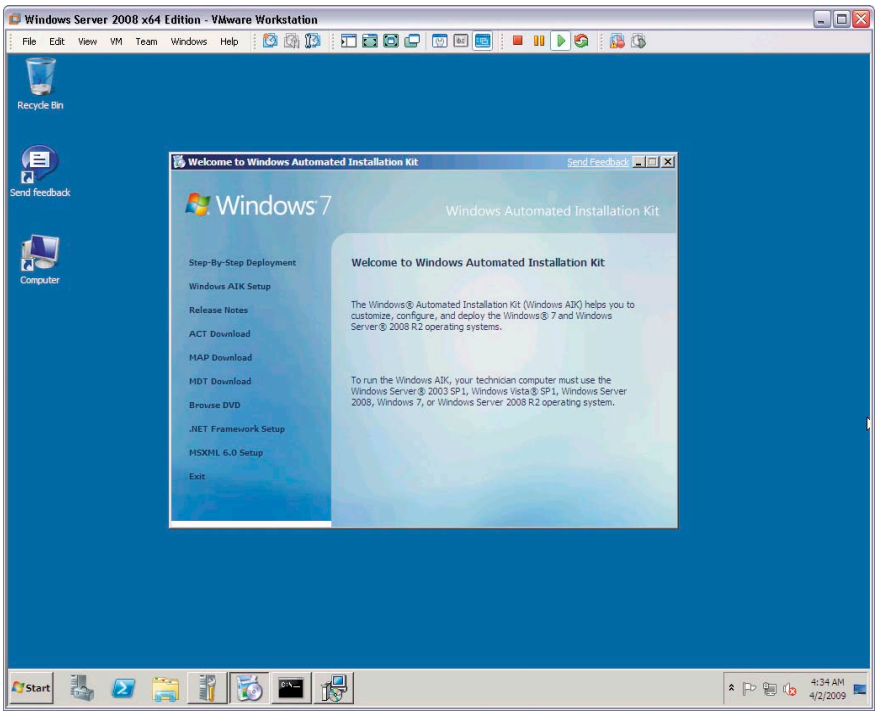
Пакет автоматической установки Windows (WAIK, Windows Automated Installation Kit) позволяет упростить развертывание операционных систем Windows за счет интеграции приложений, драйверов в установочный образ и автоматизации самой установки системы. В зависимости от ОС, которую планируется развертывать, следует подбирать и версию WAIK. Предыдущий релиз пакета назывался «Пакет автоматической установки (AIK) для Windows Vista с пакетом обновления 1 (SP1) и Windows Server 2008», на работу с которыми он собственно и был рассчитан. Уже в нем появились нововведения по сравнению с версией для XP. В частности WIM-файл (Windows Image), который поддерживает редактирование и может содержать несколько образов ОС. Плюс единый файл ответов в формате XML, избавлявший от путаницы. Выход Windows 7 потребовал и обновления соответствующих инструментов – WAIK и Microsoft Deployment Toolkit 2010 Beta (ранее пакет MDT назывался Business Desktop Deployment). Утилиты из старой версии отказались работать с новыми сервисными функциями.

Версия Windows Automated Installation Kit (Windows 7 Beta), как показывает название, ориентирована на развертывание Windows 7, а также Windows Server 2008 R2. Установить пакет можно на Windows Server 2003SP1/VistaSP1/2008/2008R2/7. Традиционно WAIK распространяется бесплатно, установочные файлы в виде DVD ISO-образа можно свободно скачать с сайта корпорации Microsoft [1].

Состав WAIK

В состав обновленного комплекта WAIK входит несколько уже традиционных инструментов:

- **WinPE** (Microsoft Windows Preinstallation Environment, среда предустановки Windows) – загрузочная версия системы, является основой при развертывании ОС, можно настроить при помощи утилит из состава WAIK;
- **набор утилит PETools** – ImageX, PEimg и Expand, при помощи которых создаются и модифицируются эталонные образы Windows, в том числе интегрируются драйверы, обновления и языковые пакеты;



Изменения в WAIK видны уже начиная с загрузочного меню

- **Windows System Image Manager** (Windows SIM, диспетчер образов системы Windows) – графический инструмент, предназначенный для создания файлов ответов, файлов каталога для управления компонентами ОС, интеграции драйверов, установки дополнительных пакетов и приложений и так далее;
- **Pkgmgr** (Package Manager, менеджер пакетов) – командная утилита, позволяющая создать новый пакет (драйвера, языковый) и включить его в образ для автоматической установки в процессе развертывания.

Плюс в комплект входит документация и дополнительные пакеты, необходимые для установки WAIK в некоторых системах.

Самым заметным изменением стало появление в составе комплекта двух новых инструментов: средства миграции пользовательской среды USMT 4.0 (User State Migration Tool) и командной утилиты DISM (Deployment Image Servicing and Management).

Удалён пакет установки службы развертывания Windows – WDS (Windows Deployment Services), предназначенный для обновления RIS. Учитывая, что WDS является одной из ролей в Windows 2008, включать его в WAIK уже нет необходимости. Теперь рассмотрим использование новинок подробнее.

Утилита DISM

Новая утилита DISM, входящая помимо WAIK и в поставку Windows 2008R2/7, при работе с образами Windows 2008R2/7 заменяет, как минимум, три утилиты из состава WAIK: Pkgmgr, PEimg и Intlcfg (International Settings Configuration Tool), плюс может использоваться в настройке рабочей системы. К слову, перечисленные утилиты никуда не делись, поэтому ничто не мешает не менять привычек и использовать старые команды. DISM позволяет добавлять, обновлять, удалять и получать список драйверов и пакетов, в том числе и языковых, включать/отключать компоненты системы, работать с образами системы и производить некоторые другие операции.

DISM также работает и с WIM-файлами Vista и Windows Server 2008 (без R2), но нужно быть готовыми, что в этом случае и при использовании WAIK в ОС, отличных от Windows 2008R2/7, часть функций будет недоступна. Правда что-то испортить не получится, так как если параметр не поддерживается, пользователь получает сообщение об ошибке. Например, в Vista и Server 2008 не работает ключ /Online, указываю-

щий, что команду необходимо выполнить в рабочей ОС, а не с образом. Например:

```
PETools> DISM /Online /Get-Packages
```

Получаем:

```
DISM does not support servicing Windows Vista or Windows Server 2008
with the /Online Options.
```

Формат самой команды и вывода результата в DISM существенно отличается от используемых ранее. Общий синтаксис DISM такой:

```
DISM.exe {/image:<path_to_image> | /online} [dism_options] ↵
      {servicing_command} [<servicing_argument>]
```

Например, чтобы получить информацию о WIM-образе, при помощи ImageX вводим:

```
PETools> imagex /INFO d:\sources\install.wim
```

В ответ получали:

```
<IMAGE INDEX="1">
<NAME>Windows Vista BUSINESS</NAME>
```

С использованием DISM команда и результат выглядят так:

```
PETools> DISM /Get-WimInfo /WimFile:d:\sources\install.wim
```

```
Index : 1
Name : Windows Vista BUSINESS
```

Если нужно обратиться к определенному образу (например, первому), следует использовать дополнительный ключ /index:1.

Ключей в DISM используется достаточно много, при помощи /? можно увидеть лишь самые востребованные, кроме этого каждый ключ также имеет дополнительные параметры. В документации WAIK расписаны все основные моменты по использованию DISM в различных ситуациях. Разберем некоторые из команд, чтобы увидеть разницу.

Образ перед использованием следует смонтировать:

```
PETools>DISM /Mount-Wim /WimFile:d:\sources\install.wim ↵
      /index:1 /MountDir:c:\img
```

Поддерживается также и ключ /ReadOnly, показывающий, что монтировать нужно только в режиме для чтения. Для получения списка различных составляющих системы следует использовать соответствующий ключ: Get-Drivers (драйвера), Get-Features (компоненты), Get-Packages (пакеты, обновления), Get-Intl (установки локализации).

```
PETools> DISM /Image:c:\img /Get-Drivers | more
PETools> DISM /Image:c:\img /Get-Features | more
```

Теперь, чтобы узнать подробнее о конкретном драйвере, вводим:

```
PETools> DISM /Get-DriverInfo /driver:oem1.inf
```

Это для установленного драйвера, иначе следует указать полный путь к файлу.

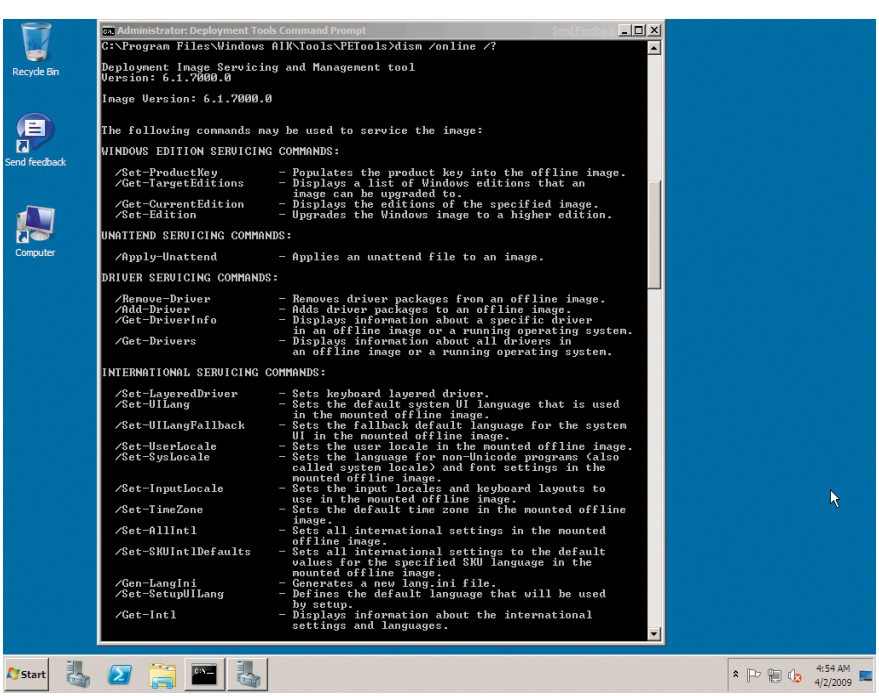
Команды можно использовать и в рабочей системе:

```
PETools>DISM /Online /Get-Features | more
```

Теперь, получив нужные названия, отключаем встроенные игры и активируем роль веб-сервера IIS:

```
PETools> DISM /Online /Disable-Feature:InboxGames
PETools> DISM /Online /Enable-Feature ↵
      /FeatureName:IIS-WebServerRole
```

Для установки или удаления пакетов используется соответственно Add-Packages и Remove-Package. Но DISM поддерживает работу только с .cab-, .msu- и .inf-файлами. Для установки .msi следует использовать OCSetup, сервис-паков – Windows Update Stand-alone Installer (Wusa.exe).



DISM имеет достаточно много различных параметров

По окончании работы образ следует размонтировать:

```
PETools> DISM /Unmount-Wim ↵
      /MountDir:c:\img /commit
```

Если сохранять изменения не планируется, то вместо /commit используем /discard. То есть общий принцип работы с образами остался неизменным. Но теперь как к образу, так и к рабочей машине можно применить файл ответов с настройками:

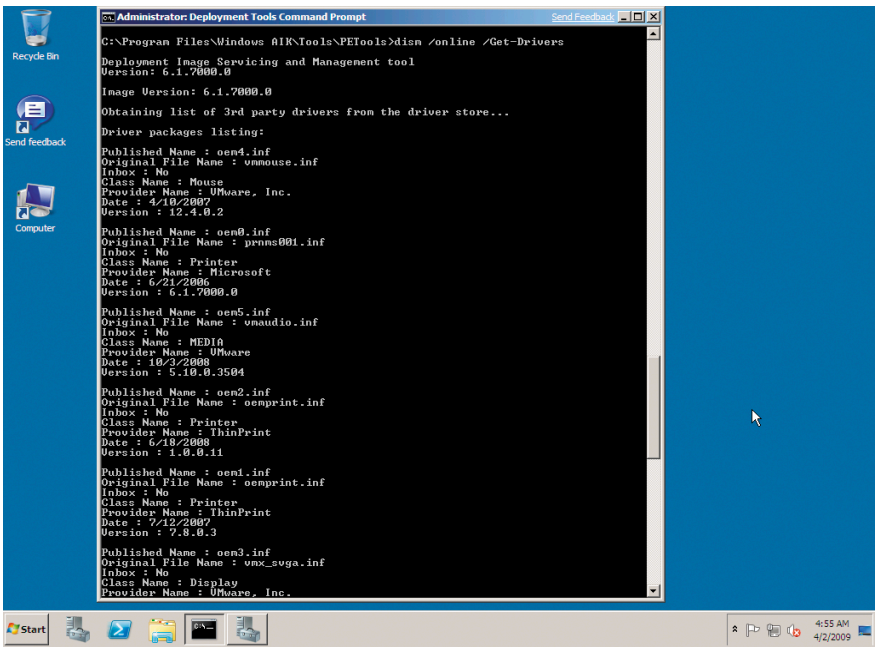
```
> DISM /Online /Apply-Unattend: ↵
      C:\answerfiles\unattend.xml
```

Параметров у DISM действительно много, причиной появления этой утилиты, скорее всего, является желание

разработчиков дать пользователю универсальный инструмент.

Возможности USMT 4.0

Набор программ USMT предназначен для настроек ОС, приложений и файлов, индивидуальных настроек пользователей во время переустановки или при масштабном развертывании ОС. Как результат ее применения весь процесс происходит быстрее и проще, а пользователь в новой системе сразу попадает в знакомую среду. USMT 4.0 поддерживает только десктопные Windows XP/Vista/7 (за исключением редакций Starter). Попытка запуска одной из утилит набора в серверной ОС приведет к ошибке Unsupported OS version. Возможна миграция с 32 на 64-битные системы, но наоборот нельзя (очевидно, в этом уже и нет острой необходимости).



Получаем список драйверов при помощи DISM

Вся работа с USMT состоит в использовании двух утилит ScanState (сбор файлов и параметров с исходного компьютера) и LoadState (восстановление среды). Теперь к ним добавлена еще одна – UsmtUtils, при помощи которой можно получить список поддерживаемых алгоритмов шифрования (/ec) или удалить жесткую ссылку из базы, сформированой ScanState (/rd).

К ранее поддерживаемому 3DES, в список алгоритмов шифрования данных, которые можно указывать в параметре /encrypt (ScanState) или /decrypt (LoadState), добавлен AES 128/192/256.

Но самым заметным нововведением USMT 4.0 является так называемая миграция жестких ссылок (Hard Link Migration), активируемая параметром /hardlink. Суть ее такова. Ранее чтобы сохранить и перенести пользовательские данные, их вначале каталогизировали, переносили на другой ресурс, а затем после установки ОС возвращали все обратно. Учитывая возможные объемы данных, которые хранил пользователь, весь процесс занимал достаточно много времени и требовал к тому же дополнительного места для хранения информации. Теперь после ввода команды:

```
> ScanState c:\store /o /c /i: app.xml /i: user.xml ↵
      /nocompress /hardlink
```

Вместо непосредственного копирования данных в каталоге c:\store создаются только жесткие ссылки. При переносе ОС жесткий диск будет очищен, кроме файлов, заблокированных такими ссылками. В итоге объем копируемых данных и время, затраченное на развертывание системы, значительно уменьшаются.

Кроме этого, при использовании USMT уже не требуется обязательное подключение к домену, а запуск ScanState можно производить из Windows PE. В связи с появлением Hard Link Migration в команде ScanState предусмотрен новый ключ /vsc, позволяющий задействовать службу теневого копирования (Volume Shadow Copy) для захвата заблокированных другими приложениями файлов.

В сценарии Config.xml (автоматически создается при помощи ScanState /genconfig) появились новые параметры и секции. Например, секция <HardLinkStoreControl> позволяет настроить Hard Link Migration.

```
<Policies>
 <HardLinkStoreControl>
  <fileLocked>
    <createHardLink>c:\Users\* [*]</createHardLink>
    <errorHardLink>C:\* [*]</errorHardLink>
  </fileLocked>
 </HardLinkStoreControl>
</Policies>
```

При этом:

- **createHardLink** – указывает на каталоги, для которых жесткая ссылка создается в любом случае, даже если файл заблокирован другим приложением;
- **errorHardLink** – в таких каталогах ссылка на заблокированный приложением файл может не создаваться.

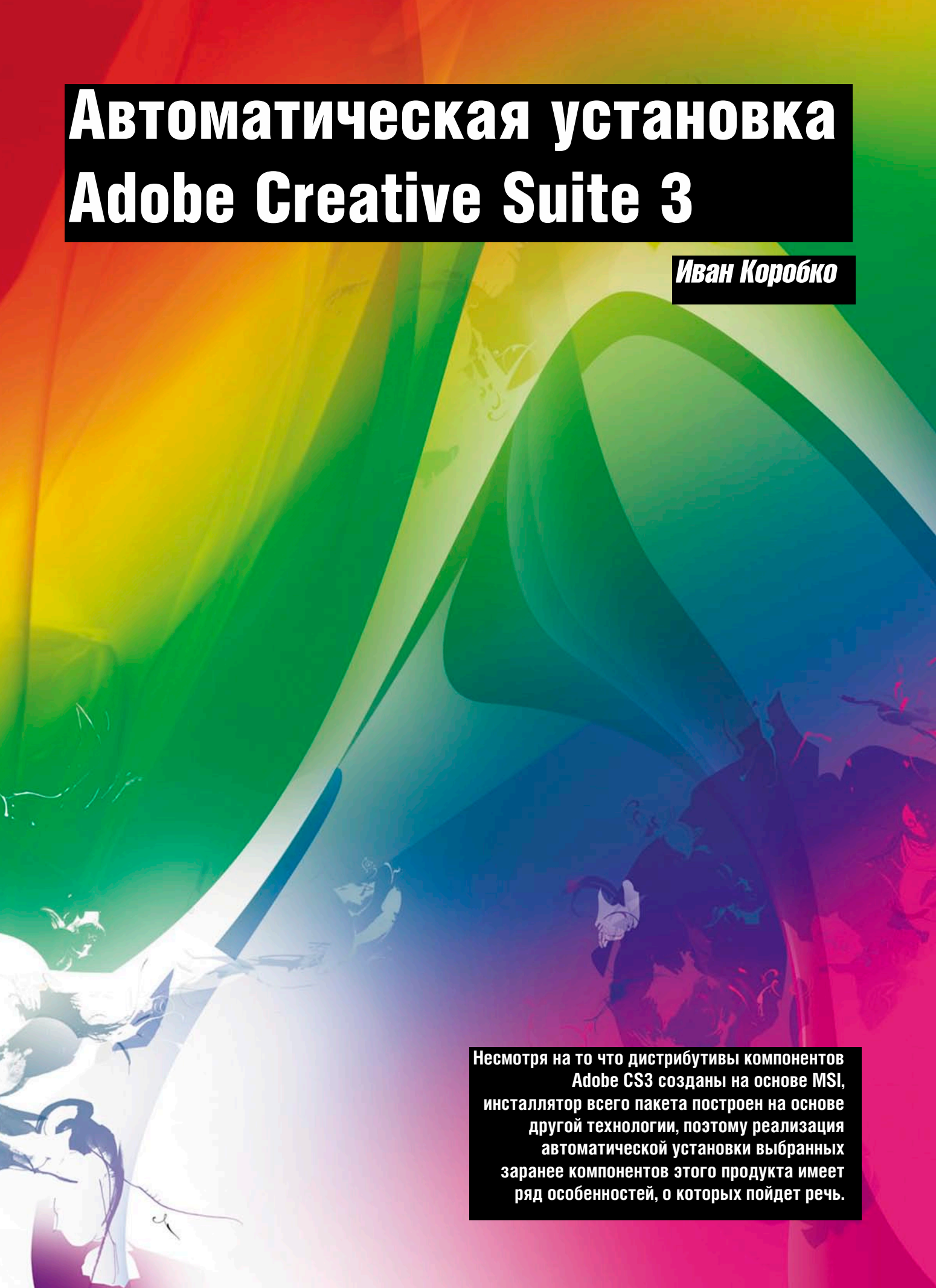
Для контроля миграции файлов по определенным критериям (размер, время создания и модификации и так далее) используется секция MigXmlHelper.FileProperties, поиск определенных документов на компьютере может быть организован при помощи MigXmlHelper.GenerateDocPatterns.

Как видите, разработчики пошли по пути унификации и упрощения использования пакета, что нельзя не приветствовать. Те же, кому хоть раз пришлось работать с USMT, должны оценить использование миграции жестких ссылок вместо копирования больших объемов информации. В статье затронуты только основные изменения, появившиеся в пакете автоматической установки WAIK для Windows 2008R2/7, более подробную информацию о релизе можно получить на ресурсах TechNet [2].

1. Бета-версия Windows AIK для Windows 7/2008R2 – http://go.microsoft.com/fwlink/?LinkId=136976.
2. Windows Automated Installation Kit for Windows 7 Beta – http://technet.microsoft.com/en-us/library/dd349343.aspx.

Автоматическая установка Adobe Creative Suite 3

Иван Коробко



Несмотря на то что дистрибутивы компонентов Adobe CS3 созданы на основе MSI, инсталлятор всего пакета построен на основе другой технологии, поэтому реализация автоматической установки выбранных заранее компонентов этого продукта имеет ряд особенностей, о которых пойдет речь.

Adobe CS3 достаточно большой по размерам пакет, и специалистов системной поддержки с администраторами больше волнует вопрос не автоматической установки, а выборочной автоматической установки. Это обусловлено тем, что ошибка в выборе перечня устанавливаемых программ обойдется примерно в час излишне потраченного времени. Поэтому стоит особое внимание уделить выборочной установке пакета.

Аргументы файла setup.exe

Традиционно мастер управления устанавливаемых компонентов запускается с помощью файла setup.exe, который поддерживает следующие аргументы:

- **--record.** Используется для создания файла ответов, принимает значение 0 или 1;
- **--silent.** Реализует установку приложения в «тихом режиме»;
- **--deployment.** Обеспечивает установку с помощью созданного файла ответов.

Структура файла ответов

Файл ответов представляет собой текстовый файл в формате XML. Как любой XML-файл, он содержит стандартный заголовок, в котором указаны используемая версия языка и кодировка файла (см. **рис. 1**).

Обратите внимание, что все конфигурационные файлы для Adobe Creative Suite имеют кодировку UTF-8.

Мастер установки по родительскому тегу <Deployment> определяет, что это файл ответов. В нем находятся два тега, логически разделяющие файл на две части. В первом теге – <Properties> – описываются параметры установки (папка, в которой будет находиться CS3, название продукта и т. д.), во втором – <Payloads> – компоненты (Photoshop, Illustrator, Indesign и т. д.).

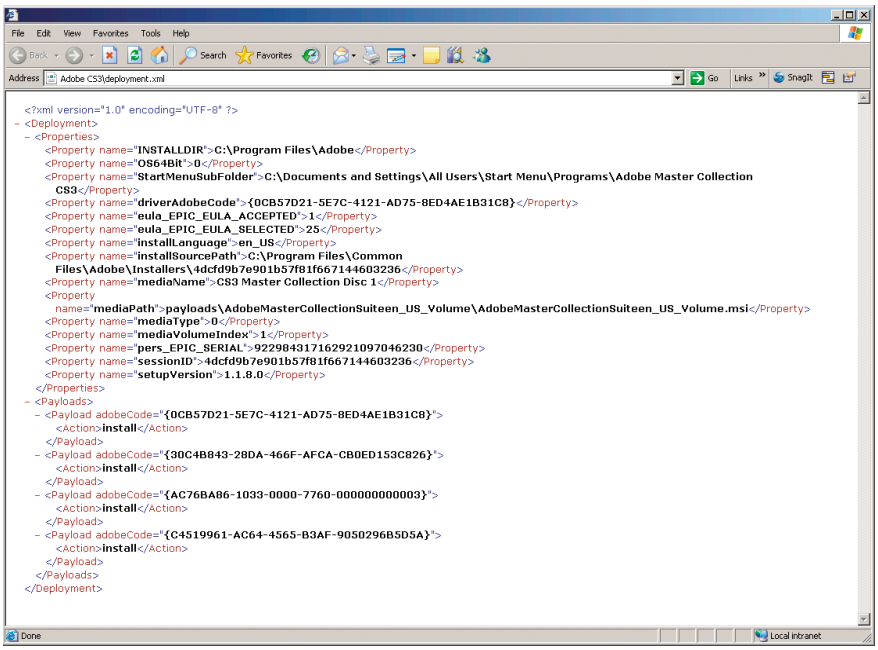


Рисунок 1. Пример файла ответов

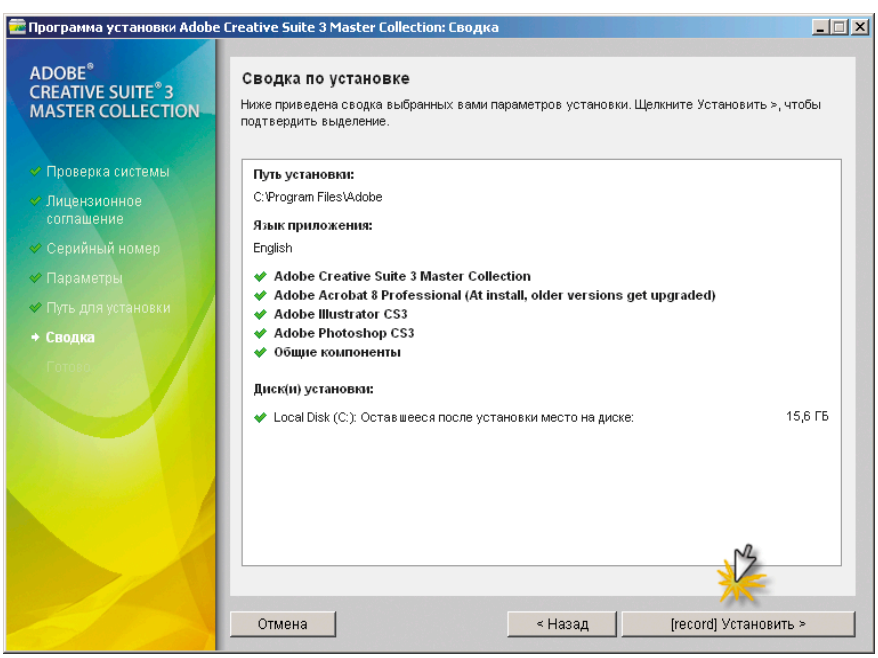


Рисунок 2. Запись файла ответов

Тег <Properties>

В теге <Properties> содержатся параметры установки приложения. Каждый из них описывается тегом <Property>, в котором название конкретного параметра указывается с помощью свойства name, а значение присваивается традиционным способом (см. **листинг 1**).

Листинг 1. Управление параметрами установки CS3

```
<Properties>
     <Property name="INSTALLDIR"> ↵
                    C:\Program Files\ADOBE CS3</Property>
     ...
</Properties>
```

К часто используемым параметрам можно отнести следующие:

- **INSTALLDIR.** Путь установки Adobe CS3.
- **OS64BIT.** Принимает значение 0 или 1. Если операционная система 64-битная (OS64BIT=1), то наряду с 32-битной версией Photoshop устанавливается 64-битная.
- **StartMenuSubFolder.** GUID компонента AdobeMasterCollectionSuiteen_US_Volume, отображаемого в папке «Установка и удаление программ», как Adobe Creative Suite 3 Master Collection. По умолчанию в Windows XP путь C:\Document and Settings\All Users\Start Menu\Programs\Adobe Master Collection CS3.

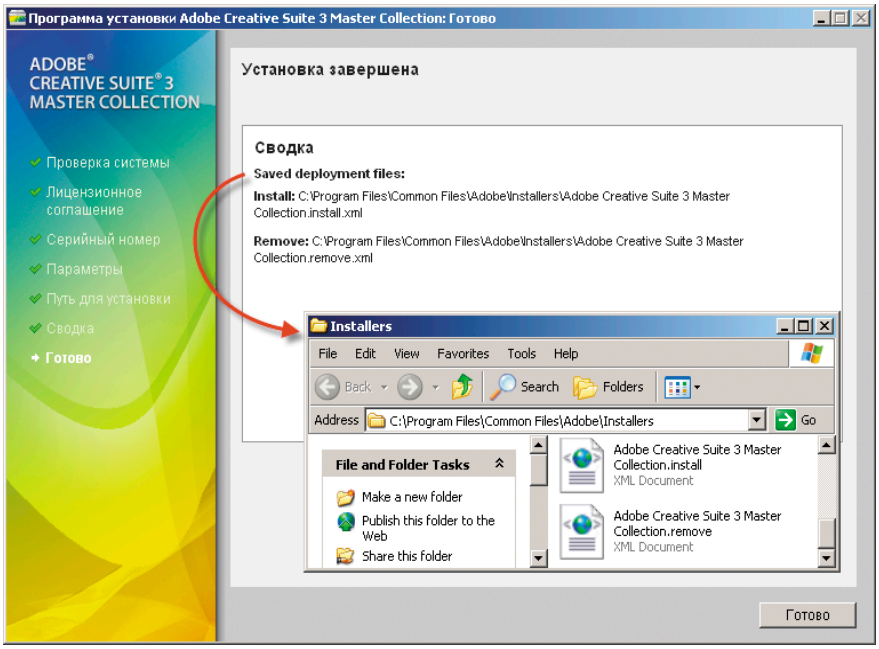


Рисунок 3. Местоположение файла ответов

- **eula_EPIC_EULA_ACCEPTED.** Всегда равен 1. Обозначает, что пользователь ознакомился и согласен с лицензионным соглашением.
- **pers_EPIC_SERIAL.** Серийный номер, хранящийся в зашифрованном виде.

Тег <Payloads>

В теге <Payloads> содержится список устанавливаемых приложений. Каждое из них идентифицируется двумя тегами: <Payload>, описывающего приложение с помощью GUID, и <Action>, указывающего производимую с ним манипуляцию (install, remove). Для инсталляции Adobe Photoshop CS3 (v. 10), которому соответствует GUID, равный {30C4B843-28DA-466F-AFCA-CB0ED153C826}, содержимое тега <Payloads> описано в **листинге 2**.

Листинг 2. Управление устанавливаемыми компонентами CS3

```
<Payloads>
      <Payload adobeCode = ↵
               "{30C4B843-28DA-466F-AFCA-CB0ED153C826}">
               <Action>install</Action>
      </Payload>
      ...
</Payloads>
```

Создание файла ответов

Файл ответов можно создать вручную, однако наиболее оптимальное решение – записать выполняемые действия мастера установки в XML-файл. Для включения мастера в режиме записи необходимо выполнить команду «setup.exe --record=1». Выполняемые действия ничем не будут отличаться от обычной работы мастера: необходимо согласиться с условиями лицензионного соглашения, затем ввести серийный номер, сформировать список устанавливаемых приложений, выбрать путь установки Adobe CS3. Исключение составляет последний шаг: вместо обычной кнопки «Установить» будет отображена кнопка «[record] Установить» (см. **рис. 2**).

При ее нажатии вместо установки приложения будут сформированы два файла: один для его инсталляции, второй – для деинсталляции. В обоих файлах содержимое <Properties> идентично, а <Payloads> отличается тем, что значение в теге <action> с install меняется на remove. Местоположение файлов показано на последнем шаге (см. **рис. 3**).

Использование файла ответов

Файл для автоматического удаления приложения используется достаточно редко, поэтому его можно удалить. Уделим особое внимание файлу, обеспечивающему автоматическую установку пакета. Его имя произвольно, поэтому присвоим ему более короткое имя – deployment.xml. Расположение файла также не имеет значения. Для удобства использования рекомендуется поместить его в один каталог с файлом setup.exe (см. **рис. 4**), обеспечивающим запуск мастера установки: --mode=Silent --deploymentFile=Deployment.xml. Рекомендуется эту команду поместить в командный файл, при этом необходимо помнить, что он должен запускаться с сетевого хранилища.

Приведенный на **рис. 4** листинг командного файла имеет особенности: поскольку установка выполняется в «тихом» режиме, то трудно понять, завершен процесс установки или нет. Воспользуемся свойством установки возвращать код ошибки по завершении процесса и командой «Start / Wait», которая будет выводить CMD-консоль, пока работа мастера не закончится. Команда pause позволит увидеть код ошибки.

Заключение

К сожалению, официальная документация [1] расходится с действительностью, тому свидетельство – многочисленные форумы на эту тему. Я надеюсь, что, прочитав статью, на все вопросы по автоматической установке Adobe CS3 вам удастся найти ответ.

1. Enterprise Deployment Options for Adobe® Creative Suite 3 Editions and Components – http://www.adobe.com/support/deployment/cs3_deployment.pdf.

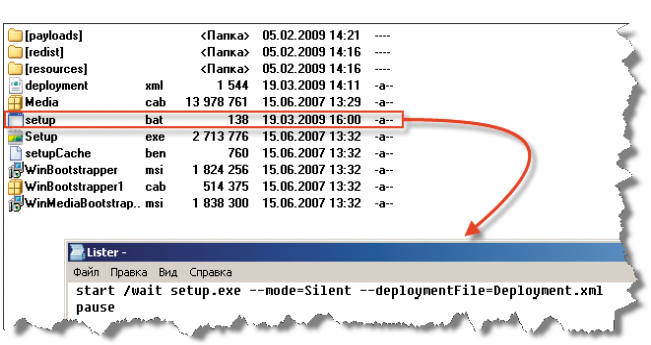
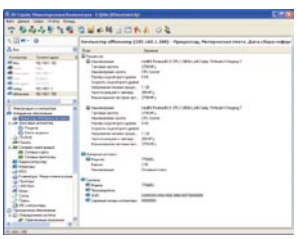
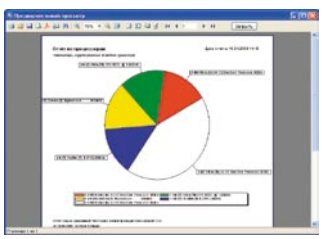


Рисунок 4. Использование файла ответов при установке CS3

«10-Страйк» – незаменимые программы для системных администраторов

«10-Страйк: Инвентаризация Компьютеров» – программа для учёта программного и аппаратного обеспечения на компьютерах в локальных сетях. Администратор сети может легко создать базу данных оборудования и программ, установленных на компьютерах, чтобы всегда иметь под рукой эту важную информацию.





Нажав пару кнопок, администратор может создать отчёты с конфигурациями компьютеров, списками установленных программ, ОС, и лицензий к ним. В программу заложены десятки шаблонов отчётов, можно создать собственные отчёты любой конфигурации, включая сводные таблицы. **«Инвентаризация Компьютеров»** имеет развитые средства подсчёта и контроля лицензий программного обеспечения.

При повторных проверках компьютеров программа сравнивает новую информацию с уже имеющимися данными и заносит изменения в отдельный журнал. Таким образом, можно легко увидеть, какие программы были добавлены или удалены пользователями, какое железо было унесено домой.

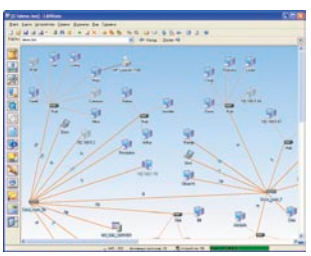
Системный администратор имеет возможность просматривать папки автозагрузки, списки процессов и служб на компьютерах пользователей, обнаруживать вредоносные программы.

При планировании апгрейдов программа позволяет создать отчёты, содержащие списки компьютеров с недостаточным объемом дисковой или оперативной памяти.

Сбор информации возможен как средствами WMI без установки дополнительных программ на удаленные компьютеры (при наличии прав администратора), так и с помощью вспомогательных программ и служб. Для более тщательного анализа администратор может выгрузить данные во внешнюю базу данных.

Таким образом, программа **«10-Страйк: Инвентаризация Компьютеров»** позволяет администратору всегда быть в курсе происходящего на подотчётных компьютерах, контролировать безопасность и лицензионную чистоту программ, создавать отчёты.

10-Strike LANState – программа мониторинга серверов и компьютеров в сети, позволяющая визуально наблюдать текущее состояние сети в любой момент времени. Администратор может вовремя узнать о произошедшем сбое (разрыв связи, завершение места на диске сервера, останов службы, и т. п.) и устранить проблему с минимальными потерями времени. LANState отслеживает состояние хостов, папок, файлов, баз данных, процессов и служб на компьютерах сети. Дополнительно администратор сети может использовать собственные скрипты в качестве проверок и даже контролировать различные переменные в управляемых коммутаторах по протоколу SNMP. Программа сигнализирует о неполадках с помощью звука, экранных сообщений, по e-mail, может запускать внешние программы и службы, а также перезагружать компьютеры для устранения неполадок.

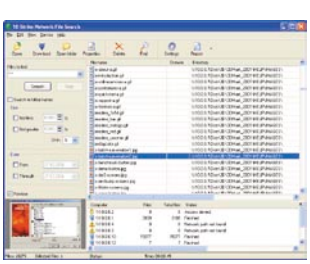


Но это еще не всё. LANState содержит развитые средства администрирования удаленных компьютеров: включение и выключение компьютеров, получение различной информации (списки программ, доступ к реестру, event log и т. п.), рассылка сообщений. Pro-версия программы имеет встроенный веб-сервер для удаленного доступа к карте сети.

«10-Страйк: Мониторинг Сети» – следит за хостами сети и оповещает о неполадках, как и LANState, но не имеет графической карты и работает в виде службы.

«10-Страйк: Схема Сети» – обнаруживает сетевые устройства и строит карту-схему сети. Используя SNMP-протокол, программа сама нарисует связи между устройствами.

10-Strike Network File Search – позволяет искать файлы в локальной сети (в сетевых папках и на FTP-серверах), в том числе в скрытых административных ресурсах. Новая версия 1.6 поддерживает поиск фраз внутри файлов.



Pro-версия осуществляет циклический мониторинг, заносит найденные файлы в лог, может удалять найденные файлы (например, вредоносные программы), а также содержит веб-сервер для просмотра списка найденных файлов удаленными пользователями.

10-Strike Connection Monitor – программа мониторинга подключений к общим папкам, позволяет узнать, кто, когда и что скачивал. Программа ведет лог подключений и оповещает администратора о различных событиях. Pro-версия программы реализована в виде службы.

Дмитрий Степанов

* * *

Для всех программ доступны для скачивания оценочные русскоязычные версии (под Windows) по адресу: **http://www.10-strike.com/rus**.

Делегируем права на перемещение учетных записей пользователей в Active Directory

Часть 2. Реализация основных функций

Вадим Андросов

В этой части статьи на примере создания надстройки нестандартного делегирования административных полномочий для Windows 2003 Server рассмотрим методики разработки классов COM на языках сценариев (VBScript). Также подробно рассмотрим программный доступ к спискам контроля доступа (ACL).

Главный класс надстройки

Итак, было решено разбивать процесс переноса пользователей на этапы: извлечение из целевого отдела, ожидание принятия в целевой отдел, принятие в целевой отдел. Кроме того, важную роль играет еще один необязательный этап: отмена перевода пользователя на этапе ожидания.

В соответствии с этими этапами и будем разрабатывать необходимые функции.

Предлагаемая надстройка является достаточно сложной, и в ходе ее реализации потребуется разработать ряд функций. Этим подпрограммам понадобятся общие данные (например, объект пользователя – менеджера по персоналу).

Одно из решений – использовать глобальные переменные, однако разрабатывается библиотека базовых функций, которые будут использоваться в надстройке. Способ их использования пока до конца не ясен, поэтому реализовывать библиотечные функции, зависящие от контекста, опасно.

Так, потребуется создать функцию инициализации контекста, а потом не забыть ее вызвать. Также не исключена ситуация, что могут потребоваться функции, работающие на основе разных контекстов.

Чтобы прочно и относительно безопасно связать функции с глобальным по отношению к ним контекстом, нужно оформить библиотеку в виде класса. В этом случае можно создавать и затем спокойно использовать любое количество его объектов.

Подробно процесс оформления и регистрации сценариев в виде COM-классов описан в [3].

Так будет выглядеть общий каркас разрабатываемого класса:

Листинг 1. Каркас класса UserMove.Engine

```
<?xml version="1.0"?>
<component>
<registration
      description="User moving routines"
      progid="UserMove.Engine"
      version="1.00"
      classid="{5452eeea-6f5e-42e0-b6ce-b99184ea0f68}"
>
</registration>
<public>
'методы, доступные пользователям класса
</public>
<reference guid="{97D25DB0-0363-11CF-ABC4-02608C9E7553}"/>
<object id="info" progid="ADSystemInfo"/>
<script language="VBScript">
<![CDATA[
'объявление глобальных переменных (контекста объекта)
'вызовы процедур инициализации объекта
'реализация методов
]]>
</script>
</component>
```

Все методы, разрабатываемые в статье, будут помещаться в этот класс. Возможно и другое безопасное решение – отказаться от контекста вовсе и передавать все необходимое функции в виде параметров. Однако в этом случае практически все функции требовали бы большого количества параметров, что отрицательно сказалось бы на читабельности кода.

Для начала обратим внимание на тег <reference guid="..."/>. Начав разрабатывать функции, я столкнулся с необходимостью использования большого количества констант ADSI. Раньше, когда их требовалось 2-3 штуки, особой проблемы не было. Достаточно было найти в MSDN [1] значения этих констант и заново объявить их в сценарии в виде:

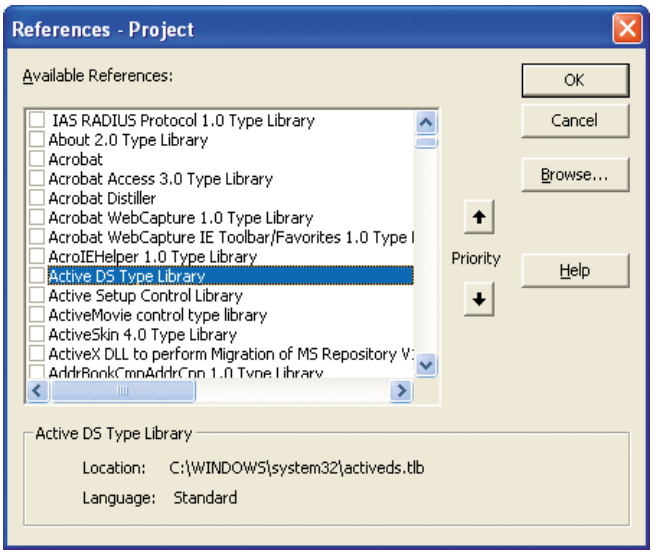


Рисунок 1. Подключение библиотеки типов в VBA

```
Const ADS_RIGHT_DS_DELETE_CHILD = 2
```

Назвав их таким же образом, как это было сделано в стандартных библиотеках, можно было без проблем пользоваться примерами программ из документации. Однако такое переопределение стандартных констант является очень опасной практикой.

Существуют более цивилизованные методы решения этой проблемы. В VBA это делалось посредством подключения необходимой библиотеки (см. **рис. 1**). Для ADSI она называется Active DS Type Library.

После этой операции можно пользоваться всеми константами ADSI, не объявляя их самостоятельно. То же самое можно сделать и в сценариях WSH с помощью тега reference. Только он требует указания не имени библиотеки, а ее идентификатора (GUID), который можно получить, используя специальные утилиты. Например, для этого можно воспользоваться программой Microsoft OLE/COM Object Viewer, которая поставлялась в комплекте еще с Visual Studio версии 6 (см. **рис. 2**).

Теперь в сценарии также можно пользоваться константами, определенными в библиотеке. Например, следующая строчка кода приведет к выводу на экран числа 2.

```
msgbox ADS_RIGHT_DS_DELETE_CHILD
```

Также интерес представляет тег:

```
<object id="info" progid="ADSystemInfo"/>
```

С его помощью объявляется и инициализируется глобальная переменная info класса ADSystemInfo. Того же эффекта можно было бы достичь, написав в разделе CDATA:

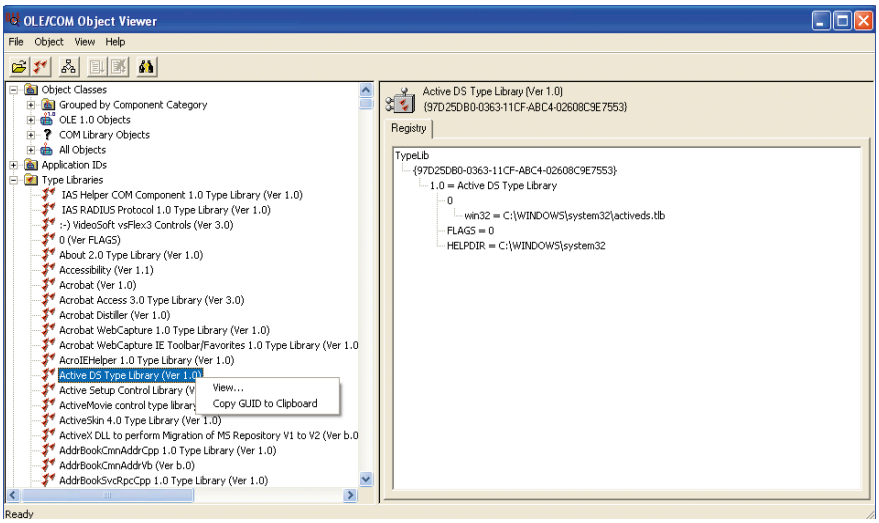


Рисунок 2. Получение GUID библиотеки

```
Dim info
Set info = CreateObject("ADSystemInfo")
```

Запись в виде тега позволяет наглядно и относительно компактно определить глобально используемые объекты. Ее преимущества проявляются разве что в случае использования различных языков сценариев при создании одного класса, когда всем им необходим доступ к одной переменной. Здесь я использовал ее исключительно в иллюстративных целях, чтобы показать, что так тоже можно.

Общие подпрограммы

Начнем с функции, которая будет отвечать на вопрос, а имеет ли текущий пользователь (предполагаемый менеджер по персоналу) права на манипуляции необходимыми объектами. Такие проверки придется делать достаточно часто, чтобы выбрать правильный режим функционирования надстройки.

Для начала определимся, что это за необходимые объекты. Для нашей надстройки это экземпляры классов пользователь (user), команда (UserMoveCommand), команда отмены операции (UserMoveDenyCommand) и команда начала перемещения (UserMoveStartMoveCommand).

Чтобы избежать трудноуловимых ошибок, имена всех классов запишем в соответствующие константы.

Листинг 2. Константы названий классов надстройки

```
Const ROOM_CLASS = "UserMoveWaitingRoom"
Const START_MOVE_COMMAND_CLASS = "UserMoveStartMoveCommand"
Const DENY_COMMAND_CLASS = "UserMoveDenyCommand"
Const COMMAND_CLASS = "UserMoveCommand"
Const CHAIR_CLASS = "UserMoveChair"
Const LINK_CLASS = "UserMoveChairLink"
```

Во-первых, пользователь должен иметь права на добавление и удаление пользователей из организационной единицы. Как выяснилось, для этого необходимо разрешение:

- Добавлять и удалять дочерние объекты перечисленных типов в организационную единицу (см. **рис. 3**).
- Право записи свойств объектов. Наличия только предыдущего набора прав не достаточно для переноса объектов пользователей из одной организационной единицы в другую. Дело в том, что в объекте хранится ссылка на контейнер, и нужно иметь право на ее изменение. Право на запись всех свойств пользователей является явно избыточным для операции переноса, однако вполне оправдано для других операций, которые входят в обязанности менеджера по персоналу. Поэтому я решил не искать здесь более «тонкого» способа наделения полномочиями (см. **рис. 4**).

Поскольку требуется написать подпрограмму проверки наличия прав работы с несколькими типами объектов, их удобно сгруппировать в один массив. Инициализироваться он будет один раз при создании экземпляра класса надстройки.

Рассмотрим метод, который это делает.

Листинг 3. Метод инициализации основного класса надстройки

```
sub initialize()
  timeZoneOffset = "?"
  set re = new Regexp
  re.ignoreCase = True
  re.global = True
  delegationClasses = Array(getClassGUID("user"), ↵
      getClassGUID(COMMAND_CLASS), ↵
      getClassGUID(DENY_COMMAND_CLASS), ↵
      getClassGUID(START_MOVE_COMMAND_CLASS))
end sub
```

Итак, можно видеть, что массив, содержащий классы для проверки, называется delegationClasses. Также в этом методе инициализируется ряд вспомогательных переменных, используемых в других методах класса. Конечно, все они должны быть объявлены выше.

```
dim timeZoneOffset, re, delegationClasses
```

Затем нужно не забыть вручную вызвать метод инициализации, потому что автоматически этого, к сожалению, не происходит.

```
Initialize
```

Этот вызов просто записывается выше всех объявлений функций. В массиве хранятся идентификаторы классов, так как при анализе списка контроля доступа необходимы будут именно они. GUID класса пользователя можно найти в MSDN [1]: {BF967ABA-0DE6-11D0-A285-00AA003049E2}.

Но для наших классов его там, естественно, нет. Причем это не тот идентификатор, который генерировался при создании классов [4]. Для получения GUID класса по его имени пришлось написать специальную функцию. Рассмотрим ее.

Листинг 4. Определение GUID класса

```
function getClassGUID(className)
  dim classObj
```

```
    set classObj = getObject("LDAP://schema/" & className)
    getClassGUID = GUID2Str(classObj.schemaIDGUID)
  end function
```

Функция подключается к объекту заданного класса, используя провайдер LDAP. Например, путь к классу пользователя имеет вид LDAP://schema/user. Желанный идентификатор хранится в свойстве schemaIDGUID полученного объекта. Однако он имеет тип Octet String, который в VBScript не поддерживается. Для его преобразования в обычную строку используется функция GUID2Str. Рассмотрим теперь ее.

Листинг 5. Преобразование Octet String к обычной строке

```
Function GUID2Str(Guid)
  Dim i, b(16)
  For i = 1 To 16
    b(i - 1) = Right("0" & Hex(Ascb(Midb(Guid, i, 1))), 2)
  Next
  GUID2Str = "{" & b(3) & b(2) & b(1) & b(0) & ↵
      "-" & b(5) & b(4) & ↵
      "-" & b(7) & b(6) & "-" & b(8) & b(9) & "-"
  for i = 10 to 15
    GUID2Str = GUID2Str & b(i)
  next
  GUID2Str = GUID2Str & "}"
End Function
```

Функция выглядит сумбурно, потому что писалась «методом тыка». С помощью первого цикла строка Octet String превращается в массив байт b. Для этого с помощью функции Midb из нее выделяется один байт в заданном месте, который потом преобразуется в число функцией Ascb (возвращает первый байт переданного ей символа). Полученное число переводится в шестнадцатеричный формат (функция Hex), которому в начало добавляется 0. Последнее нужно потому, что каждый байт должен представляться двузначным шестнадцатеричным числом (1A, FF, D5). Поэтому когда получилось, скажем, F, то ему в начало добавляется 0. Если число сразу было двузначным, то ноль станет третьим лишним символом, который будет отсечен с помощью функции right.

Эксперименты с системными классами (для которых мне был известен правильный идентификатор) показали, что порядок полученных байт в результате немного меняется.

Получаемая строка состоит из 4-х групп. В первую входят первые четыре байта исходной строки в обратном порядке, во вторую – 5-й и 6-й байты тоже в обратном (в программе это 4-й и 5-й элементы массива, поскольку нумерация начинается с нуля), затем еще два байта в обратном порядке, далее два байта в прямом порядке и, наконец, последние 6 снова в прямом. Группы должны быть разделены дефисами. Весь идентификатор заключается в фигурные скобки.

Несмотря на способ написания и внешний вид, функция корректно работает. Полученные с ее помощью идентификаторы дополнительных классов без проблем воспринимались системой.

Переменная re будет использоваться для работы с регулярными выражениями, а timeZoneOffset – содержать смещение временной зоны для текущего часового пояса. Но это потом, а сейчас перейдем к методу проверки полномочий.

Листинг 6. Проверка прав управления подразделением

```
function canHeManageOU(ou, user)
dim canChange, canMoveChild, i
canHeManageOU = true

for i = 0 to UBound(delegationClasses)
  canChange = canDo(ou, user, ↵
      ADS_RIGHT_DS_WRITE_PROP, delegationClasses(i), true)
  canMoveChild = canDo(ou, user, ↵
      ADS_RIGHT_DS_DELETE_CHILD Or ↵
      ADS_RIGHT_DS_CREATE_CHILD, ↵
      delegationClasses(i), false)
  canHeManageOU = canHeManageOU and ↵
      (canChange And canMoveChild)
next

end function
```

Функции передается два параметра – объекты «Организационная единица» и пользователя, права которого проверяются. Функция очень простая, поскольку вся реальная работа сосредоточена в более низкоуровневых подпрограммах, которые будут рассмотрены позже.

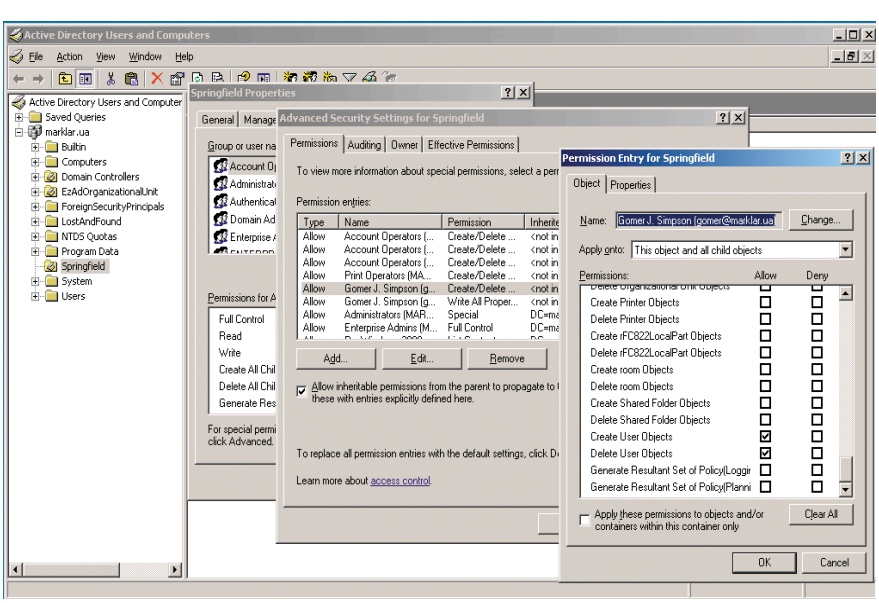


Рисунок 3. Права на создание дочерних объектов пользователей

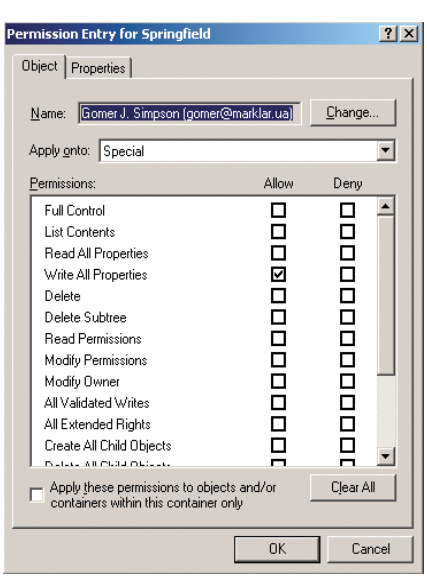


Рисунок 4. Право изменения свойств объектов пользователей

Вся работа метода сводится к последовательной проверке необходимых прав для всех объектов массива delegationClasses посредством вызовов функции canDo. Сначала проверяется возможность изменения свойств объектов (константа ADS_RIGHT_DS_WRITE_PROP), а затем – добавления и удаления пользователей из контейнера (комбинация констант ADS_RIGHT_DS_DELETE_CHILD и ADS_RIGHT_DS_CREATE_CHILD). То есть всю полезную работу выполняет функция canDo. Она является вспомогательной и в качестве открытого метода класса не используется в отличие от canHeManagePath. Чтобы сделать его доступным пользователям класса, нужно добавить его в раздел public описания (см. **листинг 1**).

```
<method name="canHeManagePath">
      <PARAMETER name="ouPath"/>
      <PARAMETER name="whoPath"/>
</method>
```

Метод представляет собой простую обертку для функции canHeManageOU (см. **листинг 6**), которая получает в качестве параметров не сами объекты, а пути к ним (ADSI Path).

```
Листинг 7. Проверка наличия прав управления организационной
единицей

function canHeManagePath(ouPath, whoPath)
  dim ou, user
  set ou = getObject(ouPath)
  set user = getObject(whoPath)
  canHeManagePath = canHeManageOU(ou, user)
end function
```

В подпрограмме создаются экземпляры необходимых объектов и вызывается canHeManageOU. Также в классе определено еще два метода-обертки для canHeManageOU, позволяющих проверять права доступа к организационной единице текущего пользователя.

```
Листинг 8. Дополнительные функции проверки прав

function canCurrentManagePath(ouPath)
  dim ou
  set ou = getObject(ouPath)
  canCurrentManagePath = canCurrentManageOU(ou)
end function

function canCurrentManageOU(ou)
  dim user
  set user = getObject("LDAP://" & info.userName)
  canCurrentManageOU = canHeManageOU(ou, user)
end function
```

Здесь info – глобальный объект типа ADSystemInfo (см. **листинг 1**). Его свойство username содержит путь к объекту текущего пользователя.

Далее перейдем к реализации функций для работы с ACL. Список контроля доступа (Access Control List) сопоставлен каждому объекту Active Directory, он состоит из набора записей (Access Control Entry или ACE), которые собственно и определяют уровень доступа пользователя или группы к ресурсу. Эта тема достаточно обширна, при необходимости получить дополнительную информацию можно из соответствующих источников [1, 2]. Windows предоставляет графические средства редактирования ACL (см. **рис. 3**).

Подробно рассмотрим работу функции canDo. Ей передается три параметра:

- **oper.** Проверяемая операция. То есть то действие, возможность выполнения которого требуется проверить. Для обозначения действий используются константы из библиотеки Active DS Type Library.
- **targetClass.** Идентификатор целевого класса, допустимость работы с объектами которого проверяется.
- **isInherited.** Параметр логического типа. Запись контроля доступа может описывать как разрешения, относящиеся непосредственно к объекту, к которому прикреплен ACL, так и его непосредственным потомкам. Разрешение на добавление и удаление пользователей из организационной единицы, это относится к контейнеру, поэтому значение этого параметра будет истинно. В то же время право на изменение свойств вложенных объектов уже относится скорее к ним, чем к организационной единице. В этом случае значение – ложь. На что влияет параметр, станет понятно из реализации.

```
function canDo(oper, targetClass, isInherited)
  canDo = false
  Dim sec, acl, ace
```

В начале работы функция осуществляет привязку к списку контроля доступа организационной единицы. Для этого нужно сначала подключиться к объекту ntSecurityDescriptor организационной единицы, а затем уже с его помощью получить список DiscretionaryAcl.

```
  Set sec = ou.Get("ntSecurityDescriptor")
  Set acl = sec.DiscretionaryAcl
  dim result
```

Затем нужно проверить все записи (ace) списка. Удобнее всего это делать с помощью оператора цикла For Each, основное предназначение которого как раз обход коллекций.

Проверка каждой записи осуществляется с помощью функции checkACE, которая возвращает одну из трех констант (эти константы определяются в классе, их конкретные значения могут быть любыми, лишь бы разными, у меня это было 1, 2 и 3):

- **CHECK_ACE_SKIP.** Проверенная запись контроля доступа не относится ни к проверяемому пользователю, ни к содержащей его группе. Или же запись относится к нужному пользователю, но управляет несущественными в данном контексте правами, то есть не соответствующими параметру oper. Такие записи при проверке игнорируются.
- **CHECK_ACE_YES.** Запись относится к проверяемому пользователю и разрешает действие, заданное в oper.
- **CHECK_ACE_NO.** Запись относится к проверяемому пользователю и запрещает действие, заданное в oper.

Встретив разрешение или запрет операции, функция сразу завершает работу, возвращая соответствующий результат. Дальше проверять список нет необходимости. Опе-

рационной системой гарантируется, что запись о явном запрете встретится раньше разрешения. Так что, встретив запись с разрешением, можно быть уверенным, что далее записи с запретом этой же операции уже не будет.

```
  For Each ace In acl
   result = checkACE(ace, oper, targetClass, isInherited)

   if result <> CHECK_ACE_SKIP then
     canDo = (result = CHECK_ACE_YES)
     exit function
   end if

  Next
end function
```

Функция checkACE в качестве первого параметра принимает проверяемую запись списка контроля доступа, остальные параметры передаются от canDo.

Приведенная функция также не относится к самому низкому уровню реализации, она использует подпрограмму проверки записи контроля доступа checkACE. Рассмотрим теперь и эту функцию.

Логика ее работы и возвращаемые значения уже были описаны выше, поэтому расскажу об особенностях реализации.

```
function checkACE(ace, oper, targetClass, isInherited)
  checkACE = CHECK_ACE_SKIP
  if not isTrusteeInteresting(ace.Trustee) then exit function
```

Сначала функция проверяет свойство Trustee записи контроля доступа. Оно определяет субъект доступа, к которому относится запись. Функция продолжает работу только в том случае, если это или пользователь, права которого сейчас проверяются, или группа, содержащая этого пользователя.

В противном случае функция возвращает константу CHECK_ACE_SKIP, извещающую о том, что текущую запись нужно пропустить.

```
  dim classGUID
  if isInherited then
    classGUID = ace.InheritedObjectType
  else
    classGUID = ace.ObjectType
  end if
```

Затем в зависимости от значения параметра isInherited определяется объект доступа. Это или дочерние объекты (используется при проверке прав на добавление и удаление объектов пользователей в организационной единице), или текущий объект (проверка возможности изменения свойств пользователей).

Объект доступа в данном случае – это идентификатор класса, права на манипуляции экземплярами которого предоставляются.

```
  if isMask(ace.accessMask, oper) then
```

Дальнейшая работа будет выполнена только в том случае, если текущая запись контроля доступа описывает проверяемые права. Нужно убедиться с помощью функции isMask, работа которой будет рассмотрена ниже, что за-

пись содержит информацию о необходимой операции (параметр функции oper).

```
      if ace.AceType = ADS_ACETYPE_ACCESS_DENIED then
        checkACE = CHECK_ACE_NO
        exit function
      end if

      if ace.AceType =
         ADS_ACETYPE_ACCESS_ALLOWED  then
           checkACE = CHECK_ACE_YES
           exit function
      end if
```

Далее проверяем, что целевой объект представляет собой интересующий нас класс (его идентификатор передается в качестве параметра функции).

```
      if classGUID = targetClass then
        if ace.AceType =
          ADS_ACETYPE_ACCESS_DENIED_OBJECT then
            checkACE = CHECK_ACE_NO
            exit function
        end if

        if ace.AceType =
           ADS_ACETYPE_ACCESS_ALLOWED_OBJECT then
             checkACE = CHECK_ACE_YES
             exit function
        end if

      end if
    end if
end function
```

Основная логика работы уже рассмотрена, осталось только несколько вспомогательных функций. isMask – функция логического типа, возвращает истину, если в поле (первый параметр) установлен заданный флаг (второй параметр):

```
function isMask(mask, flag)
  isMask = (mask and flag) <> 0
end function
```

Рассмотрим последнюю вспомогательную функцию – isTrusteeInteresting. Она возвращает истину, если субъект доступа записи контроля доступа относится к проверяемому пользователю.

```
function isTrusteeInteresting(trustee)
isTrusteeInteresting = false

if trustee = (info.domainShortName & "\" & ↵
      user.samAccountName) then
  isTrusteeInteresting = true
  exit function
end if
```

Пользователь содержится в поле trustee записи контроля доступа в виде «ДОМЕН\ПОЛЬЗОВАТЕЛЬ». Проверку этого случая и осуществляет первый условный оператор. Если пользователь совпал с проверяемым, функция завершает работу, возвращая истину.

Однако одной такой проверки недостаточно: запись может относиться не только к пользователю, но и к содержащей его группе.

```
dim trusteeObj
if instr(1, Trustee, ↵
      info.domainShortName) <> 1 then exit function
```

Однако сначала нужно убедиться, что запись содержит в поле trustee объект, относящийся к проверяемому домену. Дело в том, что кроме записей для пользователей и групп существуют специальные экземпляры, описывающие доступ для различных системных объектов (например, NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERS), которые при решении текущей задачи нужно пропускать. Функция завершает работу и возвращает ложь (это значит, что текущая запись «не интересна»), если в значении Trustee не содержится имени домена.

```
set trusteeObj = getObject("WinNT://" & ↵
      replace(trustee, "\", "/"))
```

Затем происходит привязка к объекту Trustee. Обратите внимание, здесь я использую провайдер WinNT. Он менее функционален, чем LDAP, однако в данном случае более удобен, так как позволяет привязываться к объектам, игнорируя их положение в структуре предприятия. В наличии имеется только имя Trustee в домене. Чтобы воспользоваться провайдером LDAP потребовалось бы сначала произвести поиск объекта в иерархии, чтобы установить его отличительное имя (DN), что в данной ситуации обернулось бы только неоправданным разрастанием кода. Функция replace требуется здесь для того, чтобы привести имя Trustee к виду, пригодному для привязки с помощью провайдера WinNT. Собственно оно почти подходит, за исключением того, что нужно заменить обратный слеш на прямой.

```
if trusteeObj.class = "Group" then
```

Далее, если Trustee – группа (еще это может быть просто другой пользователь), нужно проверить с помощью метода isMember принадлежит ли ей текущий пользователь, если принадлежит, то функция возвращает истину.

```
    if trusteeObj.isMember("WinNT://" & ↵
       info.domainShortName & "/" & ↵
       user.samAccountName) then
         isTrusteeInteresting = true
         exit function
    end if
  end if
  end function
```

Итак, в этой части статьи были рассмотрены вопросы создания класса COM на языке программирования сценариев. Также были освещены базовые аспекты работы со списками контроля доступа. В следующей статье будет продолжена разработка надстройки. Будет реализована программная модификация списков контроля доступа и автоматическое делегирование полномочий.

1. msdn.microsoft.com.
2. Чарли Рассел, Шарон Кроуфорд, Джейсон Джеренд. «Windows server 2003 +SP1 и R2. Справочник администратора». – М.: Издательство «ЭКОМ», 2006 г. – 1424 с.
3. Don Jones, Jefferey Hicks. «Advanced VBScript for Microsoft Windows Administrators». – Washington: Microsoft Press, 2006 г.
4. Андросов В. Делегируем права на перемещение учетных записей пользователей в Active Directory. Часть 1. Постановка задачи. //Системный администратор, №2, 2009 г. – С. 16-21.

Синхронизируем данные между компьютерами с помощью сервисов сетевого хранения

Виталий Банковский

С появлением недорогих лаптопов и тем более ультрадешевых неттопов эти «создания» начали плодиться во всех местах моего обитания – дома, в офисе, на даче, диване и так далее. И тогда начались проблемы с синхронизацией данных между компьютерами. Знакомо?

И вот тогда, чтобы окончательно не сойти с ума с синхронизацией данных на всех этих устройствах, я задался вопросом, как хранить на каждом из них актуальную копию моих данных. Хранение данных на одном сервере или постоянное копирование данных с помощью rsync я отмел как неудобное и, как следствие, неработающее решение.

Немного побродив по Интернету, я обнаружил, что шустрые предприниматели мира сего уже пришли мне на помощь с тремя наиболее интересными продуктами:

- DriveBox;
- LiveDrive;
- SkyDrive.

Краткие сравнительные характеристики смотрите в **таблице**.

Увы, я не смог дать 5 баллов сервису DropBox по причине отсутствия асимметричного шифрования.

Будучи заядлым пользователем операционной системы Linux, я похоронил для себя LiveDrive и SkyDrive и приступил к использованию программы DropBox.

В начале эксплуатации я также поставил себе задачу, чтобы на сервис хранения данных мои файлы отправлялись в зашифрованном виде.

Начало работы

Стоит отметить легкость установки клиента DropBox даже на Linux-подобных системах. Для того чтобы начать использовать систему, нужно проделать всего несколько шагов:

- Зарегистрироваться на сайте http://drivebox.com.
- Скачать и установить, следуя инструкциям на сайте поставщика, клиентское программное обеспечение, предварительно получив его с вышеуказанного сайта.
- Перезапустить менеджер файлов Nautilius.

При этом в домашней папке пользователя появится каталог .DriveBox, в который и нужно размещать файлы для их синхронизации с другими компьютерами.

Безопасность

Как я уже говорил, при стандартном использовании наши данные попадают на сервис DropBox и уже там шифруются. Конечно, при таком способе хранения есть вероятность сбоя сервиса, и данные могут быть доступны другим пользователям, не говоря уже о ситуациях, когда данные с сервиса воруют хакеры или инсайдеры. Для устранения этого недостатка воспользуемся богатыми возможностями семейства Linux, а именно – программой cryptfs. Эта программа позволяет прозрачно шифровать каталоги, причем зашифрованные файлы и каталоги также представлены в виде файлов и каталогов.

Итак, приступим. В папке ~/.Dropbox/ создаем каталог Private, куда будут размещаться наши уже зашифрованные данные. Также создаем каталог, например, ~/Dropbox.Open, куда мы будем размещать наши файлы в открытом виде. Затем монтируем каталог ~/Dropbox/Private на каталог ~/Dropbox.Open с помощью программы encfs:

```
encfs  ~/.Dropbox/Private ↵
       ~/Dropbox.Open
```

Если такая процедура происходит в первый раз, то будет задан вопрос по выбору глубины шифрования, и также будет предложено задать пароль для шифрования данных. Аналогично повторяем процедуры установки и монтирования на всех остальных компьютерах, которые должны содержать синхронизированные данные.

Использование зашифрованных данных

Для того чтобы программы на разных компьютерах использовали данные

из сетевого хранилища, необходимо перенести файлы этих программ в папку ~/Dropbox.Open. Например, для синхронизации системы мгновенных сообщений licq мы переносим папку ~/.licq в каталог ~/Dropbox.Open и создаем символическую ссылку:

```
mv ~/.licq ~/Dropbox.Open/
ln -s ~/Dropbox.Open/.licq ~/.licq
```

Аналогично поступаем и с остальными нашими программами.

Повышаем безопасность

Будучи системным администратором, я все-таки побоялся расположить наиболее критичные данные на публичном сервере, пусть даже и в зашифрованном виде. Причин для этого несколько:

- Так как на сервере будут хранится файлы стандартных программ, то злоумышленнику будет проще взломать пароль, если он возьмет зашифрованный файл с сервиса DropBox и незашифрованный стандартный файл, используемый в одной из наших программ.
- Сохранность данных. Если сотрудники вашей компании пользуются таким сервисом, то невозможно централизованно управлять созданием архивных данных пользователей.
- Политика безопасности многих компаний запрещает размещать важные данные в каком-либо виде на серверах сторонних компаний.

При наличии «собственного» защищенного сервера и канала VPN для подключения к этому серверу можно использовать кеширующую файловую систему Coda.

Coda использует локальный кеш для доступа к данным, когда сетевое подключение к серверу потеряно, при этом сохраняя список изменений в локальных файлах. Когда соединение восстановлено, локальный клиент Coda синхронизирует локальные данные с сервером.

С программой можно ознакомиться на домашней странице проекта по адресу http://www.coda.cs.cmu.edu, а теперь мы сосредоточимся на процедурах установки, настройки и использования.

На своих рабочих компьютерах я использую Debian, тогда как мой «собственный» сервер с Coda работает под дистрибутивом CentOS, поэтому описание процедуры установки и настройки будет ориентировано на эти дистрибутивы. При таком «зоопарке» мне пришлось обратится к установке программы Coda из исходных кодов.

Процедура установки программы Coda

Программа Coda состоит из трех подсистем:

- серверная часть;
- клиентская часть;
- поддержка в ядре на сервере и клиенте.

Установка и настройка серверной части

Предварительно получаем и устанавливаем следующие библиотеки:

- LWP;
- RPC2;

Сравнение сервисов сетевого хранения

	DropBox	LiveDrive	SkyDrive
Windows	Да	Да	Да
Linux	Да	Нет	Нет
MacOS	Да	Нет	Нет
Шифрование между клиентом и сервисов	Да	Да	Да
Шифрование данных на стороне сервиса	Да	Нет	Нет данных
Субъективная оценка сервиса	4	0	0

- RVM.

Эти библиотеки доступны по адресу ftp://ftp.coda.cs.cmu.edu/pub/coda/linux/src и устанавливаются стандартным способом, последовательно выполняя три команды:

```
./configure
make
make installing
```

Включение поддержки Coda в ядре системы Linux

Модуль поддержки файловой системы Coda входит в стандартное ядро Linux, включение которого находится в разделе File systems, Network filesystem. На момент написания статьи я использовал ядро Linux версии 2.6.24.2 на клиенте и 2.6.22.12 на сервере. После включения поддержки необходимо загрузить модуль в ядро или перегрузить сервер, если этот модуль «coda» вкомпилирован в новое ядро.

Установка и настройка демона Coda

Как обычно, получаем исходные коды демона Coda по адресу ftp://ftp.coda.cs.cmu.edu/pub/coda/linux/src, компилируем и устанавливаем:

```
tar -xzvf coda-x.x.x.tar.gz
cd coda-x.x.x
./configure
make
make install
```

На момент написания статьи была использована версия 6.9.4 демона Coda.

Далее нам необходимо настроить серверную часть программы Coda. Для этого в состав программы входит утилита vice-setup, которая при запуске задаст несколько вопросов:

Путь, где программа Coda будет хранить свои рабочие файлы:

```
What is the root directory for your coda server(s)? [/vice]
```

Является ли данный сервер главным сервером. Программа Coda поддерживает многосерверную конфигурацию в режиме репликации:

```
Is this the master server, aka the SCM machine? (y/n) y
```

Настройка идентификации для коммуникации между несколькими серверами Coda и между клиентом Coda и сервером:

Пароль для идентификации между серверами Coda:

```
Enter a random token for update authentication : **********
```

Пароль, который должны указывать клиенты для идентификации на сервисе Coda:

```
Enter a random token for auth2 authentication : **********
```

Пароль, который нужно использовать при работе с утилитой volutil. Эта утилита с помощью вызовов RPC позволяет удаленно работать с томами файловой системы Coda.

```
Enter a random token for volutil authentication : **********
```

Так как я являлся единственным пользователем, то использовал один и тот же пароль для всех трех подсистем идентификации. Подробнее о системе репликаций файловой системы Coda и об управлении томами можно прочитать на домашней странице проекта по адресу http://www.coda.cs.cmu.edu.

Идентификатор данного сервера Coda, произвольное между 0 и 255, исключая 0,127 и 255. Я использовал 100.

```
Enter an id for the SCM server. (hostname server.domain.com)

The serverid is a unique number between 0 and 255.
You should avoid 0, 127, and 255.
serverid: 100
```

Создание учетной записи администратора Coda. Нужно указать ID и имя учетной записи пользователя, который уже существует в системе:

```
Enter the uid of this user: 1000
Enter the username of this user: cooluser
```

Указание имени файла, где будут храниться лог-файлы. Coda поддерживает разделы raw на системах хранения и обычные файлы на файловой системе.

```
What will be your log file (or partition)? /home/coda/log
```

Задание файла, где будут храниться данные:

```
Where is your data file (or partition)? /home/coda/data
```

Далее будут заданы вопросы про размер данных и максимального количества файлов, затем утилита завершит настройки и запустит серверные подсистемы программы Coda.

Если все нормально было установлено и настроено, то мы должны получить сообщение об успешной установке и запуске серверной части программы Coda:

```
That seems to have worked...

If you have a working Coda client you should now be able to

access the new Coda realm

 - cfs lv /coda/servername.domain.com/



enjoy Coda.

 for more information see http://www.coda.cs.cmu.edu.
```

Установка и настройка программы Coda на клиентском компьютере

Аналогично устанавливаем библиотеки LWP, RPC2, RVM, программу Coda, включаем поддержку файловой системы Coda в ядре.

Далее необходимо настроить и подключиться к серверу с сервисом Coda с помощью утилиты venus-setup:

```
venus-setup servername.domain.com
```

где servername.domain.com – имя сервера или IP-адрес сервера, где мы установили сервис Coda. При запуске программы будут заданы два вопроса:

Имя сервера, которое мы уже указали в качества параметра для утилиты venus-setup:

```
Default realm for authentication:
```

Размер локального кэша (эквивалентен размеру хранимых на сервере данных) в килобайтах:

```
Amount of diskspace used for caching:
```

По завершении работы этой утилиты можно увидеть следующее сообщение:

```
22:25:17 Mounting root volume...

22:25:17 Venus starting...

22:25:17 /coda now mounted.
```

Это значит, что файловая система Coda успешно смонтирована в каталог /coda. Но перед началом работы необходимо пройти идентификацию на сервере, чтобы мы могли записывать в этот каталог. Для этого существует утилита clog, которая запускается следующим образом:

```
clog username
```

где username – имя учетной записи пользователя службы Coda на сервере Coda. При этом будет запрошен пароль, причем нужно использовать тот же пароль, который был задан при настройке программы Coda на сервере:

```
Enter a random token for auth2 authentication:
```

Если все было сделано правильно, то наше хранилище на сервере будет смонтировано в каталог /coda/servername.domain.com/. В этот каталог и нужно записывать наши данные, которые должны быть синхронизированы между компьютерами.

Также существует возможность использования клиента Coda для семейства операционных систем Windows XP, NT, используя среду Cygwin. Более подробно об этом можно почитать по адресу http://www.coda.cs.cmu.edu/windowsnt.html. У меня не было возможности рассказать об этом подробнее по причине «финальной кончины» Windows XP на моем компьютере.

В этой статье я показал два подхода к синхронизации данных между несколькими компьютерами, и вам решать, какой из вариантов наиболее подходит в вашей ситуации.

1. http://www.coda.cs.cmu.edu.
2. http://en.wikipedia.org/wiki/Coda_(file_system).
3. http://dropbox.com.
4. http://skydrive.com.
5. http://www.livedrive.com.
6. http://www.coda.cs.cmu.edu/windowsnt.html.

PowerShell. Определяем имя текущего домена

Иван Коробко

При подключении к каталогу Active Directory первый шаг – определение имени текущего домена. Для этого используется виртуальный объект RootDSE.

PowerShell позиционируется компанией Microsoft как скриптовый язык, поддерживающий библиотеки .Net Framewok и призванный заменить существующие VBScript и JScript. Принципиальное отличие PowerShell от своих предшественников – поддержка объектно-ориентированного программирования.

Знакомые с управлением Active Directory знают, что на VBScript или JScript доступ к каталогу Active Directory осуществляется с помощью ADODB, а при использовании .Net Framework – с помощью библиотеки System.DirectoryServices.

Поскольку PowerShell в первую очередь ориентирован на работу с .NET Framework, то администраторам придется узнать основы программирования в Visual Studio в облегченном варианте.

Способы подключения к Active Directory

Для доступа к каталогу Active Directory используется один из двух провайдеров: WinNT или LDAP. Первый из них использовался в доменах Windows NT и в настоящее время используется для совместимости. Основным протоколом доступа к Active Directory является LDAP. В связи с этим провайдер WinNT рассматриваться не будет.

Виртуальный каталог RootDSE

Виртуальный объект RootDSE является точкой входа в любой домен, который содержит информацию о домене в целом, доступных пространствах имен, поддерживаемой версии LDAP и др. Первоначально объект RootDSE был определен в RFC 2251 [1] как часть спецификации LDAP версии 3.

Для подключения к RootDSE используется бессерверное подключение. Чтобы определить имя домена, используется локатор контроллера домена, который находится на каждом контроллере. Доступ к объекту осуществляется анонимно.

Определение имени текущего домена

На практике виртуальный каталог RootDSE используется для определения имени текущего домена, а также косвенным образом для определения принадлежности сервера к тому или иному сайту.

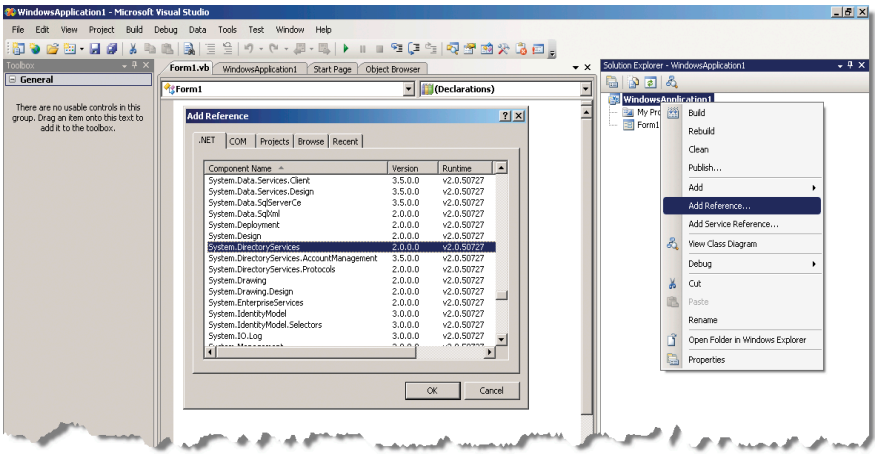
Как известно, существуют три варианта обозначения имени домена:

- **RDN (Relative Distinguished Name)** – относительное уникальное имя, например DC=ISLAND,DC=RU;
- **FQDN (Fully Qualified Domain Name)** – полное доменное имя или DNS-имя, в котором компоненты разделены точкой, например ISLAND.RU;
- **NetBIOS-имя** – обычно первая часть DNS-имени домена, например ISLAND.

На практике для управления объектами Active Directory с помощью провайдера LDAP используется имя домена,

записанное в RDN-формате. Домен имеет древовидную структуру, в которой доступны несколько пространств имен. Каждое из них имеет свою точку входа:

- **defaultNamingContext.** Описываемое этим параметром пространство имен используется для управления учетными записями пользователей, групп, контейнеров и других объектов в оснастке Active Directory Users and Computers.
- **schemaNamingContext.** Данным параметром описывается местоположение схемы домена.
- **configurationNamingContext.** Содержит RDN-путь к разделу, содержащему путь к конфигурации леса текущего домена.
- **rootDomainNamingContext.** Значением параметра является RDN-путь к корню домена (домен, который был создан первым в лесу).



Подключение к проекту библиотеки System.DirectoryServices

Имя текущего домена является значением параметра defaultNamingContext. Поскольку PowerShell представляет собой нечто среднее между Visual Studio и VBScript, то рассмотрим управление объектом RootDSE с помощью трех языков программирования (VB.NET, VBScript и PowerShell), чтобы наглядно показать все преимущества PowerShell.

Удаленное подключение к каталогу Active Directory (провайдер LDAP) обеспечивается с помощью Active Directory Services Interface (ADSI). В VBScript для этого используется функция GetObject(), в качестве аргумента которой фигурирует путь к объекту. В PowerShell для решения идентичной задачи вместо функции GetObject() в квадратных скобках указывается ключевое слово ADSI, а в кавычках, следующих далее, – путь к объекту.

Определение RDN-имени текущего домена с помощью виртуального объекта RootDSE состоит из двух этапов. Первый этап – получение доступа к RootDSE, второй – чтение строкового значения параметра defaultNamingContext. На языке VBScript для этого используется функция GetObject(), аргументом которой является путь к виртуальному объекту: LDAP://RootDSE, и чтение значения с помощью функции GET, аргументом которой в свою очередь является имя параметра (см. **листинг 1**).

Листинг 1. Определение RDN-имени домена (VBScript)

```
Set obj = GetObject("LDAP://rootDSE")
domain = obj.Get("defaultNamingContext")
MsgBox domain
```

Эту же операцию в VB.NET можно осуществить двумя способами:

- с помощью COM-объекта. Так же как и в VBScript, осуществляется вызов функции GetObject() и с помощью метода GET – чтение значения параметра defaultNamingContext;
- с помощью класса DirectoryEntry, входящего в состав библиотеки Directory Services .Net Framework (см. **листинг 2**).

Первый способ рассматриваться не будет, поскольку его скорость работы по сравнению со вторым оставляет желать лучшего.

Для реализации второго способа необходимо к созданному в Visual Studio проекту подключить библиотеку System.DirectoryServices (см. **рисунок**).

Листинг 2. Определение RDN-имени домена (VB.NET)

```
Imports System.DirectoryServices

...
     Dim obj As New DirectoryEntry("LDAP://RootDSE")
     Dim domain As String = ↵
            obj.Properties("defaultNamingContext").Value
     MsgBox (domain)
```

Несмотря на то что PowerShell поддерживает COM-объекты, рекомендуется отдать предпочтение библиотекам .NET Framework, поскольку скорость работы сценария в этом случае увеличится. Для получения имени домена все так же используется библиотека System.DirectoryServices, однако ее нет необходимости подключать, поскольку многие популярные библиотеки сразу доступны после запуска оболочки PowerShell. Листинг сценария очень похож на листинг VB.NET (см. **листинг 3**).

Листинг 3. Определение RDN-имени домена (VB.NET)

```
$obj=[ADSI]"LDAP://RootDSE"
$domain=$obj.defaultNamingContext
$domain
```

Обратите внимание: все имена переменных в PowerShell начинаются с символа доллара ($).

Как видно, листинг по своему объему сопоставим с листингом на VBScript, а по скорости работы – с VB.NET.

Заключение

После того как определено имя текущего домена, вы получаете доступ ко всему каталогу Active Directory. Степень дозволенности определяется правами, выданными системным администратором. В общем случае доступны поиск объектов, на которые есть соответствующие разрешения, и чтение их свойств.

1. RFC2251 – Lightweight Directory Access Protocol (v3) – http://www.faqs.org/rfcs/rfc2251.html.

PowerShell. Поиск объектов в каталоге Active Directory

Получение информации из каталога Active Directory в девяти случаях из десяти сводится к поиску объектов, которые удовлетворяют заданным критериям.

Иван Коробко

В каталоге Active Directory объекты хранятся в иерархической структуре, поэтому для получения доступа к нужному объекту необходимо указать к нему путь в формате RDN. Обычно местоположение объекта неизвестно или оно может измениться, поэтому программный способ поиска объекта/объектов по заданным условиям – самое эффективное средство для определения его местонахождения.

Иерархическая структура каталога Active Directory

Иерархическая структура каталога Active Directory обычно формируется с помощью специфических объектов – контейнеров (Organizational Unit), которые имеют префикс «OU=». Контейнеры, образующие домен, имеют префикс DC (Domain Component). Большинство других объектов кроме корня домена имеют префикс «CN=» (Common Name).

Однако и здесь существует исключение: служебные контейнеры также имеют префикс «CN=».

Тип объекта определяется набором элементов массива objectClass. Поиск объектов осуществляется по заданным критериям. Чаще всего одним из заданных критериев является тип объекта.

В **таблице 1** приведены типы объектов и соответствующие им значения элементов массива objectClass. При составлении запроса необходимо учитывать пересечения элементов массива.

Ярким примером является запрос поиска всех учетных записей компьютеров и пользователей. Оба класса относятся к классу person, однако учетные записи компьютеров дополнительно входят в класс Computer (см. **таблицу 1**), именно поэтому для выборки всех учетных записей необходимо в фильтре это учесть: (&(objectClass=user)(!objectClass=Computer)).

Поиск объектов в Active Directory при помощи библиотек .NET Framework осуществляется с помощью класса DirectorySearcher, относящегося к пространству имён System.DirectoryServices.

Процедура поиска имеет несколько характеристик, основные из которых:

Таблица 1. Типы объекта в Active Directory

Комментарий	Тип объекта	Значение objectClass	Фрагмент поискового запроса
Учетная запись компьютера	Computer	Top Person OrganizationalPerson User Computer	(&(objectClass=Computer)
Контакт, используется в почтовых приложениях	Contact	Top Person OrganizationalPerson Contact	(&(objectClass=Contact)
Группа безопасности	Group	Top Group	(&(objectClass=Group)
Учетная запись пользователя, не совместимая с доменами Windows 2K	InetOrgPerson	Top Person OrganizationalPerson User InetOrgPerson	(&(objectClass=InetOrgPerson)
Папка дерева каталогов Active Directory	OU	Top OrganizationalUnit	(&(objectClass=OrganizationalUnit)
Опубликованный в Active Directory сетевой принтер	Printer	Top Leaf ConnectionPoint PrintQueue	(&(objectClass=PrintQueue)
Опубликованная в Active Directory сетевая папка	Shared Folder	Top Leaf ConnectionPoint Volume	(&(objectClass=Volume)
Учетная запись пользователя, совместимая с доменами Windows NT	User	Top Person OrganizationalPerson User	(&(objectClass=Person) (!(objectClass =Computer)))
Контейнер (папка), создаваемая в каталоге Active Directory по умолчанию	Container	Top Container	(&(objectClass=Container))
Корень домена	RootDSE	Top Domain DomainDNS	(&(objectClass=DomainDNS)

- **Область поиска.** Область, или глубина, поиска определяется с помощью свойства SearchScope.
- **Критерии поиска.** Задаются с помощью свойства filter.
- **Сортировка элементов.** С помощью метода sort и его свойств задают поле и направление сортировки.

Область поиска

Поиск объекта может быть осуществлен в одной из трех областей:

- **Base (obj.SearchScope=0)** – поиск осуществляется по корневому объекту, указанному в первой части запроса. Всегда возвращается либо один объект, либо пустой набор объектов. Эта область поиска используется чаще всего для проверки на существование объекта, указанного в запросе.
- **OneLevel (obj.SearchScope=1)** – поиск осуществляется только по непосредственным дочерним объектам контейнерного объекта, указанного в первой части запроса. Поиск по вложенным объектам более низких уровней не производится. В поиск не попадет и сам контейнерный объект.
- **SubTree (obj.SearchScope=2)** – поиск осуществляется по всем вложенным объектам. В поиск при этом не попадает сам контейнерный объект. Эта область поиска задана по умолчанию.

Таблица 2. Операторы фильтра поиска

Оператор	Значение
=	Эквивалентно
~=	Примерно равно
<=	Меньше или равно
>=	Больше или равно
>	Больше
<	Меньше
&	И
|	ИЛИ
!	НЕ
*	Любое количество любых символов

Критерии поиска

При составлении критерия, или, как его еще называют, фильтра поиска, используют выражения, строящиеся по определенным правилам, а именно (см. **таблицу 2**):

- каждое выражение должно быть заключено в скобки;
- в выражениях допускается использование операторов сравнения: «<», «<=», «=», «>=» и «>';
- допускаются составные выражения, образуемые с помощью префиксных операторов: «&», «|», «!»;
- между оператором и операндами пробелы не допускаются.

Приведу несколько характерных примеров фильтров:

- **(name=A*).** Результатом поиска являются все объекты, имена которых начинаются с буквы «А».
- **&((objectClass=person)(!objectClass=computer)).** Будут возвращены все объекты, принадлежащие к классу person и не принадлежащие к классу computer, то есть все учетные записи пользователей.

Встречаются случаи, в которых необходимо в качестве значений указывать служебные символы, например звездочку, скобку и др.

Чтобы реализовать эту возможность, нужный символ требуется заменить на соответствующее ему кодовое значение, приведенное в **таблице 3**.

Сортировка элементов

Сортировка элементов осуществляется с помощью метода sort, поддерживающего два свойства: PropertyName и Direction.

С помощью свойства PropertyName производится назначение поля, по которому будет осуществляться сортировка, а с помощью Direction – направление.

В случае Sort.Direction=0 осуществляется упорядочивание от «А» до «Я», при Sort.Direction=1 – в обратном порядке.

Пример поиска объектов в Active Directory

Приведу пример поиска всех учетных записей пользователя на букву «А». Дополнительно выполним сортировку выводимых данных от А к Я по полю name. Рассмотрим алгоритм работы сценария.

Сначала необходимо вызвать объект DirectorySearcher и указать точку монтирования к каталогу Active Directory. Затем укажем область поиска – весь каталог Active Directory: SearchScope = subtree.

Таблица 3. Служебные символы, используемые в фильтрах поиска

Символ	Значение
*	\2a
(	\28
)	\29
\	\5c
NUL	\00
/	\2f

Критерий поиска логически складывается из двух элементов: в первом выберем все учетные записи пользователей. Это можно сделать, указав фильтр (&(objectClass=user)(!(objectClass=Computer))).

Вторым критерием из всех найденных учетных записей отберем начинающиеся с буквы «А». Для этого используем фильтр (&(name=A*)). С помощью символа «*» обозначают произвольное содержимое. Объединив оба фильтра в один, получим: (&((objectClass=user)(name=A*))(!(objectClass=Computer))).

После определения всех необходимых атрибутов поиска необходимо запустить процесс поиска с помощью одной из функций: FindAll() или FindOne().Поскольку объектов заведомо больше одного, то в данном случае рекомендуется использовать FindAll().

Чтение элементов осуществляется с помощью цикла For...Next. Массив – коллекция SearchResults, возвращаемая функцией FindAll(), элемент коллекции – SearchResult.

Для чтения свойств найденных объектов необходимо получить доступ к его пространству имен с помощью метода DirectorySearcher и, наконец, с помощью параметризированного свойства (Parameterized property) Properties() получить требуемое значение.

В **листинге 1** приведен рассмотренный пример на языке VB.NET, а **в листинге 2** – на PowerShell.

Листинг 1. Поиск объектов по заданным критериям (VB.NET)

```
Dim obj As New DirectorySearcher("LDAP://RootDSE")

obj.SearchScope = SearchScope.Subtree
obj.Filter = "(&(objectClass=person)(name=a*) ↵
      (!objectClass=computer))"

obj.Sort.PropertyName = "cn"
obj.Sort.Direction = SortDirection.Ascending
      For Each element As SearchResult In obj.FindAll()
       Dim obj2 As DirectoryEntry = ↵
               element.GetDirectoryEntry()
               msgbox obj2.Properties("cn").Value)
      Next
```

В **листинге 2** код отличается. Значительные изменения в сторону упрощения претерпевает вывод значений элементов полученного массива. В отличие от классического For.... Next, в PowerShell можно использовать инструкцию foreach-object или короткий псевдним % (знак процента). В этом случае доступ к элементам массива осуществляется с помощью специальной переменной «$_».

Листинг 2. Поиск объектов по заданным критериям (PowerShell)

```
$obj =  new-object DirectoryServices.DirectorySearcher ↵
      ("LDAP://RootDSE")

$obj.SearchScope = "Subtree"
$obj.Filter = "(&(objectClass=person)(name=a*) ↵
      (!objectClass=computer))"

$obj.Sort.PropertyName = "cn"
$obj.Sort.Direction = "Ascending"

$obj.FindAll() | %{
$obj2=$_.GetDirectoryEntry()
$obj2.cn
```

Заключение

Подводя итог написанному, необходимо сказать, что поиск объектов – очень мощное средство. Грамотно составленный запрос, верно определенные области поиска и сортировка позволят с легкостью формировать сложнейшие отчеты, на составление которых могло бы уйти гораздо больше времени.

Множественные уязвимости в Wireshark

Программа: Wireshark версии до 1.0.7.
Опасность: Высокая.
Описание: 1. Уязвимость существует из-за ошибки форматной строки в PROFINET/DCP (PN-DCP) диссекторе при обработке названий станций, содержащих символы форматной строки. Удаленный пользователь может с помощью специально сформированного пакета вызвать отказ в обслуживании или выполнить произвольный код на целевой системе.

2. Уязвимость существует из-за ошибки в LDAP-диссекторе. Удаленный пользователь может с помощью специально сформированного LDAP-пакета аварийно завершить работу приложения. Уязвимость распространяется только на Windows-платформы.

3. Уязвимость существует из-за ошибки в Check Point High-Availability Protocol (CPHAP)-диссекторе. Удаленный пользователь может аварийно завершить работу приложения.

4. Уязвимость существует из-за ошибки при обработке Tektronix .rf5-файлов. Удаленный пользователь может с помощью специально сформированного .rf5-файла аварийно завершить работу приложения.
URL производителя: www.wireshark.org.
Решение: Установите последнюю версию 1.0.7 с сайта производителя.

Переполнение буфера в IBM Access Support ActiveX-компоненте

Программа: IBM Access Support ActiveX 3.20.284.0 и более ранние версии.
Опасность: Высокая.
Описание: Уязвимость существует из-за ошибки проверки границ данных в методе GetXMLValue() в библиотеке IbmEgath.dll. Удаленный пользователь может с помощью специально сформированного веб-сайта передать слишком длинный аргумент уязвимому методу, вызвать переполнение стека и выполнить произвольный код на целевой системе.
URL производителя: www.ibm.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Использование небезопасного метода в SAP GUI KWEdit ActiveX-компоненте

Программа: SAP GUI 6400.1.1.41, возможно, другие версии; SAP GUI 7100.1.1.43, возможно, другие версии.
Опасность: Высокая.
Описание: Уязвимость существует из-за наличия небезопасного метода SaveDocumentAs() в Kweditcontrol.KWEdit.1 ActiveX-компоненте (KWEDIT.DLL). Удаленный пользователь может с помощью специально сформированного веб-сайта получить доступ к важным данным и скомпрометировать целевую систему.
URL производителя: www.sap.com.
Решение: Установите последнюю версию с сайта производителя.

Множественные уязвимости в Kerberos

Программа: Kerberos 5.x.
Опасность: Высокая.
Описание: 1. Уязвимость существует из-за ошибки разыменования нулевого указателя в функции spnego_gss_accept_sec_context() в файле src/lib/gssapi/spnego/spnego_mech.c. Удаленный пользователь может аварийно завершить работу демона с помощью«NegTokenInit-токена, содержащего специально сформированные ContextFlags.

2. Уязвимость существует из-за ошибки в функции get_input_token() в реализации SPNEGO. Удаленный пользователь может вызвать повреждение памяти и аварийно завершить работу демона и также получить доступ к важным данным.

3. Уязвимость существует из-за математической ошибки в функции asn1buf_imbed() в ASN.1-декодере. Удаленный пользователь может аварийно завершить работу kinit или KDC.

4. Уязвимость существует из-за ошибки в функции asn1_decode_generaltime(), которая позволяет освободить неинициализированный указатель через некорректную DER-кодировку. Удаленный пользователь может выполнить произвольный код на целевой системе.
URL производителя: web.mit.edu/kerberos/www.
Решение: Установите исправление с сайта производителя.

Множественные уязвимости в ClamAV

Программа: Clam Antivirus версии до 0.95.1.
Опасность: Высокая.
Описание: 1. Уязвимость существует из-за ошибки при обработке файлов, запакованных с помощью UPack. Удаленный пользователь может аварийно завершить работу приложения.

2. Уязвимость существует из-за ошибки проверки границ данных в функции cli_url_canon() в файле libclamav/phishcheck.c. Удаленный пользователь может с помощью специально сформированных ссылок вызвать переполнение буфера и выполнить произвольный код на целевой системе.
URL производителя: www.clamav.net.
Решение: Установите последнюю версию 0.95.1 с сайта производителя.

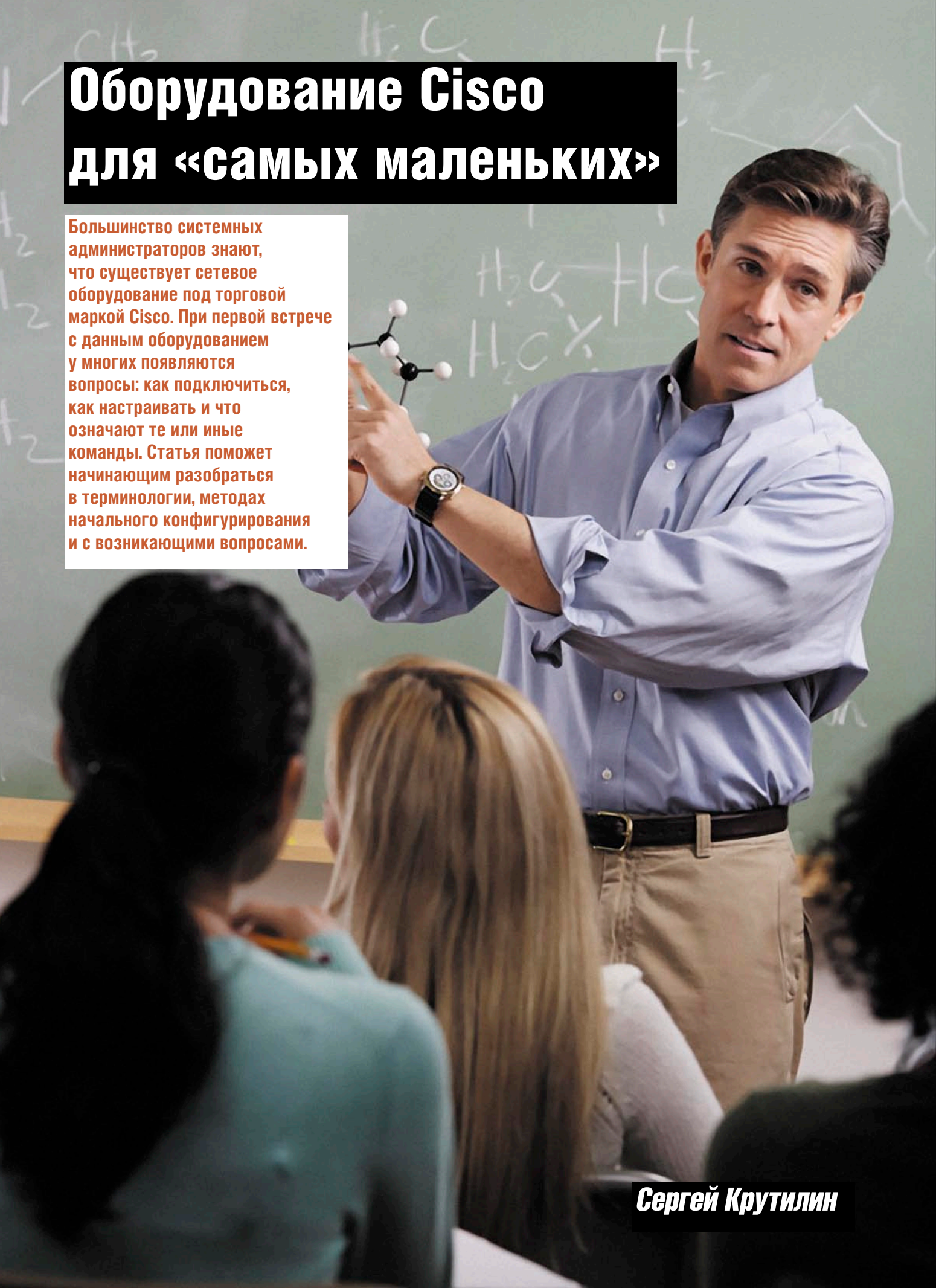
Переполнение буфера в Ghostscript

Программа: Ghostscript 8.64, возможно, другие версии.
Опасность: Высокая.
Описание: Уязвимость существует из-за ошибки проверки границ данных во время декодирования сегментов словаря символов JBIG2 в библиотеке jbig2dec. Удаленный пользователь может с помощью специально сформированного PDF-файла вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www.ghostscript.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Составил Александр Антипов

Оборудование Cisco для «самых маленьких»

Большинство системных администраторов знают, что существует сетевое оборудование под торговой маркой Cisco. При первой встрече с данным оборудованием у многих появляются вопросы: как подключиться, как настраивать и что означают те или иные команды. Статья поможет начинающим разобраться в терминологии, методах начального конфигурирования и с возникающими вопросами.



Сергей Крутилин

Итак, начнём с простейшего. Ваша организация приобрела маршрутизатор Cisco 871 для следующих задач: доступ в Интернет, сервис для автоматического назначения IP-адресов локальным ПК (DHCP), публикация внутреннего почтового сервера.

Вы распаковываете коробку с новым оборудованием и видите перед собой устройство, внешний вид которого представлен на **рис. 1**.

Рисунок 1. Внешний вид устройства

Первый вопрос – как подключиться? Достаточно просто – в комплекте с оборудованием идёт «шнурок» синего цвета (консольный кабель, см. **рис. 2**), его нужно подключить одной стороной (разъём RJ45) к порту с названием Console, а другой стороной к COM-порту вашего ПК.

Для управления оборудованием нам понадобится программа, к примеру, Hyper Terminal в Microsoft Windows («Главное меню → Программы → Стандартные → Связь»). Запускаем Hyper Terminal, вводим название подключения (к примеру, Cisco), затем нажимаем OK, на втором окне выбираем требуемый COM-порт, затем настраиваем Hyper Terminal, как показано на **рис. 3**, и нажимаем OK. Теперь включаем маршрутизатор. На экране появится окно загрузки с примерно следующим содержанием:

```
System Bootstrap, Version 12.2(11r)YV3, RELEASE SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2004 by cisco Systems, Inc.
C800/SOHO series (Board ID: 29-129) platform with 65536 Kbytes of main memory
program load complete, entry point: 0x80013000, size: 0x753404
Self decompressing the image : ########################################
################################################################################
############### [OK]
```

После распаковки образа Cisco IOS (Internetwork Operating System) и загрузки всех компонентов вы оказываетесь в пользовательском режиме, который обозначается знаком «>» (Router>). Помимо подключения по консоли, возможно также удалённое подключение по telnet или ssh. Настройка подключения по telnet будет рассмотрена чуть позже. Пользовательский режим предназначен для отладочных целей, таких как ping, подключения к другим маршрутизаторам и прочее. При вводе вопросительного знака вы получаете весь список команд. В дальнейшем вы часто будете использовать систему справки. Рассмотрим пример.

```
Router>show ?

aaa                 Show AAA values
aal2                Show commands for AAL2
appfw               Application Firewall information
auto                Show Automation Template
backup              Backup status
bgp                 BGP information
```

```
Router>show ip ?

accounting          The active IP accounting database
admission           Network Admission Control information
aliases             IP alias table
arp                 IP ARP table
as-path-access-list List AS path access lists
auth-proxy          Authentication Proxy information
```

Помимо пользовательского режима, существует привилегированный режим. Для доступа к нему необходимо ввести команду enable и пароль (стандартные имя пользователя и пароль обычно Cisco, список стандартных паролей для различного оборудования смотрите по адресу http://www.phenoelit-us.org/dpl/dpl.html).

В данном режиме выполняются основные действия конфигурирования, такие как настройка сетевых интерфейсов, правила брандмауэра и т. д.

Для просмотра текущей конфигурации маршрутизатора выполните команду «show config» (или сокращённо «sh conf»). Просматривать конфигурацию можно при помощи клавиши пробела.

Для настройки основных параметров необходимо использовать команду «configure terminal» (или сокращённо «conf t»).

Рассмотрим пример настройки.

Пункт 1. Сетевые интерфейсы

В привилегированном режиме просматриваем доступные интерфейсы (не забывайте поставить пробел после знака «|»):

```
Router#sh conf | include interface
```

Получаем примерно следующие результаты:

```
interface Ethernet0
interface Ethernet1
interface FastEthernet1
interface FastEthernet2
```

где:

- **Ethernet** – это физические интерфейсы, которым можно присваивать IP-адреса;
- **FastEthernet** – это так называемые линки (ссылки) на внутренний интерфейс, IP-адреса данным линкам присваивать нельзя.

Заходим в режим конфигурирования и настраиваем интерфейсы:

```
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
```

Выбираем интерфейс Ethernet 0 («interface Ethernet 0»):

```
Router(config)#int et 0
```

Присваиваем интерфейсу IP-адрес и маску подсети:

```
Router(config-if)#ip address 192.168.5.3 255.255.255.0
```

Нажимаем <CTRL> + <Z> для выхода из режима конфигурирования:

```
Router(config-if)#^Z
Router#
4d01h: %SYS-5-CONFIG_I: Configured from console by console
```

Сохраняем конфигурацию («write memory»):

```
Router#wr mem
```

Конфигурация маршрутизатора хранится в 2 файлах:

- **running-config** – текущая конфигурация маршрутизатора;
- **startup-config** – конфигурация, применяемая при загрузке.

После внесения изменений в конфигурацию вы также можете выполнить команду «copy run start» («copy running-config startup-config»). Также будет полезным сохранять конфигурацию на внешний TFTP-сервер (Trivial File Transfer Protocol) (к примеру, этот – http://tftpd32.jounin.net). Выполним следующие команды.

Выбираем копируемую конфигурацию:

```
Router#copy tftp: startup-config
```

или

```
Router#copy tftp: running-config
```

Вводим IP-адрес TFTP-сервера:

```
Address or name of remote host []?192.168.5.1
```

Задаём имя сохраняемого файла:

```
Source filename []?cisco871-strat-config
```

Пункт 2. Смена стандартного пароля и настройка telnet

По умолчанию telnet отключён на маршрутизаторах Cisco. В первом пункте мы настроили внутренний интерфейс на использование IP-адреса 192.168.5.3 и при попытке подключения посредством telnet увидим следующее сообщение:

```
C:\>telnet 192.168.5.3
```

```
Trying 192.168.5.3 ... Open
Password required, but none set
[Connection to 192.168.5.3 closed by foreign hos
```

Заходим в режим конфигурирования и выполняем следующие команды:

```
Router#conf t
Router(config)#line vty 0 4
Router(config-line)#login
```

```
% Login disabled on line 66, until 'password' is set
% Login disabled on line 67, until 'password' is set
% Login disabled on line 68, until 'password' is set
% Login disabled on line 69, until 'password' is set
% Login disabled on line 70, until 'password' is set
```

Устанавливаем пароль samag:

```
Router(config-line)#password samag
```

Пробуем подключиться к маршрутизатору:

```
C:\>telnet 192.168.5.3
```

```
User Access Verification
Password:samag
Router>
```

Устанавливаем пароль samagsecret для доступа к привилегированному режиму (enable):

```
Router#conf t
Router(config)#enable secret samagsecret
```

Рисунок 2. Консольный кабель

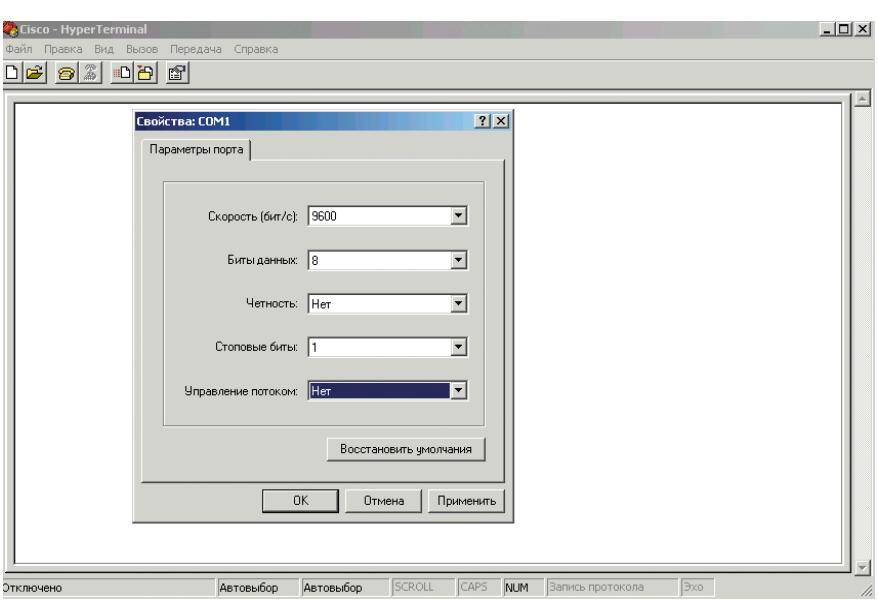


Рисунок 3. Настройка Hyper Terminal

Создаём пользователя user с паролем samaguser:

```
Router(config)#username user secret samaguser
Router(config)#^Z
Router#wr mem
```

Пробуем подключиться к маршрутизатору:

```
C:\>telnet 192.168.5.3

User Access Verification
Username: user
Password:samaguser
Router>enable
Password:samagsecret
```

Пункт 3. Настраиваем доступ в Интернет для внутренних пользователей при помощи NAT (Network Address Translation – трансляция сетевых адресов)

Создаём пул с публичными IP-адресами или адресом (на эти адреса будут отвечать внешние устройства сети – отправленные пакеты данных должны возвращаться к источнику отправки).

Предположим, что провайдер выделил вашей организации внешний пул IP-адресов: 1.2.3.4-1.2.3.6 с маской 255.255.255.248, тогда пул с названием Internet будет выглядеть так:

Используем один внешний адрес:

```
Router#conf t
Router(config)#ip nat pool Internet 1.2.3.4 1.2.3.4 ↵
      netmask 255.255.255.248
```

Либо используем несколько внешних адресов:

```
Router#conf t
Router(config)#ip nat pool Internet 1.2.3.4 1.2.3.6 ↵
      netmask 255.255.255.248
```

Теперь создаём список доступа (Access Control List, ACL) с перечислением адресов подсетей, которым разрешён выход в Интернет. Более подробно списки доступа рассмотрены ниже.

```
Router(config)#ip access-list 10 permit 192.168.5.0 ↵
      0.0.0.255
```

Настраиваем правила трансляции для IP-адресов внутренней подсети, используя ACL:

```
Router(config)#ip nat insi\de source list 10 pool ↵
      Internet overload
```

Определяем, на каких интерфейсах будет входящий/исходящий NAT:

```
Router(config)#int et 0
Router(config-if)#in nat inside
Router(config-if)#int et 1
Router(config-if)#in nat outside
```

Настраиваем шлюз по умолчанию:

```
Router(config)#ip route 0.0.0.0 0.0.0.0 et 1 ↵
      (ip route 0.0.0.0 0.0.0.0 Ethernet 1)
```

или

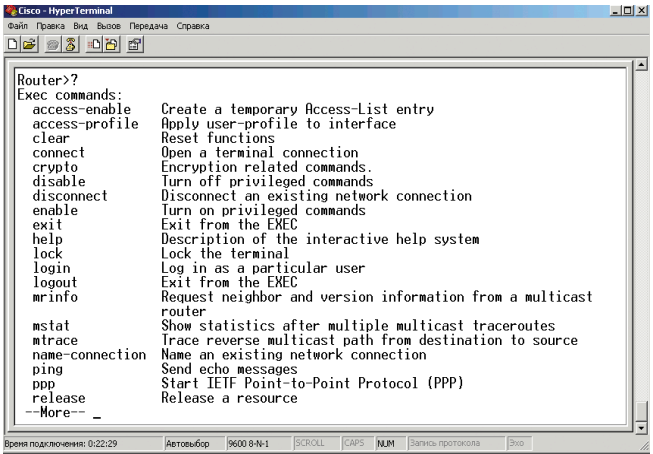


Рисунок 4. Вывод справки

```
Router(config)#ip route 0.0.0.0  0.0.0.0 ↵
      внешний_IP-адрес_шлюза_провайдера ↵
      (ip route 0.0.0.0  0.0.0.0 a.b.c.d)
```

Далее как обычно:

```
Router(config-if)#^Z
Router#wr mem
```

Пункт 4. ACL – списки доступа

- **<1-99>** – стандартные списки доступа;
- **<100-199>** – расширенные списки доступа;
- **<1100-1199>** – расширенные 48-битные списки доступа по MAC-адресам;
- **<1300-1999>** – стандартные списки доступа (дополнительные номера);
- **<200-299>** – списки доступа, основанные на типах протоколов;
- **<2000-2699>** – расширенные списки доступа (дополнительные номера);
- **<700-799>** – 48-битные списки доступа по MAC-адресам;
- **dynamic-extended** – расширенные ACL, зависящие от времени;
- **rate-limit** – специфические ACL, построенные на оценке лимитов.

Рассмотрим наиболее часто используемые.

Стандартный список доступа (присваиваем 10-й номер списку и разрешаем доступ пятой подсети):

```
Router(config)#ip access-list 10 permit 192.168.5.0 ↵
      0.0.0.255
```

Создаём список доступа с определением протокола, в частности, TCP (рекомендуется создавать «зеркальные» списки доступа). Разрешаем обмен трафиком между хостами a.b.c.d и w.x.y.z:

```
Router(config)#ip access-list  101 permit tcp ↵
      host a.b.c.d host w.x.y.z
Router(config)#ip access-list  101 permit tcp ↵
      host w.x.y.z host a.b.c.d
```

Примечание: нельзя удалять в режиме конфигурирования одну из строчек ACL при выполнении команды.

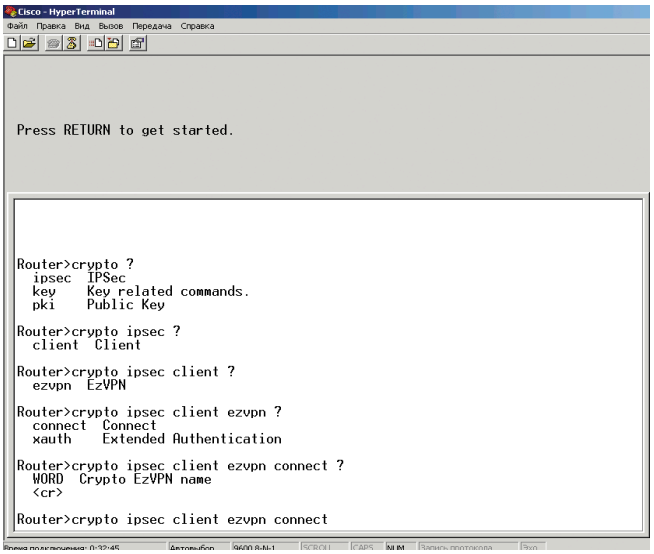


Рисунок 5. Пример использования системы справки

Удалим список доступа под номером 101:

```
Router(config)# no ip access-list 101 permit tcp ↵
      host a.b.c.d host w.x.y.z
```

Создаём расширенный список доступа с именем SamagACL:

```
Router(config)# ip access-list extended SamagACL
```

Разрешаем прохождение TCP-трафика между хостами a.b.c.d и w.x.y.z:

```
Router(config-ext-nacl)# permit tcp host a.b.c.d ↵
      host w.x.y.z
```

Для удаления одного из правил в списке доступа выполняем команду:

```
Router(config-ext-nacl)# no permit tcp host a.b.c.d ↵
      host w.x.y.z
```

Список доступа по портам. Разрешаем прохождение входящего/исходящего трафика на 80 TCP-порт:

```
Router(config)#access-list 111 permit tcp any any eq 80
```

где eq – определение номера порта.

Также можно вместо цифр вводить имена для распространенных портов, к примеру:

```
Router(config)#access-list 111 permit tcp any any eq WWW
```

Полный список доступных имён смотрите в справке данной команды:

```
Router(config)#access-list 111 permit tcp any any eq ?
```

Пункт 5. Перенаправляем входящий трафик на внутренние серверы

К примеру, мы хотим опубликовать 25 TCP-порт (электронная почта). Делается это следующей командой:

```
Router(config)#ip nat inside source static ↵
      tcp 192.168.5.2 25 a.b.c.d 25 extendable
```

где:

- **192.168.5.2** – адрес почтового сервера;
- **a.b.c.d** – внешний IP-адрес нашего маршрутизатора.

Пункт 6. Реализуем поддержку раздачи динамических IP-адресов для локальной подсети (DHCP)

Создаём пул DHCP-сервера:

```
Router(config)# ip dhcp pool  Office
```

где Office – имя создаваемого пула.

Определяем, какую подсеть будет обслуживать DHCP-сервер:

```
Router(config-dhcp)# network 192.168.5.0 255.255.255.0
```

Задаём имя домена для клиентов DHCP:

```
Router(config-dhcp)# domain-name samag.ru
```

Назначаем шлюз по умолчанию:

```
Router(config-dhcp)#default-router 192.168.5.3
```

Примечание: адрес шлюза должен быть из подсети клиентов, в нашем случае это подсеть 192.168.5.0.

Настраиваем список DNS-серверов для выдачи клиентам локальной подсети:

```
Router(config-dhcp)# dns-server 192.168.5.2
```

где 192.168.5.2 – адрес внутреннего DNS-сервера.

Устанавливаем срок аренды выданных IP-адресов на 7 дней:

```
Router(config-dhcp)#lease 7
```

Исключаем адреса из пула для предотвращения конфликтов (серверам присваиваем статические IP-адреса):

```
Router(config)# ip dhcp excluded-address 192.168.5.3 ↵
      192.168.5.4
Router(config)#^Z
Router#wr mem
```

Заключение

Теперь наш маршрутизатор способен предоставлять ПК внутренней подсети доступ в Интернет, сервис DHCP, почтовый сервер способен принимать почту с внешних адресов на 25-й порт.

1. http://cisco.com/en/US/support/index.html.
2. http://www.faq-cisco.ru.
3. http://www.ciscolab.ru.
4. http://ru.wikipedia.org/wiki/Cisco.

Уязвимости в Microsoft ISA Server и Forefront Threat Management Gateway

Программа: Microsoft Forefront Threat Management Gateway Medium Business Edition; Microsoft ISA Server 2004; Microsoft ISA Server 2006.

Опасность: Средняя.

Описание: 1. Уязвимость существует из-за ошибки при обработке состояния TCP-сессии в механизме МСЭ для веб-прокси и Web publishing listener. Удаленный пользователь может с помощью специально сформированного TCP-пакета заставить Web listener не принимать новые запросы.

2. Уязвимость существует из-за недостаточной обработки входных данных в аутентификационном компоненте HTML-форм (cookieauth.dll) в ISA Server и Forefront TMG. Удаленный пользователь может с помощью специально сформированного запроса выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта. Для успешной эксплуатации уязвимости должен быть включен Web publishing, и аутентификация через HTML-формы должна быть включена на Web listener, который используется по умолчанию. Уязвимость не распространяется на Microsoft ISA Server 2004.

URL производителя: www.microsoft.com.

Решение: Установите исправление с сайта производителя.

Множественные уязвимости в IBM WebSphere Application Server

Программа: IBM WebSphere Application Server версии до 7.0.0.3.

Опасность: Средняя.

Описание: 1. Уязвимость существует из-за недостаточной обработки URL в /ibm/console/. Удаленный пользователь может выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.

2. Уязвимость существует из-за того, что временные исправления, которые переписывают существующие или создают новые файлы, устанавливают для них привилегии 777.

3. Уязвимость существует из-за неизвестной ошибки, которая может привести к тому, что процесс JAX-RPC WS-Security некорректно проверит подлинность UsernameToken.

4. Уязвимость существует из-за неизвестной ошибки в спецификации цифровых XML-подписей. Подробности уязвимости не сообщаются.

URL производителя: www-01.ibm.com/software/webservers/appserv/was.

Решение: Установите последнюю версию 7.0.0.3 с сайта производителя.

Переполнение буфера в FreeBSD

Программа: FreeBS 7.0, 7.1.

Опасность: Низкая.

Описание: Уязвимость существует из-за ошибки проверки границ данных в ktimer. Локальный пользователь может перезаписать произвольные участки памяти и выполнить произвольный код на целевой системе с привилегиями учетной записи root.

URL производителя: www.freebsd.org.

Решение: Установите исправление с сайта производителя.

Множественные уязвимости в Cisco IOS

Программа: Cisco IOS 12.x.

Опасность: Средняя.

Описание: 1. Уязвимость существует из-за ошибки в функционале Cisco Tunneling Control Protocol (cTCP). Удаленный пользователь может с помощью специально сформированных TCP-пакетов аварийно завершить работу cTCP-сервера. Уязвимости подвержены версии 12.4(9)T и выше с включенной инкапсуляцией Cisco Tunneling Control Protocol (cTCP) для EZVPN-сервера.

2. Уязвимость существует из-за неизвестной ошибки в функционалах Cisco IOS WebVPN и Cisco IOS SSLVPN (SSLVPN). Удаленный пользователь может с помощью специально сформированных HTTPS-пакетов аварийно завершить работу устройства.

3. Уязвимость существует из-за утечки памяти в функционалах Cisco IOS WebVPN и Cisco IOS SSLVPN (SSLVPN). Удаленный пользователь может через SSLVPN-сессию потребить всю доступную память.

4. Уязвимость существует из-за неизвестной ошибки в Cisco IOS при включенном Mobile IP NAT Traversal или Mobile IPv6. Удаленный пользователь может с помощью специально сформированных пакетов прекратить обработку трафика устройством.

5. Уязвимость существует из-за неизвестной ошибки в Cisco IOS SCP-сервере. Злоумышленник может записать произвольные файлы на устройство независимо от настройки CLI. Для успешной эксплуатации уязвимости требуются действительные учетные данные и доступ к CLI и SCP-сервер должен использовать ролевые списки контроля доступа.

6. Уязвимость существует из-за неизвестной ошибки при обработке TCP-пакетов. Удаленный пользователь может с помощью специально сформированной последовательности TCP-пакетов вызвать перезагрузку устройства. Для успешной эксплуатации уязвимости требуется полное трехкратное рукопожатие TCP для соответствующего TCP-порта.

7. Уязвимость существует из-за неизвестной ошибки в реализации Session Initiation Protocol (SIP). Удаленный пользователь может с помощью специально сформированного SIP-сообщения вызвать перезагрузку устройства.

8. Уязвимость существует из-за ошибки при обработке UDP-пакетов. Удаленный пользователь может с помощью специально сформированного UDP-пакета заблокировать доступ к интерфейсу устройства.

9. Уязвимость существует из-за неизвестной ошибки при обработке IP-сокетов. Удаленный пользователь может запретить устройству принимать новые подключения или сессии, потребить большое количество памяти и процессорного времени или вызвать перезагрузку устройства. Для успешной эксплуатации уязвимости требуется полное трехкратное рукопожатие TCP для соответствующего TCP-порта.

URL производителя: www.cisco.com.

Решение: Установите исправление с сайта производителя.

Составил Александр Антипов

Почтовый клиент Alpine

Игорь Штомпель

Производительность компьютеров стремительно увеличивается, требования программного обеспечения к системным ресурсам растут. Как быть в ситуации, когда нет возможности модернизировать аппаратную составляющую компьютерной системы, а оптимизация программного обеспечения стала насущной необходимостью? Одним из выходов из данного положения может стать внедрение Alpine.

В конце 2005 года началась разработка почтового клиента Alpine на базе проекта Pine. Последний был создан в недрах Вашингтонского университета и выпущен под несвободной лицензией [1]. Товарный знак – Pine на данном этапе стал мешать группе Pine Team в ее желании реорганизовать исходный код и распространять программу. Обязательства, связанные с товарным знаком, и новые устремления разработчиков привели к тому, что был запущен проект – Alpine [2]. Свое детище разработчики выпустили под лицензией Apache License, Version 2.0, сделав возможной, в отличие от Pine, редистрибуцию модифицированных версий программы.

Мы рассмотрим установку, настройку, отправку и получение почты на примере дистрибутива Debian GNU/Linux 4.0. В последнем имеется два «мощных» почтовых клиента для работы в консоли – Mutt и Alpine. По своей функциональности оба клиента схожи, но, как представляется на наш субъективный взгляд, Alpine выделяется более удобным интерфейсом. Скачиваем пакет alpine_2.00_i386.deb с этой страницы – http://www.washington.edu/alpine/acquire. Alpine, в репозиториях Debian Etch 4.0, находится в Backports в виде пакета версии 1.10, которая перекочевала в основной репозиторий Debian Lenny 5.0. Переходим в консоль, получаем права администратора и выполняем команду:

```
dpkg -i alpine_2.00_i386.deb
```

Впоследствии, если появится необходимость, программу можно будет удалить следующим способом:

```
dpkg -r alpine
```

Итак, программа установлена. Тот, кто вынужден использовать устаревшее оборудование или предпочитает работать в консоли, получил всю необходимую функциональность для работы с электронной почтой. Теперь программу необходимо настроить для работы с почтовыми серверами. Запускаем наш новый почтовый клиент:

```
alpine
```

При работе с программой можно использовать как клавиши управления курсора + <Enter>, так и зарезервированные клавиши, назначение которых всегда отображается в нижней части экрана программы. После старта Alpine мы увидим следующее (см. **рис. 1**).

Для получения справки необходимо нажать клавишу <?> (HELP), для создания и отправки сообщения <C> (COMPOSE MESSAGE), для просмотра сообщений в текущей активной папке (папка, которая открывалась последней) клавишу <I> (MESSAGE INDEX), для просмотра списка папок <L> (FOLDER LIST), для просмотра адресной книги <A> (ADDRESS BOOK), для выхода из программы нажимаем <Q> (QUIT).

Для перехода в режим конфигурации программы, как видно на **рис. 1**, надо нажать клавишу <S>. В данном режиме, изображенном на **рис. 2**, пользователь получает широкие возможности для настройки программы:

- **<E> (Exit Setup)** – мы вернемся в главное меню Alpine, которое рассмотрели выше.
- **<P> (Printer)** – переход в режим настройки печати.
- **<N> (Newpassword)** – установка и смена пароля.
- **<C> (Config)** – большинство общих настроек Alpine.
- **<S> (Signature)** – создание и редактирование сигнатур, которые будут добавляться в каждое новое создаваемое и отправляемое письмо.
- **<A> (AddressBooks)** – настройка и редактирование адресных книг.
- **<L> (collectionLists)** – здесь вы можете группировать ваши папки для лучшей организации электронной почты.
- **<R> (Rules)** – назначение различных правил на базе шести подкатегорий: для отображения элементов MESSAGE INDEX цветом, фильтрации сообщений, назначения ролей (например, разные подписи для различных адресатов) и другого.
- **<D> (Directory)** – настройка Alpine для использования с сервером каталогов LDAP.
- **<K> (Kolor)** – установка пользовательских цветов для различных частей почтового клиента.
- **<M> (S/MIME)** – настройка использования S/MIME для проверки подписанных сообщений, расшифровки сообщений, а также для подписи или шифрования исходящих сообщений.
- **<Z> (RemoteConfigSetup)** – настройка удаленной конфигурации, например, для использования с сервером, работающим по протоколу IMAP.
- **<X> (eXceptions)** – настройка команды-переключателя, которая меняет поведение других команд.

Нажимаем <C> и переходим в режим настройки своей учетной записи. Заполняем поле Personal Name – ваше имя, отображаемое в поле «От» перед почтовым адресом писем, которые вы отправляете.

В User Domain указываем имя домена почтового сервера, то, что в вашем почтовом адресе следует за именем пользователя. Например, если почтовый адрес выглядит так – имя@gmail.com, то доменом почтового сервера будет – gmail.com.

В поле SMTP server (for sending) прописываем имя почтового сервера для отправки электронных писем. Например, для почтового ящика на сервере gmail.com – smtp.gmail.com.

В Inbox Path указываем – pop.gmail.com/user=имя@gmail.com/pop3/ssl.

Для использования imap-сервера вместо /pop3/ssl необходимо прописать /imap/ssl и, конечно, pop.gmail.com изменить на imap.gmail.com.

Для задания Alpine пути к локальному архиву входящей почты используется поле Incoming Archive Folders. Укажем в нем, например: /home/имя/mail, что означает – почта будет сохраняться в домашнем каталоге пользователя в директории mail. Последнюю необходимо создать перед началом настройки почтовой программы. Все настройки завершены. Нажимаем клавишу <E> и попадаем в главное меню.

Подключимся к почтовому серверу. Для этого переходим к списку папок <L>. Нажимаем <Enter> на Incoming-Folders, а затем на INBOX, отвечаем «Да» (Y) на вопрос «Re-open folder to check for new message?» (открыть входящие с проверкой получения новых сообщений?). Вы увидите список сообщений из папки «Входящие» на почтовом сервере.

Для сохранения письма в локальную папку saved-messages необходимо нажать «S». Чтобы вернуться к предыдущему экрану, надо нажать «<» (соответственно «>» – вперед). Переходим в Mail, где находятся локальные папки: sent-mail (отправленные сообщения), saved-messages (сохраненные сообщения).

Подготовим и отправим письмо. Находясь в главном меню, переходим в COMPOSE MESSAGE или нажимаем клавишу <C>. Заполняем поля To (электронный адрес получателя письма), Cc (адреса, на которые отправить копии письма), Attchmnt (приложение), Subject (тема письма). Заполняем тело сообщения после Message Text.

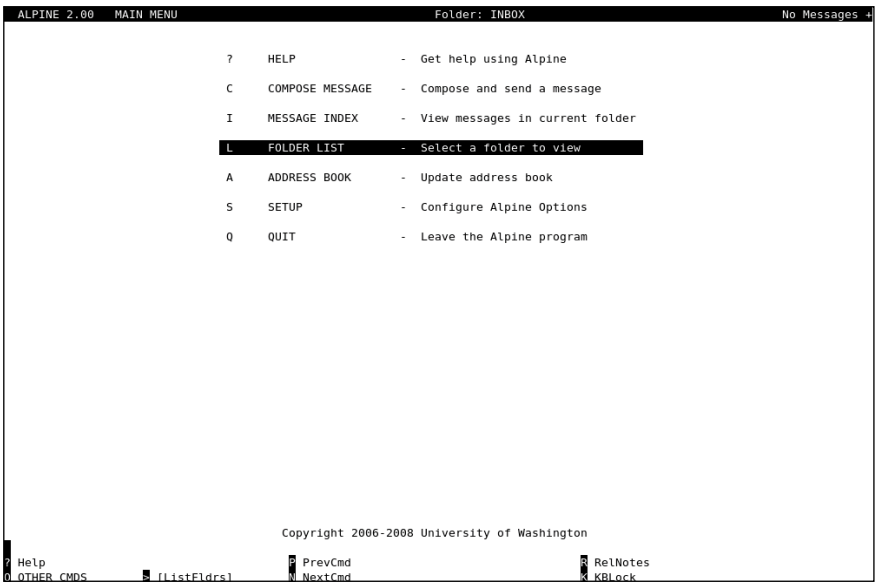


Рисунок 1. Старт программы Alpine

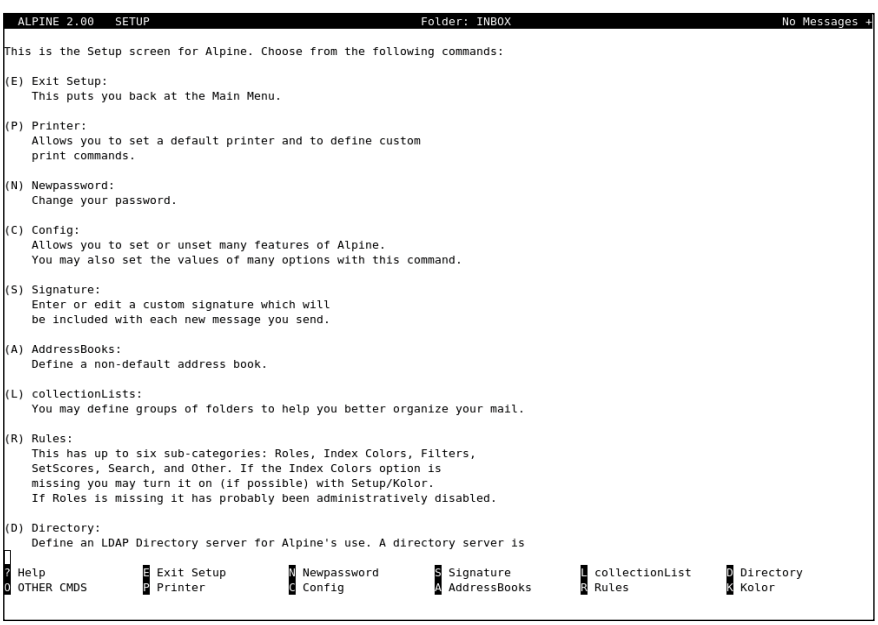


Рисунок 2. Окно конфигурирования Alpine

Если требуется прикрепить файл к письму, то перемещаем курсор на любое поле заголовка письма, например, To, нажимаем «^J» (<Ctrl> + <J>) и прописываем путь к файлу. Например, если файл arch.zip хранится в корне домашнего каталога пользователя, то прописываем: /home/arch.zip.

Для отправки письма нажимаем «^X» (<Ctrl> + <X>), вводим имя пользователя и пароль, и все – сообщение отправлено.

Несколько слов в заключение. Alpine – это свободный и гибкий инструмент для работы с электронной почтой, процесс настройки и работы с которым познавательный и интересный. В данной статье мы затронули только «верхушку айсберга». Читателям будет интересно узнать, что программу использует создатель ядра операционной системы GNU/Linux – Линус Торвальдс [3]. Приглашаем вас в мир Alpine!

1. http://www.washington.edu/pine/overview/legal.html.
2. http://www.washington.edu/alpine/overview/story.html.
3. http://www.lifehacker.com.au/tips/2008/01/31/lifehacker_australia_interview.html.

Выбираем антивирус для небольшой сети

Сергей Яремчук

Сегодня на рынке предлагается большое количество антивирусных пакетов, ориентированных на применение в организациях разного размера, и, несмотря на некоторое сходство в архитектуре, все они весьма отличаются функционально, не говоря уже о стоимости. В итоге выбрать наиболее подходящее не так уже и просто.

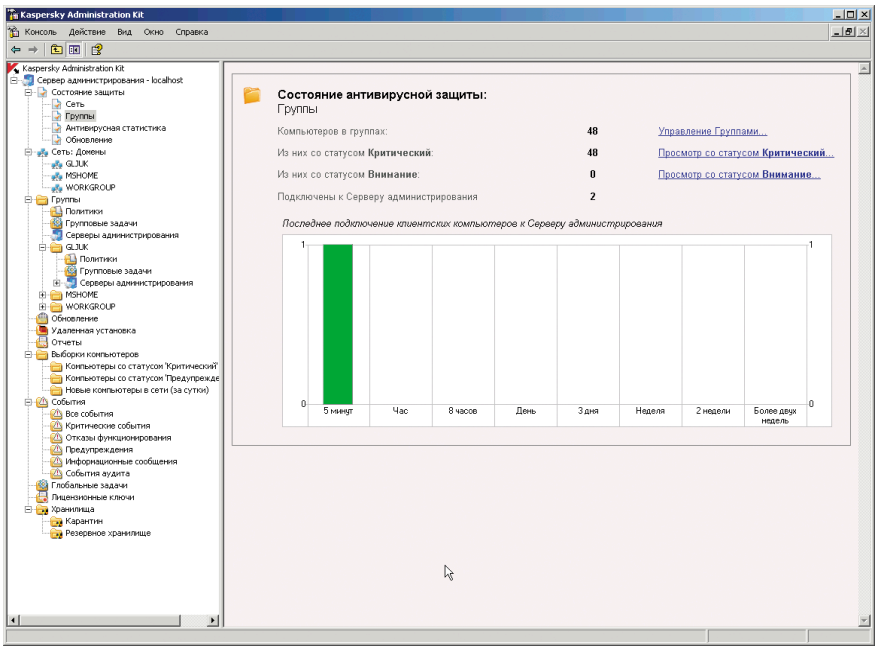
Учитывая, что в каталогах одного поставщика имеется несколько комплексных решений, рассчитанных на сети разного размера, чтобы не перебирать все, определимся с параметрами будущей сети. Пусть это будет небольшая сеть на 50 рабочих мест, имеющая файловый и почтовый серверы, выход в Интернет защищает шлюз. Использование «обычных» персональных антивирусов для защиты такого количества систем крайне неудобно, ведь все операции, начиная с установки, настройки и контроля за обновлениями, придется выполнять вручную. В итоге все старания могут быть сведены на нет, например, если пользователь отключил монитор или базы по разным причинам вовремя не обновились. Хотя сегодня в небольших организациях еще нередко можно встретить системного администратора, бегающего с флешкой, обновляющего таким образом антивирусные базы.

Специализированные решения, построенные по клиент-серверной схеме, более предпочтительны, так как они обеспечивают централизованное управление, упрощенную процедуру развертывания, контроль за работой агентов и выполнением всех предусмотренных заданий и политик, установку обновлений с единой локальной базы, экономя интернет-трафик, создание отчетов и так далее.

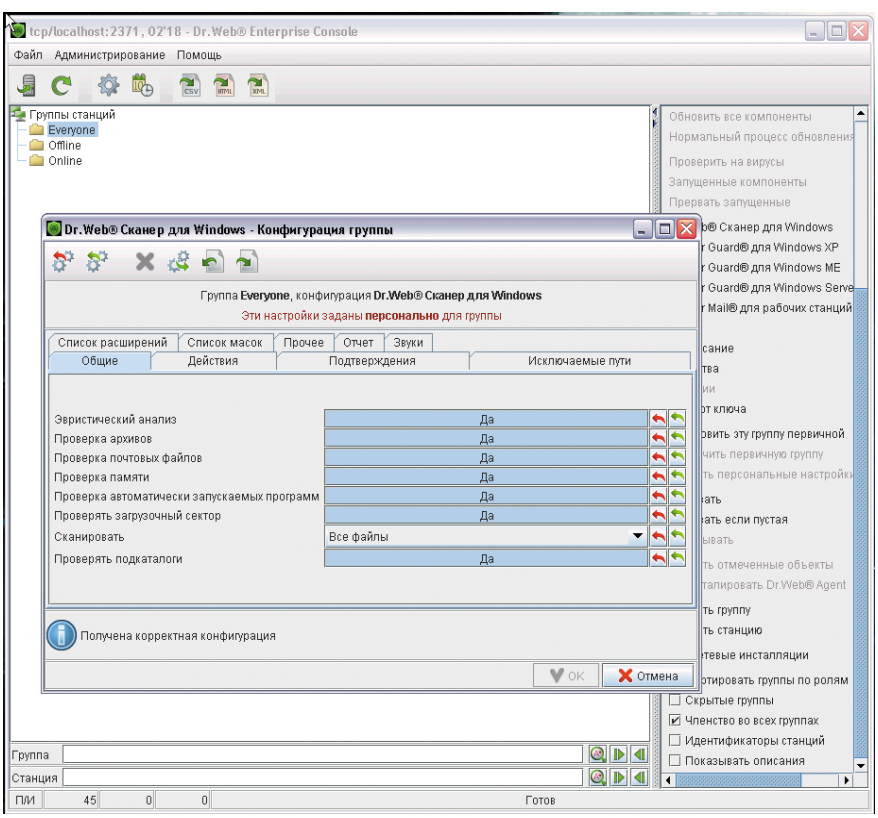
В статье вы не найдете тестов антивирусных движков и информации о количестве записей в базе. Основное внимание уделено особенностям реа-

лизации антивирусных продуктов 7 популярных разработчиков, подходящих для решения нашей задачи, – компонентам, возможностям клиентского модуля и централизованного управления. Ориентировочная цена приведена в **таблице 1**. Окончательную цену составить тяжело, так как на её формирование может повлиять срок лицензии (на два года – дешевле), количество одновременно покупаемых продуктов или лицензий на одно решение, скидки и акции магазинов и так далее. Чтобы легче было определиться и увидеть отличия, основные параметры сведены в **таблицу 2**.

Сразу отвечу на два вопроса, которые, скорее всего, возникнут у читателя. Да, вполне возможно установить шлюз, файловый и почтовый сервер на *BSD или GNU/Linux, для защиты которого использовать свободный антивирус ClamAV. Таким образом несколько уменьшить итоговую стоимость, хотя и за счет некоторой децентрализации управления. Такие решения неоднократно рассматривались на страницах журнала, поэтому если есть соответствующий опыт, можно самостоятельно собрать нужную систему из «кирпичиков». Также часто рекомендуют на компонентах сети сервер/шлюз/ПК использовать антивирусы разных разработчиков. Некоторый смысл в этом есть, так как всегда существует вероятность, что во время очередной эпидемии одна из компаний среагирует чуть быстрее. Да и мне неоднократно попадались вирусы, в том числе и старые, которые определялись далеко не всеми движками. Но суммарная стоимость будет выше, а вот говорить о том, что применение двух антивирусов вдвое увеличит защиту, не приходится. Поскольку, точно следуя логике, «одна среагирует быстрее», нужно согласиться с тем, что непременно другая среагирует медленнее.



Консоль управления Kaspersky Administration Kit



Консоль администратора Dr.Web Enterprise Suite

«Лаборатория Касперского»

«Лаборатория Касперского» для защиты корпоративной сети любого масштаба и сложности предлагает линейку продуктов, объединенных общим названием Kaspersky Open Space Security. KOSS состоит из четырех решений, ориентированных на разный уровень организаций и защищающих рабочие станции и смартфоны, файловые, почтовые серверы и шлюзы. Кроме этого предлагаются и продукты для защиты отдельных узлов сети. Для целостной защиты сети любого масштаба предложен Kaspersky Total Space Security, который может состоять из 17 компонентов (полный список на http://www.kaspersky.ru/total_space_security) и имеет все необходимое для решения такой задачи.

Непосредственно на рабочих станциях и серверах, подлежащих защите, устанавливается специализированная версия антивируса. Здесь следует отметить весьма большой список систем и решений: рабочие станции (Windows, Linux), мобильные системы (Windows Mobile, Symbian), файловые

серверы (Windows, Linux, NetWare), совместной работы (Exchange 2003/2007, Lotus Notes/Domino), почтовые серверы (Sendmail, Qmail, Postfix, Exim) и шлюзы (Microsoft ISA Server, Check Point FireWall) и другие.

Для централизованного управления используется Kaspersky Administration Kit, фактически состоящий из трех составляющих, которые необязательно должны быть установлены на одном компьютере:

- оснастки MMC – выполнение настроек и получение информации о состоянии защиты;
- агент администрирования – обеспечивает взаимодействие между сервером и установленным на компьютере антивирусным приложением;
- сервер администрирования – непосредственное управление работой агентов и лицензиями, хранение настроек, сбор информации.

Интерфейс консоли локализован и достаточно прост в освоении. Предусмотрена одновременная работа нескольких связанных друг с другом серверов администрирования с поддержкой их иерархии.

Особенностью решения Kaspersky Administration Kit является объединение в логическую сеть всех подключенных к серверу администрирования компьютеров, а также подчиненных серверов. Такая сеть не связана с топологией физической сети, хотя при автоматическом ее формировании за основу берутся текущие сетевые настройки. Каждая группа может иметь свои политики и настройки. Таким образом, очень просто создать конфигурацию системы защиты, удовлетворяющую любым условиям.

Для работы сервера администрирования необходим SQL-сервер. Здесь подходит MySQL, Microsoft SQL Server 2000/2005/2008, Microsoft SQL Server 2000 Desktop Engine (MSDE) или SQL Server 2005 Express Edition.

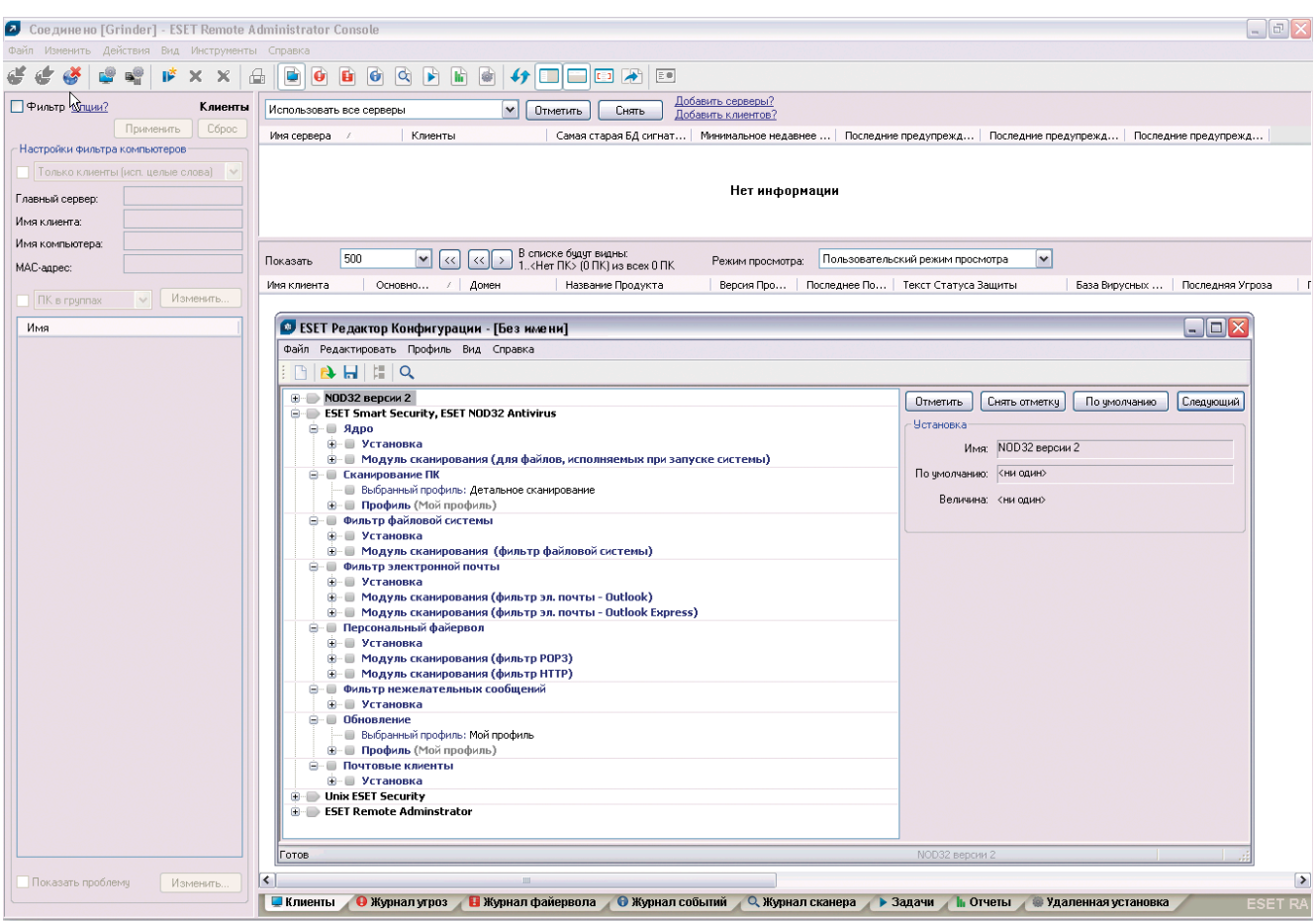
Как видите, все составляющие Kaspersky Administration Kit зависят от платформы.

Подготовленная версия MSDE 2000 SP 3 для Administration Kit доступна на установочном компакт-диске и странице загрузки продуктов «Лаборатория Касперского». Для небольшого офиса его возможностей вполне достаточно, и во время установки он обнаруживается автоматически без лишних донастроек.

Принцип лицензирования в продуктах KOSS самый простой – указываем количество систем, на которые будут установлены компоненты, вне зависимости от того, рабочая это станция или сервер, и без подсчета почтовых ящиков или пользовательских аккаунтов.

«Доктор Веб»

Компания «Доктор Веб» предлагает несколько комплексов защиты, объединенных под названием Dr.Web Security Suite. И в отличие от продуктов «Лаборатория Касперского» каждое ориентировано на применение исключительно в своей сфере: настольные системы, почтовые и файловые серверы, Интернет и SMTP-шлюзы, мобильные устройства и так далее. В контексте статьи нас будет интересовать:



Редактор конфигурации в ESET Remote Administrator Console

- Dr.Web Enterprise Suite (ES) – защита рабочих станций и серверов под управлением Windows 98-Vista, с возможностью централизованного управления;
- Dr.Web для файловых серверов Windows/UNIX/Novell NetWare;
- Dr.Web для интернет-шлюзов UNIX;
- Dr.Web для почтовых серверов UNIX/MS Exchange/IBM Lotus Domino.

Функции последних трех, думаю, понятны, интерес может вызвать ES. Это решение также имеет клиент-серверную архитектуру, позволяя управлять удаленными клиентами при помощи единого графического интерфейса.

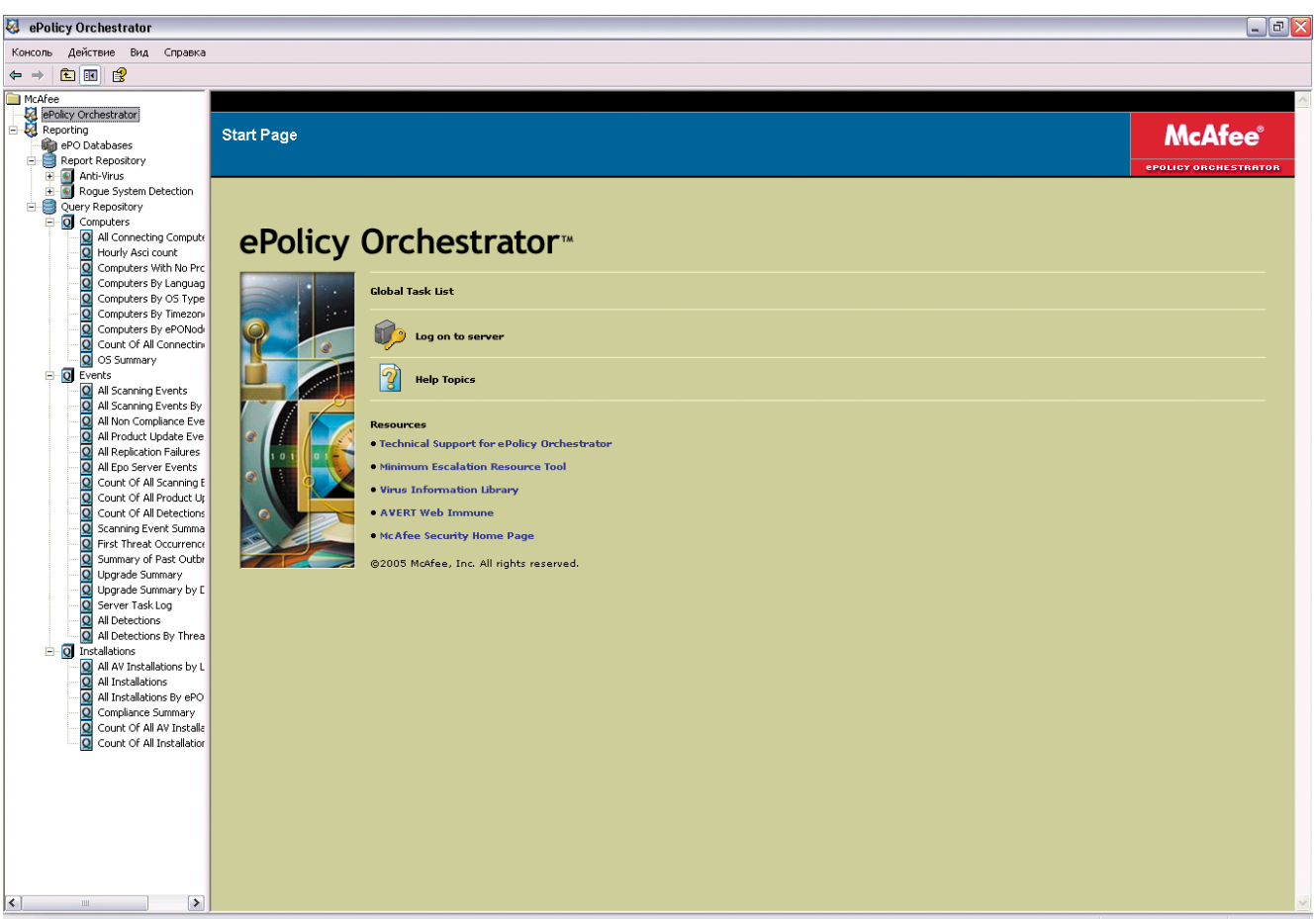
Состоит ES из антивирусного сервера (ES-сервера), консоли администратора, SQL-сервера и ES-агента. Назначение всех компонентов, в общем, аналогично KOSS, отличия составляет лишь агент. Здесь в отличие от «Касперского» агент является полноценным антивирусом, который устанавливается на все защищаемые компьютеры и сам антивирусный сервер. Агент включает все, что пользователи привыкли видеть в продуктах Dr.Web, – антивирусный сканер, файловый монитор SplDer Guard, почтовый фильтр SplDer Mail. Именно поэтому во избежание конфликтов на рабочих станциях не должно быть установлено другое антивирусное ПО, в том числе другие версии Dr.Web. Кстати, в большинстве рассматриваемых в статье решений уже рабочий антивирус этого производителя можно просто подключить к серверу для централизованного управления, что довольно неплохо, особенно в том случае, если за него уплачено.

Специальный компонент позволяет администратору, подключившись к ним, удаленно просканировать компьютеры в «тихом» режиме. В этом случае пользователь вообще не заметит работу сканера, а администратор может наблюдать за проверкой в реальном времени. Для установки агентов предложено использовать один из трех способов:

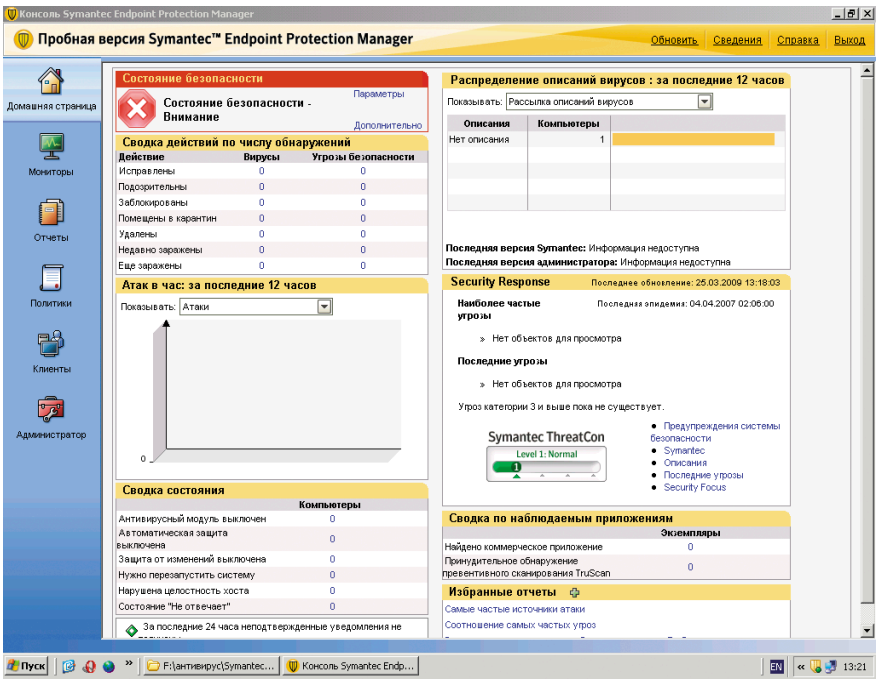
- вручную при помощи утилиты drwinst – подходит для большинства сетей;
- удаленно при помощи ES-консоли – в документации сказано, что лучше подходит не для массового развертывания, а при обычной работе, хотя с его помощью установить агента систем на 50 вовсе не проблематично;
- используя возможности Active Directory – для этого с сайта следует скачать специальный инсталлятор.

Обновление агентов и антивирусных баз производится централизованно с антивирусного сервера, но агенты, установленные на мобильных системах и находящиеся вне своей сети, «умеют» обновляться напрямую с сайтов Dr.Web.

Сервер в базе данных содержит настройки агентов, статистику по сканированиям и найденным вирусам и прочую информацию. В одной сети может быть несколько ES-серверов, связанных с главным сервером.



Консоль управления продуктами McAfee – ePolicy Orchestrator



Основная страница Symantec Endpoint Protection Manager

Архитектура серверной части независима от платформы, что позволяет установить его как на Windows NT/2000/XP/2003 Server, так и на UNIX-системах – Linux, FreeBSD (5.1-7.0) или Solaris (x86). В Linux-версии имеются пакеты для ASPLinux, Debian Etch/Sarge, Fedora Core, SuSE, Mandriva, Ubuntu (для некоторых и под 64-битную систему), а также Linux generic под разные версии Glibc (2.3 – 2.7). В качестве БД может быть использована встроенная СУБД (IntDB), подключение через ODBC (для Windows) или PostgreSQL (в UNIX). Консоль управления написана на Java, поэтому ее установка возможна в любой системе, для которой доступен JRE не ниже 5.0. Для антивирусного сервера с агентами может использоваться как протокол TCP/IP, так и IPX/SPX/NetBIOS, что позволяет не менять настройки сети. Предусмотрена возможность шифрования и сжатия соединения, что позволяет безопасно администрировать сеть через Интернет и экономить трафик.

Кроме этого имеется несколько готовых комплектов для организаций разного размера, в том числе и образовательных учреждений. Под нашу задачу подходит комплект Dr.Web «Универсальный», обеспечивающий защиту:

- рабочих станций Windows (Антивирус Dr.Web Enterprise Suite) – от 5 до 100 (шаг 5 станций);
- файловых серверов Windows/UNIX/Novell NetWare;
- электронной почты на почтовых серверах под управлением UNIX/Microsoft Exchange\Lotus Domino (антивирус и антиспам);
- интернет-шлюзов UNIX.

Также хотелось бы особо отметить, что это единственное решение, для которого нельзя официальным путем получить ключ и построить тестовую платформу, включающую все компоненты, и таким образом определиться в выборе. При заказе ключа для сервера предлагается установить Dr.Web для Windows, который подходит лишь для агентов и Антивирусу Dr.Web для Windows Mobile. Второй ключ для сервера, несмотря на хорошо налаженную службу технической поддержки и мои двухнедельные попытки, я так и не получил. И только благодаря диску журнала [1], на котором нашелся нужный файл, удалось проверить Dr.Web Security Suite в работе. Если так добивается тестовых ключей любой клиент, желающий посмотреть на решение вживую, то, признаюсь, подход компании несколько непонятен.

ESET

На сайте компании ESET для корпоративных клиентов предлагается 6 продуктов, обеспечивающих защиту рабочих станций, шлюзов, почтовых и файловых серверов. Для решения нашей задачи выбираем три:

- ESET NOD32 Smart Security Business Edition – сканирование Windows 2000/XP/Vista;
- ESET NOD32 Gateway Security for Linux/BSD/Solaris;
- ESET NOD32 for MS Exchange Server/IBM Domino/Linux Mail Server.

Из них Smart Security Business Edition (SMBE) является комплексным решением, обеспечивающим защиту как рабочих станций, так и серверов. Его основой является антивирусный модуль ThreatSense, плюс включены модули персонального брандмауэра и антиспама. Персональный брандмауэр обеспечивает сканирование сетевых соединений на уровне канала, определяет и блокирует многие типы сетевых атак, отслеживает изменения в исполняемых файлах, умеет проверять HTTP- и POP3-трафик. Также модуль ThreatSense может интегрироваться в популярные почтовые клиенты (MS Outlook, Outlook Express, Windows Mail и другие).

Реализовано три режима настройки правил фильтрации: автоматический, интерактивный или устанавливающийся централизованно на основе политик. В последнем случае соединения, не разрешенные в правилах, будут заблокированы. Поддерживается работа с файловыми серверами на платформах Windows, Novell Netware и Linux/BSD/Solaris.

Для централизованного администрирования продуктов ESET используется Remote Administrator (ERA), состоящий из трех компонентов:

- сервер ERA (ERA Server, ERAS);
- консоль ERA (ERA Console, ERAC);
- сервер-зеркало – локальный сервер обновлений.

В сети может функционировать любое количество серверов, поддерживается репликация данных на основной (родительский) сервер. При помощи ERAS можно выполнять удаленную установку и удаление антивируса, настраивать параметры его работы и создавать зеркало обновлений. Администратор определяет, какая информация и с какой периодичностью будет передаваться на родитель-

ские серверы автоматически, а какая только по запросу. ERAS функционирует как служба, поэтому для его установки понадобится компьютер с ОС Windows на ядре NT (NT4, 2000, XP, 2003). Для хранения данных ERAS использует MDAC (Microsoft Data Access Components), кроме того, некоторые элементы сохраняются в отдельных файлах в каталоге Storage. Кстати, лицензионное соглашение не накладывает на количество ERAS никаких ограничений, лицензия NOD32 SS BE требуются только для клиентских компьютеров или автономных, файловых серверов под управлением Windows OS, Novell и Linux.

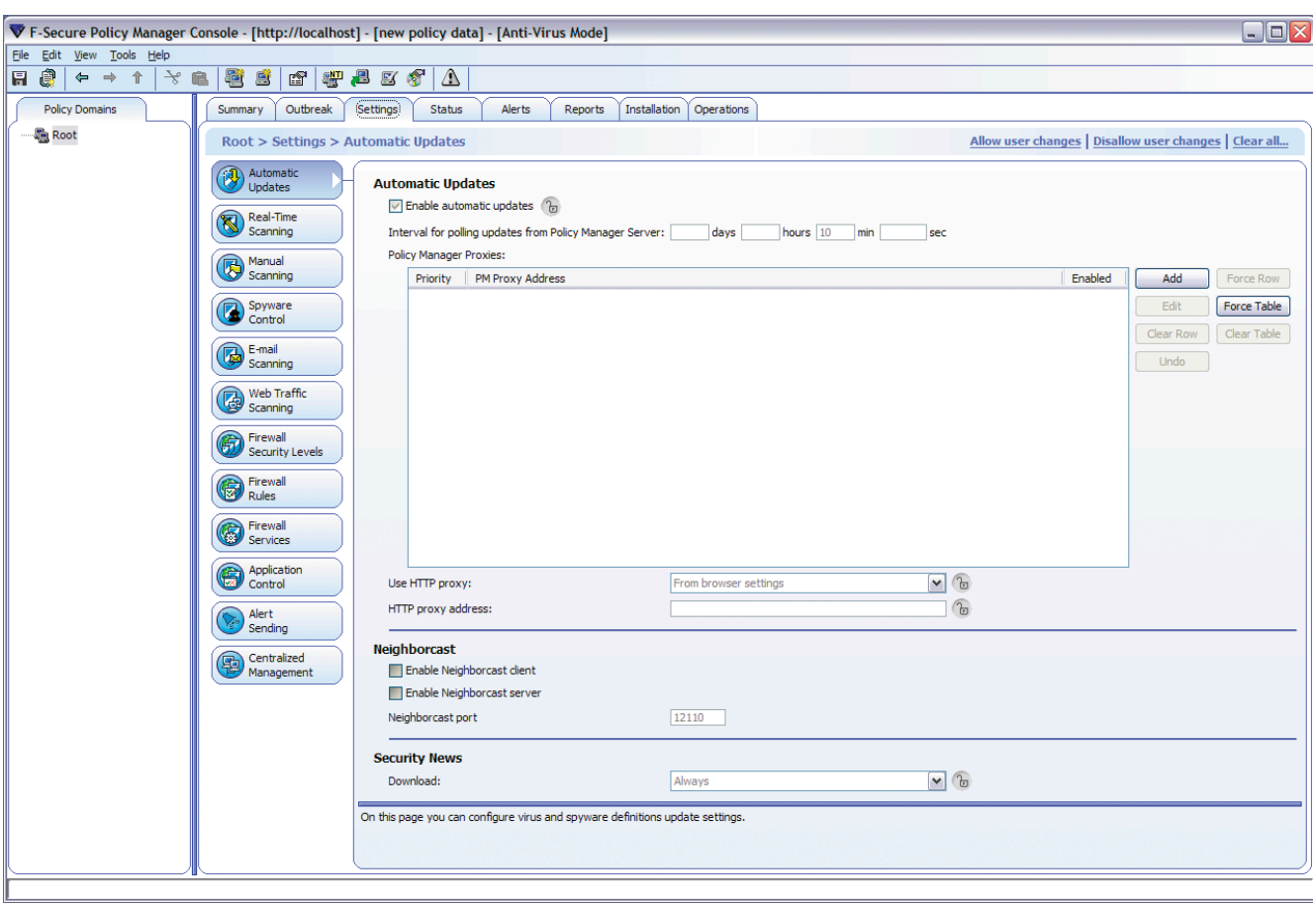
McAfee

С нужными продуктами McAfee не так легко сразу определиться, и в этом очень помогает страница McAfee SMB Product Comparison [2]. Здесь в двух вкладках Protection Type и Protection Area довольно просто выбирать продукт, удовлетворяющий нужным условиям. Для нашего примера наиболее подходит пакет McAfee Active Virus Defense. Это комплексное решение, в котором реализована антивирусная защита для всех компонентов – рабочих станций, интернет-шлюза, почтового и файлового сервера. В таблице показано, что кроме антивирусов сканирующий движок обнаруживает и spyware, но только Partial (частично). В состав пакета включены:

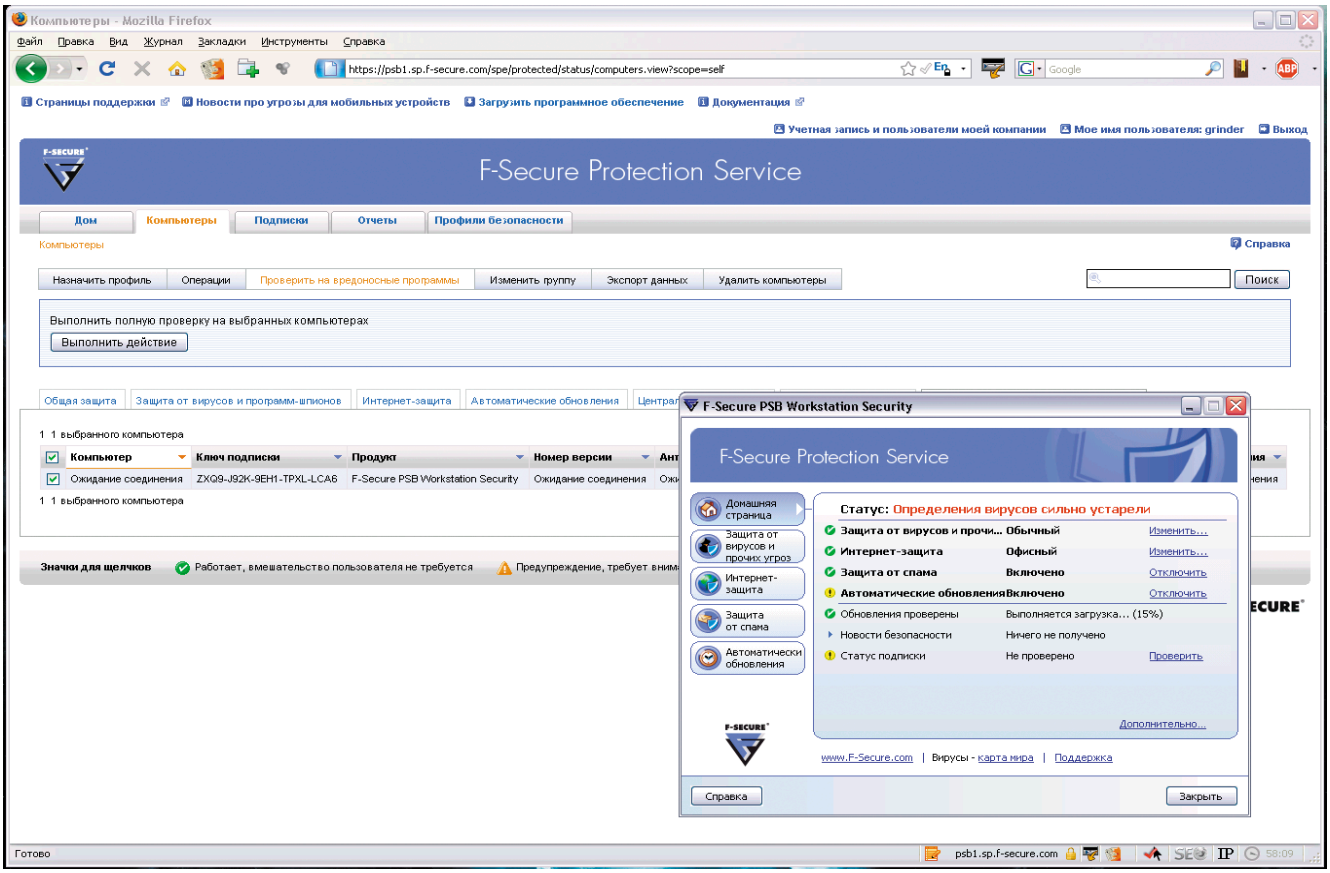
- **McAfee VirusScan Enterprise** – антивирус для рабочих станций и серверов Windows;
- **McAfee NetShield for Netware** – антивирус для серверов Novell Netware;
- **McAfee VirusScan Command Line** – антивирус командной строки для систем DOS, Windows и различных вариантов UNIX-систем, включая Linux;
- **McAfee GroupShield for Microsoft Exchange** – антивирус для серверов Microsoft Exchange 2000/2003/2007;
- **McAfee GroupShield for Lotus Domino** – антивирус для Lotus Domino от версии 6.0.3, работающего под управлением Windows 2000/2003 или Solaris от 2.6, IBM AIX 4.3.3/5.1/5.3;
- **McAfee WebShield SMTP** – SMTP-шлюз, осуществляющий сканирование входящего и исходящего почтового трафика, с интегрированным контент-фильтром.

Централизованное управление защитой осуществляется при помощи ePolicy Orchestrator (ePO). Его задача, в общем, аналогична другим подобным решениям: управление настройками антивирусов, установка обновлений через центральное хранилище Software Repository, получение при помощи AV Informant отчетов о работе защиты и отдельных компонентов. Администратор может задавать единую политику безопасности, включая наличие патчей безопасности от Microsoft для всех систем или отдельных групп.

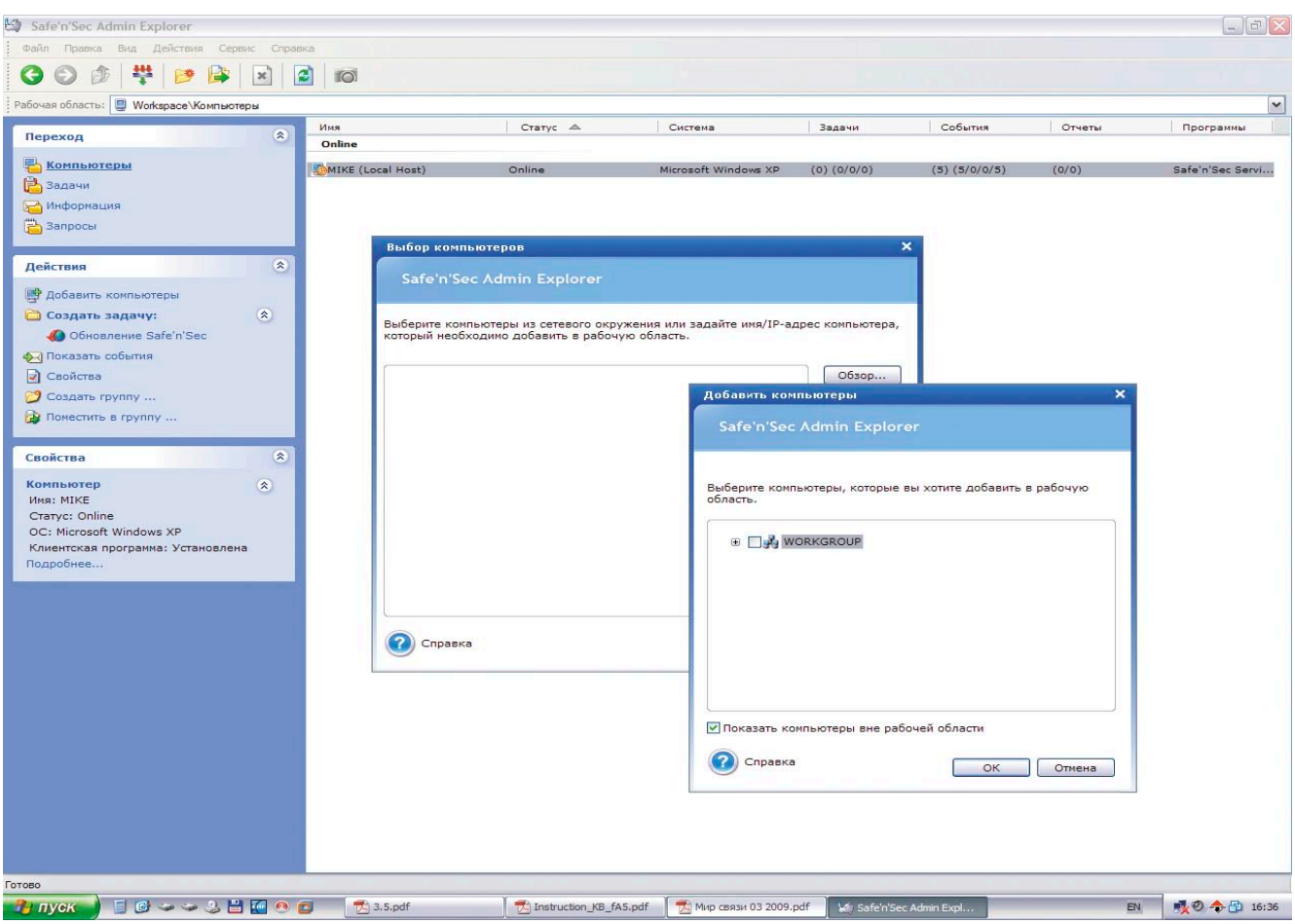
EPO построен по клиент-серверной схеме, напоминающей Kaspersky Administration Kit, то есть в клиентские системы кроме непосредственно продукта, осуществляющего защиту, устанавливается небольшой по размеру агент. Агент, получая команды от сервера и отсылая отчеты, осуществляет непосредственное управление на-



Окно настроек консоли F-Secure Policy Manager



Портал F-Secure PSB с клиентом F-Secure PSB Workstation



Консоль Safe'n'Sec Admin Explorer с подключенным клиентом

стройками клиентской программой. Мобильные системы производят обновления сигнатур при подключении к локальной сети или самостоятельно через Интернет. Для установки ePO на сервер понадобятся компьютер под управлением Windows 2000 SP4/2003 SP1/2/R2, консоль Windows 2000/XP/2003/Vista. В качестве SQL-сервера для небольших организаций рекомендован Microsoft SQL Server 2005 Express Edition, поддерживается SQL Server 2000/2005. Сегодня доступны две версии ePO: в 3.6.1 консоль реализована в виде MMC, в 4.0.0 – в виде веб-сервиса. Во втором варианте для взаимодействия с сервером используется протокол HTTP/HTTPS, что позволяет работать через Интернет с любой платформы.

В политиках ePO также задаются установки для программ-антишпионов и персонального брандмауэра McAfee Desktop Firewall. Кроме продуктов McAfee, ePO поддерживают и некоторые версии Symantec Norton Antivirus для серверов и рабочих станций.

Symantec

Продукция Symantec довольно хорошо известна ИТ-специалистам, те, кто работал в Windows 9x, хорошо помнят антивирус Norton AntiVirus, обновленная версия которого выпускается этой корпорацией до сих пор. Сегодня для небольших компаний предложен пакет Symantec Endpoint Protection Small Business Edition 11.0, фактически состоящий из двух решений: пакета Symantec Endpoint Protection 11.0 (SEP) и антивируса Symantec Mail Security for Microsoft Exchange (Symantec Mail Security for SMTP/Domino/Enterprise Edition).

Клиент SEP является логическим продолжением Norton AntiVirus и обеспечивает защиту от вирусов и шпионских программ, фильтрацию открытого и шифрованного сетевого трафика, система защиты от атак Generic Exploit Blocking позволяет блокировать угрозы эксплуатирующих уязвимости, инструмент VxMS (Veritas Mapping Service) позволяет обнаруживать руткиты. Кроме сигнатурного анализа, модуль проактивной защиты Proactive ThreatScan обнаруживает заразу на основе анализа поведения приложений. Администратор имеет возможность контролировать и при необходимости блокировать доступ пользователей и работающих программ к процессам, файлам и каталогам, контролировать элементы ОС и реестра, модулей и элементов операционной системы и приложений. За дополнительную плату доступен модуль Symantec Network Access Control, обеспечивающий проверку систем и определяющий на основе полученных данных права доступа к сети и ресурсам.

Список поддерживаемых клиентских машин и серверов довольно большой: ОС Windows 2000/XP/2003/Vista/2008 (разных редакций), Linux (Red Hat Enterprise Linux от 3.x, SuSE Linux Enterprise (server/desktop) от 9.x, Novell Open Enterprise Server (OES/OES2), Ubuntu от 7.x и Debian 4.x), а также VMWare ESX 2.5, 3.x. Что немаловажно, все ОС могут быть как 32, так и 64-битных версий. Особо отмечено, что Itanium не поддерживается (как и всеми остальными решениями).

Управление осуществляется с единой консоли. Сервер и консоль управления Symantec Endpoint Protection Manager устанавливаются на ОС Windows от 2000, в качестве SQL-базы данных может быть использована как встроенная, так и MS SQL 2000 SP3/2005. Кроме этого, для установки потребуется наличие IIS. Встроенная БД на основе Sybase рекомендуется при подключении до 100 клиентов и автоматически устанавливается при выборе режима инсталляции сервера управления «Простой». При соблюдении всех требований установка сервера достаточно проста, в ее ходе можно указать другой порт для вебсайта доступа, с которого скачивается клиент управления. По окончании установки запускается мастер переноса и развертывания, который поможет развернуть антивирус на клиентских системах и перенести группы и политики с родительских серверов Symantec AntiVirus.

В критериях поиска клиентов можно указать более 30 параметров, включая имя пользователя, компьютера, группу, IP-адрес и даже такие, как частота процессора, версия BIOS и так да-

Safe'n'Sec Enterprise

Еще один продукт, о котором бы хотелось вкратце рассказать, подходит под выдвинутые требования лишь частично, но, учитывая его возможности и стоимость, умолчать не имею права. Речь идет о системе предотвращения вторжений (HIPS, Host Intrusion Prevention System) Safe'n'Sec Enterprise, разрабатываемой российской компанией S.N.Safe&Software (http://www.safensoft.ru). В отличие от других программ обзора она не является антивирусом как таковым. В клиентских модулях Safe'n'Sec Timing Client используется собственная технология проактивной защиты V.I.P.O. (Valid Inside Permitted Operations), основанной на разграничении системных привилегий при работе компьютера.

После установки клиент сканирует систему, создает профиль приложений и формирует список доверенных программ (контролируется хеш SHA-256). При появлении активности, затрагивающей целостность системных файлов или реестра, а также нового процесса, соответствующая операция блокируется, а пользователю выдается запрос на ее подтверждение. Таким образом, технология, используемая в Safe'n'Sec, позволяет блокировать любые известные и неизвестные вредоносные программы, для которых еще нет сигнатуры в антивирусных базах. Еще один плюс Safe'n'Sec – не требуется постоянное обновление баз. Кроме обычной, существуют версии клиентской программы, содержащие антивирусный модуль Dr.Web и модуль защиты от программ-шпионов. В этом случае обеспечивается и лечение зараженных файлов.

Централизованное управление обеспечивается при помощи двух программ:

- ☑ Safe'n'Sec Admin Explorer – консоль управления, используемая для удаленного администрирования системы безопасности;
- ☑ Service Center – сервер, при помощи которого непосредственно производится управление клиентскими программами, их централизованное обновление и создание отчетов и оповещение администратора о возникновении определенных событий.

В качестве СУБД используются MSDE 2000 SP3, MS SQL 2005 Express Edition или MS SQL Server от 2000 SP, поддерживаемые ОС Microsoft Windows 2000SP3/XP/2003/Vista. Клиентская лицензия на 50 рабочих мест, включая Admin Explorer и Service Center, составляет 27600 руб.

Таблица 1. Приблизительные цены системы защиты для 1 шлюза, 50 рабочих станций, 1 файлового сервера, 1 почтового сервера (предоставлены ЗАО «Софткей» (http://www.softkey.ru)

Производитель	Адрес	Состав решения	Стоимость отдельных компонентов	Суммарная стоимость готового решения	Варианты
«Лаборатория Касперского»	http://www.kaspersky.ru	Kaspersky Total Space Security	53 комп x 2837 руб	стоимость подписки на 1 год – 150361 руб.	–
«Доктор Веб»	http://www.drweb.com	Dr.Web Enterprise Suite или «Антивирус Dr.Web для Windows»	на 50 ПК x 625 руб = 31250 руб.	111750 руб.	Комплект «УНИВЕРСАЛЬНЫЙ-50», лицензия на 12 месяцев Dr.Web ES (Антивирус) 50 рабочих станций + «Dr.Web® для Windows-серверов» + «Dr.Web защита почты (Антивирус+Антиспам)» 50 почт. ящиков + «Dr.Web для интернет-шлюзов» на 50 ПК, на 12 месяцев – 45250 руб.
		«Антивирус Dr.Web «Защита бизнеса (решения для почтовых серверов)»	на 50 ящиков 20250 руб. (лицензия 12 месяцев)		–
		Или «Dr.Web Антивирус+Антиспам для почтовых серверов UNIX, MS Exchange, IBM Lotus Domino»	на 50 пользователей 26250 руб.		–
		«Антивирус Dr.Web для интернет-шлюзов»	на 50 ПК 20000 руб (лицензия 12 месяцев)		–
		«Dr.Web для файловых серверов»	7000 руб (лицензия 12 месяцев)		–
ESET	http://www.esetnod32.ru	ESET NOD32 Smart Security Business Edition	на 52 ПК 66600 руб.	105437 руб.	–
		ESET NOD32 Gateway Security for Linux/BSD/Solaris	на 50 ПК 22450 руб.		–
		Антивирус NOD32 Exchange Mail Server	на 50 ящиков 16387 руб.		–
McAfee	http://www.mcafee.com/ru	McAfee Active Virus Defense ProtectPLUS (Perpetual License with 1yr Gold Software Support)	на 53 ПК x 2104.53 руб.	111540.09 руб.	–
Symantec	http://www.symantec.com/ru	Endpoint Protection 11.0	на 53 ПК 47222.47 руб.	80660.97 руб.	–
		Symantec Mail Security for Microsoft Exchange	на 50 ящиков 33438.50 руб.		–
F-Secure*	http://www.f-secure.ru	F-Secure Anti-Virus Small Business Suite	2144.35 руб. x 52 = 111506.2 руб.	139111.2 руб. или 70505 руб.	–
		Или F-Secure Protection Service for Business	825.00 руб x 52 = 42900 руб.		–
		F-Secure Internet Gatekeeper	552.10 руб x 50 = 27605 руб.		–

* Цена составлена по данным http://www.f-secure.ru, http://www.anysoft.ru и http://www.netsecret.ru

лее. Также хочется отметить наличие удобного и понятного инструмента сохранения и восстановления базы системы защиты.

F-Secure

Решения F-Secure для малого и среднего бизнеса [3] включают в себя все необходимые программные продукты, позволяющие защитить от вирусов рабочие станции, мобильные устройства, файловые и почтовые серверы, шлюзы. Для удобства выбора представлены готовые комплекты. Для наших условий наиболее подходят два решения: F-Secure Anti-Virus Small Business Suite (SBS) и F-Secure Protection Service for Business (PSB), каждое по-своему интересно. Лицензия на SBS ориентирована на 5-99 систем.

Состоит SBS из 4 компонентов:

- **F-Secure Anti-Virus Client Security** – интегрированное решение для Windows 98/ME/NT4.0/2000/XP, включающее антивирус, антишпион, брандмауэр, сканер руткит BlackLight и систему контроля поведения HIPS DeepGuard. Обеспечивается проверка POP3/SMTP/IMAP4/HTTP-трафика;
- **F-Secure Anti-Virus for Windows Servers** – защита файловых серверов Windows NT4.0/2000/2003 (32/62-бит) от вирусов, шпионского ПО и потенциально опасных программ;
- **F-Secure Anti-Virus for Microsoft Exchange** – контроль исходящих и входящий сообщений Microsoft Exchange 2000/2003/2007;
- **F-Secure Policy Manager** – система централизованного управления, контроля, обновления и наблюдения за работой систем безопасности.

Anti-Virus Client Security поддерживает также Cisco NAC (Network Admission Control), применение которой позволяет гарантировать, что подключающаяся к сети система будет удовлетворять требованиям всех политик компании.

Policy Manager построен по клиент-серверной схеме, в которой управление настройками производится удаленно, при помощи терминала, но есть свои особенности. Так, графические отчеты создаются в модуле, имеющем название F-Secure Policy Manager Web

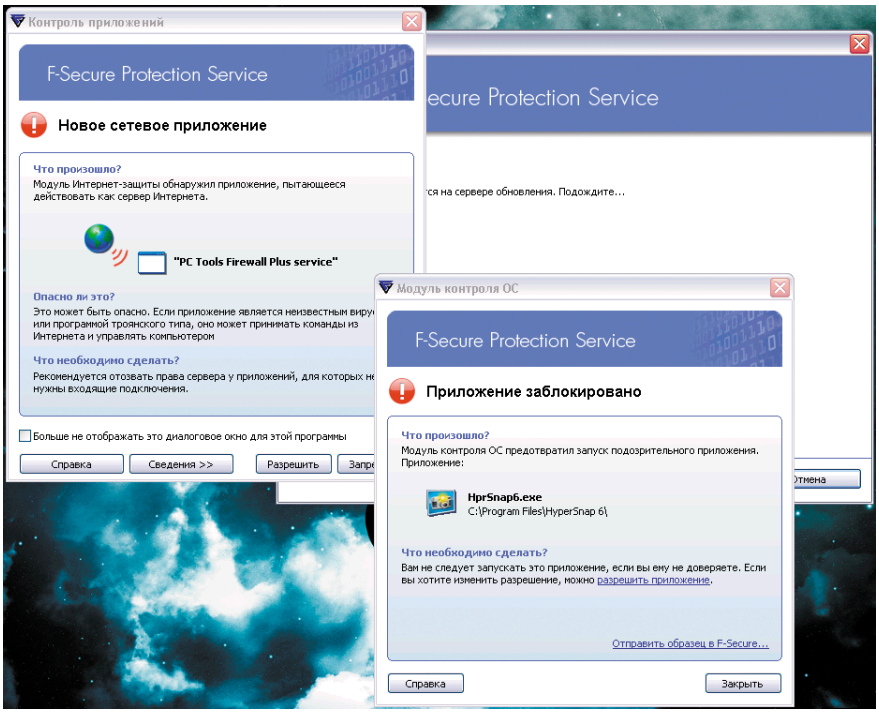
Таблица 2. Возможности антивирусных систем

Производитель	«Лаборатория Касперского»	«Доктор Веб»	ESET	McAfee	Symantec	F-Secure Anti-Virus Small Business Suite	F-Secure Protection Service for Business
Управление ОС	Windows	Кроссплатформенное. Любая система, поддерживающая JRE не ниже 5.0	Windows 98/ Me/2000/XP	Windows (версия 3.x) или WEB (4.x версия)	Windows	Windows 2000/ XP/2003	Поддерживаются Internet Explorer 6.x или свежее с разрешенными JavaScript и cookies Firefox 2.x, или свежее с разрешенными JavaScript и cookies, Profile Editor требует Java RE 1.6
Консоль локализация	+	+	+	–	+	–	+
Сервер управления ОС	Microsoft Windows NT/2000/2003/2008/ XP/Vista	Windows NT 4.0/ 2000/XP/2003/2008, Linux, FreeBSD (до 7.0), Solaris (x86 и Sparc) для 32- и 64-битных систем	Windows на ядре NT/2000/XP/2003	Windows 2000/2003	Windows 2000/ 2003/2008	Windows 2000/ 2003/XP(для небольшого количества соединений), Linux	Нет
СУБД	MySQL, Microsoft SQL Server 2000/ 2005/2008, MSDE, SQL Server 2005 Express Edition	Встроенная, подключение через ODBC, PostgreSQL	Внутренняя	MS SQL Server 2000/2005/2005 Express Edition	Встроенная, MS SQL 2000SP3/ 2005	Встроенная	Нет
Клиент ОС	Windows, Linux, мобильные системы (Windows Mobile, Symbian)	Windows 95/98/Me/ NT/2000/XP/Vista (32-бит)	Windows 2000/XP/ Vista	Windows 98/ME, NT 4.0 SP6, 2000, 2003, XP, Vista, 2008 (в том числе Core Server); Windows Terminal Server от NT; Microsoft Cluster Server (MSCS); XP Tablet PC, Citrix MetaFrame 1.8 & XP; EMC Celerra File Server Novell Netware UNIX-системы в командной строке	ОС Windows 2000/XP/2003/ Vista/2008 разных редакций и Linux (Red Hat Enterprise Linux от 3.x, SuSE Linux Enterprise (server/desktop) от 9.x, Novell Open Enterprise Server (OES/OES2), Ubuntu от 7.x и Debian 4.x), а также VMWare ESX 2.5, 3.x.	Windows 98/ME/ NT4.0/2000/XP (32-bit). Отдельная возможность подключения Linux	Microsoft Windows 2000/XP/Vista (32-бит)
Клиент-возможности	Антивирус, проактивная защита, защита файловой системы, брандмауэр с IDS/IPS, проверка POP3, IMAP, MAPI, NNTP, HTTP. Антифишинг и антиспам	Антивирусный сканер, файловый монитор, почтовый фильтр, «тихий» режим сканирования	Антивирус, брандмауэр, антиспам, блокировка многих типов сетевых атак, контроль файлов	Антивирус, частично Adware	Антивирус, брандмауэр, проактивный модуль, защита почты, контроль процессов и приложений, Symantec Network Access Control	Антивирус, брандмауэр антишпион, сканер руткит, контроль поведения (HIPS), Cisco NAC	Нет
Почтовый сервер	Sendmail, Qmail, Postfix, Exim, Exchange 2003/ 2007, Lotus Notes/ Domino	UNIX/MS Exchange/ IBM Lotus Domino	UNIX/MS Exchange/ IBM Lotus Domino	Microsoft Exchange 2000/2003/2007; Lotus Domino от 6.0.2 для Windows 2000/2003	SMTP/Domino, Linux, Microsoft Exchange 2007	Microsoft Exchange 2000/2003/2007	Нет
Файловый сервер	Windows, Linux, Samba, NetWare	Windows/UNIX/ Novell NetWare	Windows, Novell Netware/ Linux/ *BSD/Solaris	Windows NT/2000/ 2003/2008	То же, что и клиент	Windows NT4.0/ 2000/2003 (32/62-бит)	Windows 2000/2003 Server (32-бит)
Интернет-шлюз	Microsoft ISA Server, Check Point FireWall	UNIX	Linux/*BSD/Solaris	SMTP-трафик, установка на Windows NT/2000/2003	Весь трафик	SMTP/HTTP/FTP	Нет

Reporting. Благодаря наличию готовых шаблонов администратор может быстро найти системы, в которых не установлены последние обновления, проверить настройки безопасности и так далее. Заявлено, что один сервер может управлять установкой ПО на 15 тысяч узлов. При необходимости в одной сети можно использовать несколько серверов. Для отправки обновлений антивирусных баз в удаленные офисы используется специальный компонент F-Secure Policy Manager Proxy.

Policy Manager поддерживает установку на Windows Server 2000/2003 (XP, только консоль), есть и Linux-версия, ориентированная на установку в: Red Hat Enterprise Linux 3/4, SuSE Linux 9/10, SuSe Linux Enterprise Server 9 и Debian Sarge 3.1.

В том случае, если заявленной функциональности не хватает, можно использовать и другие продукты F-Secure, которые обеспечивают защиту рабочих станций и файловых серверов на Linux, серверов Citrix, мобильных устройств и так далее. Все они поддерживают управление при помощи F-Secure Policy Manager. Хотя конечная цена такого решения, вероят-

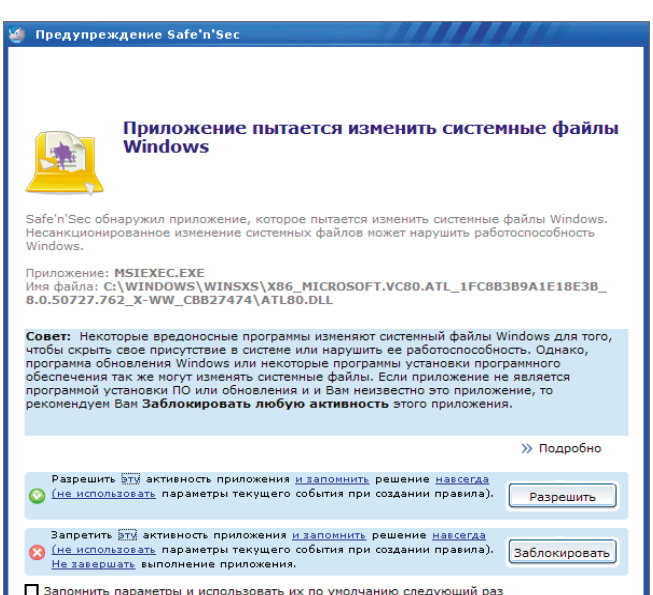


Модуль контроля приложений F-Secure блокирует неизвестные процессы

но, будет чуть выше. В частности, в нашем примере Secure Anti-Virus Small Business Suite следует дополнить F-Secure Internet Gatekeeper, который предназначен для использования в интернет-шлюзах и обеспечивает фильтрацию почтового (SMTP) и HTTP/FTP-трафика, а также контроль доступа, спама и содержимого.

Теперь следующий продукт – F-Secure PSB, доступный в двух версиях: Standard и Advanced. Это решение несколько иного рода. Возможности клиентской части F-Secure PSB Workstation, предназначенной для защиты рабочих станций и серверов, работающих под управлением Windows, в общем, аналогичны Client Security. То есть содержит средства защиты от вирусов, шпионского ПО и руткитов, имеет встроенный брандмауэр, средства предотвращения вторжений, контроля приложений и спама. Централизованное управление производится при помощи простого интерактивного портала через Интернет. Сам портал размещен на серверах F-Secure, поэтому отдельной системы для его установки не требуется, а сам портал доступен из любого места в любое время.

Пользователю для доступа к порталу выдается код подписки (subscription code), который используется при регистрации. Аналогичный код подписки вводится во время установки клиентов (инсталляционные пакеты скачиваются с портала). Система управления достаточно проста и к тому же локализована. Настройки производятся при помощи профилей безопасности, сразу после подключения доступны 7 предустановленных профилей для различных типов систем (1 – для сервера, 2 – ноутбуки и 4 – офисные машины). Отсутствие сервера обновлений в локальной сети не увеличит трафик. Агент автоматического обновления при подключении к Интернету скачивает файлы на одну рабочую станцию, откуда они распространяются на другие системы.



При появлении нового процесса Safe'n'Sec блокирует его выполнение до принятия решения пользователем

Заключение

Итак, все системы защиты, несмотря на некоторую схожесть в архитектуре, все же очень отличаются по функциям и стоимости. На общем фоне выделяется решение от Symantec и «Касперского» – оснащенный клиент для нескольких платформ, удобная консоль, но первый обойдется гораздо дешевле. Если говорить об экономии средств, то здесь неплохо выглядят F-Secure PSB и Safe'n'Sec Enterprise. Учитывая, что для F-Secure PSB не нужен свой сервера управления с СУБД, это уменьшает стоимость и упрощает администрирование, его можно рекомендовать для небольших групп, не имеющих своего штатного администратора. Для сети, в которую входят Windows-системы, можно порекомендовать комплект Dr.Web «УНИВЕРСАЛЬНЫЙ», правда, здесь нужно помнить, что за продление лицензии, вероятно, придется заплатить полную стоимость. Хотя у продуктов Dr.Web есть еще один плюс – клиент и сервер управления независимы от платформы, а работа агента практически «незаметна». Решения ESET и F-Secure SBS находятся примерно на одном уровне. Но первый хорошо знаком нашему пользователю, кроме того, консоль управления локализована, второй меньше стоит. McAfee впечатляет выбором клиентских платформ, но сам клиент сильно отстает по сравнению с другими решениями.

1. Архивный диск журнала «Системный администратор» – http://www.samag.ru/disk.
2. McAfee SMB Product Comparison – http://shop.mcafee.com/ProductRecommender.aspx.
3. Решения F-Secure для малого и среднего бизнеса – http://www.f-secure.ru/small_businesses/products_a-z.

Тотальная защита локальных сетей

В этот раз на тестирование в редакцию попало устройство Dr.Web Office Shield. Устройство от именитой фирмы предназначено для обеспечения антивирусной защиты локальных сетей. Не секрет, что подобных устройств существует достаточно много – многие антивирусные компании имеют их в своих продуктовых линейках. Но в отличие от своих конкурентов Dr.Web Office Shield в дополнение к функциям защиты почтового и интернет-трафика может защищать еще рабочие станции и серверы, так как в состав программного обеспечения устройства входит хорошо себя зарекомендовавший продукт от компании «Доктор Веб» – Dr.Web Enterprise Suite. Явными преимуществами можно назвать то, что аппарат обладает весьма скромными размерами, нетребователен в обслуживании и рассчитан на обслуживание неквалифицированным персоналом – в сопровождающих поставку инструкциях все этапы настройки расписаны достаточно подробно и богато проиллюстрированы. На этом преамбулу считаем законченной и приступаем непосредственно к тестированию.

Сразу после распаковки устройство согласно инструкции было подключено в локальную сеть через разъем LAN1, которому по умолчанию соответствует адрес 192.168.1.100. Настройка устройства полностью проходит через веб-интерфейс, поэтому в качестве первого шага открываем в браузере страницу https://192.168.1.100:1000 и загружаем полученные лицензионные ключи. Вопреки всем ожиданиям выполнение этой, казалось бы, достаточно простой задачи вызвало ряд затруднений, так как браузер от Microsoft (IE7) отказался их загружать, и загрузку пришлось выполнять с помощью Firefox. Как ни странно, в дальнейшем при выполнении настроек ни в IE7, ни в Firefox проблем замечено не было. Здесь же необходимо отметить, что для устройства важен порядок загрузки ключей – вначале необходимо загрузить enterprise.key, затем agent и только потом drweb. Как пояснили разработчики, это связано с особенностями реализации функционала в вошедших в состав устройства продуктов компании. После загрузки ключей в веб-интерфейсе становятся доступными страницы настроек.

Первым делом меняем пароль, устанавливаем, если это необходимо, правильное системное время и выбираем удобный для нас язык локализации. По умолчанию устройство способно работать в качестве основы локальной сети – на нем запущены серверы DHCP и DNS. Поскольку в нашей сети сервер DHCP уже есть, то переходим на страницу «Настройки DHCP» и снимаем соответствующую галочку. Просто останавливаем работу данного сервиса на странице «Сервер DHCP», на которую можно перейти из расположенного справа меню. Привычная для многих администраторов функция прямого редактирования списка сервисов в устройстве отсутствует, так как необходимость их функционирования во многом зависит от функционала, прописанного в лицензионных ключах, и их простое включение может привести к различного рода проблемам. Поскольку диапазон нашей локальной сети отличается от используемого в устройстве по умолчанию, то выбираем «Сетевые интерфейсы» и меняем адрес LAN на рабочий – в данном случае 192.168.10.7. Нажимаем «Применить» и дожидаемся обновления страницы веб-интерфейса. В Dr.Web Office Shield, также по умолчанию, реализована защита беспроводных сетей и VPN. Поскольку ни то ни другое нам не нужно, то отключаем их соответственно на страницах «Сетевые интерфейсы» и «Настройки VPN». Так как настройки DNS и WAN в дефолтном состоянии нас устраивают, то подключаем кабель WAN к LAN3. Устройство получает адрес по DHCP, исходящий шлюз и внешние DNS.

Проверяем фильтрацию веб-трафика. Меняем настройки рабочей станции с DHCP на ручные, указываем устройство в качестве исходящего шлюза и пытаемся загрузить тестовый инфицированный файл со страницы http://www.eicar.org/anti_virus_test_file.htm. Результатом данной попытки является уведомление о блокировке – что и требовалось получить. Для тестирования сервиса фильтрации почты можно послать тот же псевдовирус в качестве вложения. Работа же сервиса фильтрации спама становится заметной практически сразу – весь приходящий спам по умолчанию помечается префиксом.

Переходим в соответствии с инструкцией к проверке Dr.Web Enterprise Suite. Установку агентов защиты можно проводить как вручную, так и автоматически – через консоль. Как обычно, проводим установку вручную, путем запуска «drwinst 192.168.10.7». Правда, вопреки инструкции не копируем необходимые файлы на каждую рабочую станцию, а просто монтируем сетевой ресурс. Установка успешно прошла на MS Windows XP Home, MS Windows XP Prof и MS Windows Vista Home. Все установленные клиенты обновились, станции были перезагружены, и защита локальных компьютеров заработала в штатном режиме. Устанавливаем консоль Enterprise Suite, с которой в дальнейшем будем проводить настройку параметров работы антивирусной защиты сети. Также, вопреки инструкции, опять ничего копировать не будем, а просто подключаем сетевой ресурс \\192.168.10.7\\public как диск и запускаем drweb-*-consolewindows.exe прямо оттуда. После установки перетаскиваем иконку консоли на рабочий стол и запускаем консоль, как указано в инструкции.

Итак, испытания показали, что устройство удовлетворительно работает и может быть рекомендовано для использования при условии точного соблюдения приложенных инструкций.

Вячеслав Медведев

Обновление конфигурации. Как избавить себя от лишних проблем

Андрей Луконькин

Редко какая организация может работать в типовой программе. Рано или поздно понадобятся изменения или обновления. И вот когда сначала вносятся изменения, а потом возникает потребность в обновлении, тогда и возникает вопрос – а как же обновить базу, если в конфигурацию уже были внесены свои корректировки?

Прежде всего нужно определиться с целью обновления, то есть ответить на вопрос: «А зачем мы хотим это сделать?». Как правило, к новым релизам прилагается описание внесенных изменений и добавленного функционала. Например, если фирма «1С» выпустила новую версию программы, в которой только появилась возможность ведения учета по добровольным взносам в ПФР и ничего более, а в вашей организации ничего подобного не нужно, то соответственно и необходимость проведения работ по обновлению ставится под вопрос. В этом случае, возможно, стоит дождаться выхода последующих релизов для экономии времени и денег, а также для снижения риска появления ошибок.

Если всё же пришли к выводу о необходимости проведения обновления конфигурации, то в первую очередь делается резервная копия базы данных. Как говорится в современной поговорке: «Бэкап лишним не бывает. Проверено – это каждый раз так!».

Следующим важным шагом будет определение, насколько база отличается от типовой (а возможно, она и полностью типовая!). Для этого нужно сравнить текущую рабочую конфигурацию с типовой конфигурацией той же версии (релиза).

- Узнать номер релиза рабочей базы (пусть это будет 1.2.20.2).

- Установить типовую конфигурацию такой же версии и сохранить конфигурацию в файл (меню «Конфигурация → Сохранить конфигурацию в файл»), назовем его «типовая 1_2_20_2.cf».
- В конфигураторе рабочей базы провести сравнение (меню «Сравнить, объединить с конфигурацией из файла») с сохраненным нами файлом «типовая 1_2_20_2.cf».
- Если получили сообщение «Конфигурации идентичны», значит, нам повезло, используется полностью типовая база и обновить ее можно через меню «Конфигурация → Поддержка → Обновить конфигурацию». Если же появилось окно сравнения с указанием отличающихся объектов, то нужно приступать к следующему этапу работы.

Рисунок 1. Для «тонкого» обновления одновременно используется 4 окна

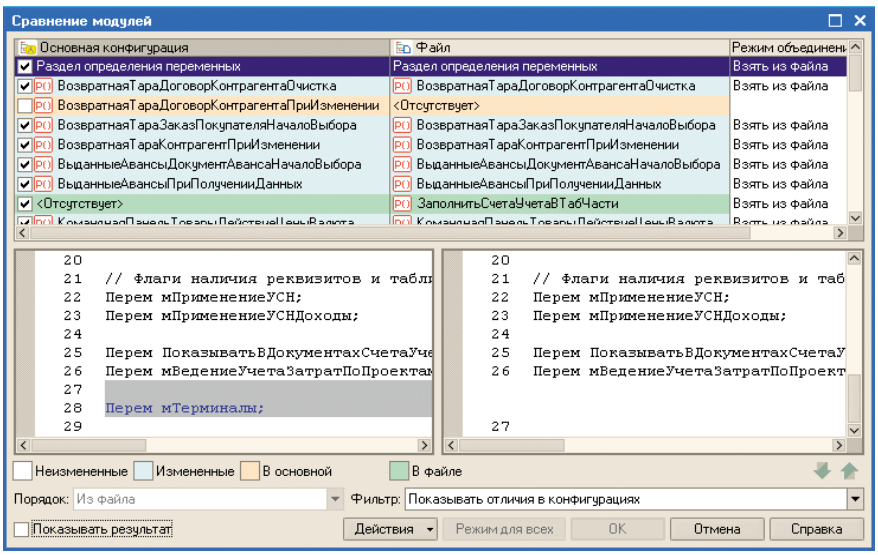


Рисунок 2. Окно сравнения модулей. Отображаются только отличающиеся части модуля

Итак, мы выяснили, что в базу вносились изменения, и обычное обновление может испортить наши доработки. Чтобы сохранить их, выясним, какие именно изменения были, с точностью до объекта. После этого проведем аккуратное, «тонкое» обновление.

Обычно я использую режим «4 конфигуратора» (см. **рис. 1**). Что же это такое?

- **1-е окно конфигуратора:** наша рабочая база (номер релиза 1.2.20), которую необходимо обновить. Здесь мы будем частично объединять с типовым релизом, частично вносить что-то руками в модули и править формы.
- **2-е окно конфигуратора:** в нем открыто окно сравнения нашей рабочей базы с типовым релизом той же версии (1.2.20). Таким образом, мы выявим, какие объекты нельзя обновлять автоматически, т.к. они отличаются от типовых. Назовем это «Список».
- **3-е окно конфигуратора:** сравнение типового релиза 1.2.20 и нового типового 1.2.21. Здесь наглядно будут видны все изменения, которые предлагает фирма «1С».
- **4-е окно конфигуратора:** типовая конфигурация нового релиза 1.2.21, чтобы отсюда можно было копировать объекты, процедуры или отдельные куски кода программы.

Имея точный список измененных объектов («Список»), можно смело запускать объединение в 1-м окне. После проверки появится окно сравнения и объединения, в котором нужно снять галки с тех объектов, которые входят в «Список».

Затем начинается самое интересное – аккуратно вручную обновляем измененные объекты. То есть в 1-м конфигураторе вносим изменения, которые видим в окне сравнения 3-го конфигуратора. Чем 8-я версия платформы выгодно отличается от 7.7, так это тем, что есть возможность видеть различия в модулях с разбивкой по процедурам. То есть выводится не один огромный текст (например, глобальный модуль в 7.7), в котором найти одну-единственную измененную строчку достаточно проблематично, а только те процедуры, в которых были корректировки (см. **рис. 2**).

После окончания процесса обновления лучше будет провести хотя бы небольшое тестирование функционала программы, хотя бы тех объектов, которые изменялись ранее самостоятельно и обновлялись вручную.

Как можно облегчить себе жизнь, если часто вносятся изменения в базу?

Для этого, во-первых, в номере релиза измененной базы ставится отличительный знак, например «*». Это будет означать, что конфигурация отличается от типовой, и тогда не придется ломать голову в попытках вспомнить «меняли мы тут что-то или нет».

Во-вторых, если соблюдать некоторые нехитрые правила, то количество измененных типовых объектов можно свести к минимуму.

- По возможности не изменять стандартные процедуры. Создайте общий модуль, в котором будут размещаться созданные или измененные вами процедуры и функции. Таким образом, все изменения сведутся только к одной строке вызова нужной процедуры.
- Если нужно скорректировать форму документа, справочника или обработки, роль или интерфейс, то лучше создать копию и её уже изменять под свои нужды.
- Печатные формы и отчеты могут храниться во внешних файлах.
- Оставляйте комментарии в текстах модулей. Этим вы избавитесь от вопросов «кто, когда и зачем это менял?».

Конечно, важно понимать смысл того или иного действия, и лучше составить небольшой план действий перед началом работы. Например, добавить сначала базовые объекты (константы, перечисления) и только потом справочники, документы и регистры. Но помните, что всегда есть архивная копия (она просто обязана быть!), которая не оставит организацию без информационной базы.

Жизнь в стиле хокку

Для меня общение – это когда один человек обменивается своей мудростью с другим. И не важно, считают ли оба собеседника себя мудрецами, много ли они говорят друг другу или просто молча улыбаются, глядя на ребенка, который носится по детской площадке… Житейской мудрости можно научиться по-разному… Мне кажется, что, завершив интервью с главой представительства Kerio Technologies в России и странах СНГ Максимом Акимовым, я не только получила очередной материал для публикации, но и поняла для себя что-то очень важное.

Что именно, трудно сформулировать. Размышляя над записанными на пленку короткими, емкими, четкими ответами Максима, я догадалась, наконец, что они мне напоминают – хокку или хайку, танки, философские лаконичные вирши японской поэзии. Знаете, некоторые любители гадают по этим стихам, пытаясь угадать, что готовит им судьба…

Тридцатилетний Максим Акимов рассказывал о своей судьбе, вернее, о маршрутах, которыми он шел до сих пор по жизни. И мне показалось весьма любопытным показать и вам эти маршруты, связав их с бессмертными строками.

О родительской мудрости

Скажи мне, для чего,
О ворон, в шумный город
Отсюда ты летишь?

Басё

Как хорош этот мир!
Звенят над лугами цикады,
Соколы кружат...

Иссa

Родной город Максима – древний Борисоглебск Воронежской области. Слышите, какое удивительное название? Борис и Глеб – русские святые, в их честь на слиянии рек Ворона и Хопер была построена в начале XVIII века церковь, отсюда имя поселения. Здесь до сих пор не перевелись густые леса… Наверное, именно в таких местах рождается понимание красоты… И – огромное мальчишеское желание познавать мир. Ему потворствовал отец:

– Он много рассказывал в детстве, что такое звезды, как они «работают», почему дерево плавает, а железо тонет. Вещи банальные, но открывающие, как устроен мир.

Максим рано понял, что ответы на вопросы дает наука. Физикой заинтересовался в 9-м классе – участвовал в олимпиадах, даже на всероссийском уровне, занимал призовые места. Куда поступать? Конечно, в Московский физико-технический институт, где лучше всех учат физике.

– Может быть, один из факторов, почему я поступил именно в этот институт – когда-то мой папа тоже занимался физикой, участвовал в олимпиадах, но в последний момент не решился поехать поступать в Москву. Однако меня настраивал: МФТИ – это высокий уровень.

– Что самое интересное в науке? – спрашиваю я.

– Это возможность понимать вещи. Любое явление природы… Вот падает капля. Можно просто на нее смотреть, а можно попытаться понять, почему именно так она падает, какова форма капли, как она разбрызгивается, какую форму принимает на поверхности, куда упала. То есть взгляд на мир немного меняется.

Родители очень поддерживали Максима в студенческие годы, гордились им – сын всегда хорошо учился. Очень были рады, когда на последних курсах МФТИ Максим поехал на стажировку в Штаты. Пока был за океаном, начал прощупывать почву – хотел остаться в аспирантуре, уже почти договорился с одним профессором из Гарварда. Размышлял о том, чтобы осесть в Америке, вывезти туда дочку, близких… И вдруг – планы изменились. Максим отказался не только от перспективы жить в США, но и от профессии, которой учился в вузе. Такой поворот родителей огорчил, конечно, – получается, сын упустил шанс хорошо устроиться в жизни, заниматься серьезным делом – наукой… Они сейчас иногда спрашивают Максима: «Не жалеешь, что тогда не уехал в Америку?».

– Я каждый раз отвечаю, что не жалею. По-моему, они уже смирились, – улыбается «беглец».

А Москва… Она стала почти родной – столько лет уже здесь. Трудно представить, как можно работать не в столице, такое уж теперь у Максима Акимова дело – для него требуются масштабы и возможности мегаполиса.

О мудрости выбора

«Познает скорбь
Тот, кто читает их!» –
Так он сказал
И в пламя бросил книги.
Мудрец, ты сделал хорошо!

Исикава Такубоку

Ему не наскучило познавать мир. Просто Максим в какой-то момент понял, что он не хочет заниматься наукой. Его сферой была физика твердого тела. Конечно, он мог добиться успеха благодаря трудолюбию и упорству. Если бы оставалось желание идти по прежнему пути и в той же «компании». Однако уже на старших курсах Максим Акимов осознал – ему не по пути с теми, кто добивается успеха в науке. А главное:

– В тот момент люди делились для меня на два типа: одни работают с информацией – люди науки, и другие работают с другими людьми – занимаются продажами, маркетингом и так далее. На тот момент я «наелся» общением с людьми информации, мне казалось, что они живут в каком-то нереальном мире, построенном ими самими и имеющим довольно косвенное отношение к реальности. Мне захотелось в реальную жизнь.

И он пошел в реальную жизнь, где царят не фундаментальные законы и абстрактные теории, а цифры прибылей и убытков, графики роста производства, где каждый день можно видеть результаты своего труда.

Питомец МФТИ еще ни разу не пожалел о своем выборе… Даже когда выяснилось, что знания, полученные в вузе, в новой профессии «не работают»:

– Меня слегка разочаровывало, когда я после института начинал работать, пытался использовать продвинутые методы анализа и понял, что это все не нужно. В бизнесе важнее понимание процесса, чем некие зависимости, выверенные с помощью математических формул.

Однако Максим совершенно уверен – учился и стажировался в Штатах не зря!

– Я приехал в Москву из другого города. Учеба была тяжелая, и она дала мне навыки преодоления сложностей, преодоления себя. Заставить себя что-то делать, и делать много, те же самые экзамены сдавать – это закаляет. После такой «тренировки» очень многие вещи в жизни видятся проще, не вызывают страха.

Кстати, последние сомнения начинающего менеджера развеял его первый начальник, к которому Акимов пришел наниматься на работу:

– Это тоже было иностранное представительство. Я спросил: «Что лучше: если я пойду в аспирантуру учиться дальше или сразу пойду работать к вам?». А он спросил в ответ: «Как ты думаешь, кого я возьму с большей охотой – аспиранта, который ничего

не умеет в этой области, но обладает огромными амбициями, или голодного студента, выпускника института, у которого мозг еще не засорен всякой дребеденью и у которого есть желание учиться дальше?». Так я попал в свою первую компанию.

Карьера выстраивалась стремительно – помощник менеджера, channel-менеджер, ответственный за работу как в России, так и в Южной, Восточной Европе, за связи с вертикальными рынками.

Интересуюсь:

– Талант в сфере организации бизнеса открылся у вас внезапно или вы предполагали его в себе?

Максим улыбается и отвечает так, что сложно понять сразу, шутит он или говорит серьезно:

– Я думаю, что единственный мой талант, вернее, качество, которое способствует успеху... Я, наверное, все-таки умный. Мне так кажется. Не скажу, что я легко общаюсь с людьми. В социальном плане я человек довольно закрытый. Не могу сказать, что мне очень нравится постоянно общаться. Есть мера, которой мне достаточно. Но во время общения на работе я хорошо понимаю, что человеку надо.

– Я слушаю и осознаю – говорит серьезно, но, можно сказать, – с научной точки зрения. Ведь с чем я пришла к Максиму? С задачей проанализировать его собственную личность. И вот мне выдают результаты «ученого исследования»… И это «исследование» показывает, что сегодня Максим Акимов встал на новую ступеньку развития – стал руководителем.

– Я думаю, что я буду развиваться именно в этом направлении.

– А дальше?

– Посмотрим. Наверное, в конечном итоге это приведет к какому-то своему делу.

Это не личные планы, это – прогноз.

О, мудрый руководитель!

Есть радостная
легкая усталость,
когда, дыханья не переводя,
закончишь
трудную работу.

Когда приходится служить
капризным,
наглым самодурам,
как страшен
кажется весь мир!

Исикава Такубоку

Сегодня Максим Акимов отвечает за бизнес компании Kerio Technologies на ее крупнейшем рынке – в России. Он руководитель и уже не учится, а сам делится опытом с коллегами.

– Я впервые столкнулся с такой ситуацией… То есть я действительно понимаю, что у меня есть, что дать этой компании, причем не только в России, но и в глобальном масштабе. Это связано и с организацией, и с процессами внутри компании, и с направлениями развития. Работа в большой компании накладывает отпечаток, накапливается «багаж» умений – от того, как правильно планировать маркетинговые компании до организации внутренних бизнес-процессов, знаний – касательно скорости ответов, технологических приоритетов развития.

Работа в Kerio не могла не привлечь моего героя. И дело тут не в том, конечно, что Максиму Акимову хотелось почувствовать себя «боссом». Глава представительства, убежден он, – «должность, подразумевающая очень большой процент креатива и очень серьезную ответственность за все, что происходит».

На этот счет у Максима есть целая теория (сказывается все-таки научное прошлое):

– Есть две стороны. Одна сторона – личностные качества, напрямую не связанные с опытом работы. И здесь дело в той ответственности, которую человек хочет и может на себя брать. Потому что если ответственности не хватает, я, по крайней мере, начинаю добирать со стороны, смотрю, что можно еще сделать, чтобы почувствовать «кайф» работы. Ведь са-

мое большое удовольствие от работы состоит в том, что есть какое-то дело, я его делаю, и оно получается. Это одна часть. Вторая часть связана с профессиональными навыками. Как я говорил, опыт работы в большой компании дает очень много. В той компании нас довольно много учили. Моя работа связана в основном с общением с партнерами, с теми, кто продает наши решения. В предыдущей компании нас учили, как правильно общаться с такими людьми и как их заинтересовывать. Основная идея – channal-менеджер – человек, который общается с партнерами, должен быть не только продавцом своих решений, но и своего рода советником по бизнесу. Он должен хорошо разбираться в бизнесе, которым занимается компания. Топ-менеджер компании-партнера предпочитает общаться с channal-менеджером на одном языке. Если я приду и стану просто рассказывать, какие у нас хорошие продукты, партнер не будет слушать. Если же я начну беседовать с ним о его бизнесе, о том, каковы его приоритеты в развитии компании и как мы можем ему помочь, то это совсем другой подход. Отношение к нам будет совсем другое. Нас этому учили на курсах и на практике. И я стараюсь выступать, приходя к партнерам, не только как поставщик наших решений, но и как эксперт в нашей отрасли. Я им даю новые знания, информацию о том, что делают наши партнеры в других странах. После маржи, которую они имеют на наших продуктах, это наиболее полезная часть нашего сотрудничества – советы, взгляд со стороны на их бизнес. Я могу оценить, что они делают правильно, а что нет. И этот подход я стараюсь сейчас внедрить в нашей компании.

Став руководителем, Максим подошел к новой роли взвешенно и разумно. Он старается взять самое лучшее из увиденного – столько начальников прошло перед глазами, и у каждого было чему поучиться. Даже некий портрет «идеального» руководителя «нарисовался»:

– Это буфер между подчиненными и высшим менеджментом. То есть он как может защищает своих подчиненных и старается выбить максимально благоприятные условия для них. В то же время он требует по максимуму с подчиненных, чтобы они его не «подставили» перед высшим начальством. А вообще есть такая пословица, что лучшие руководители – это те, которых мы не замечаем. Могу добавить, что меня коробило, когда мои начальники не признавались в своих ошибках. Когда человек что-то делает неправильно и пытается вину свалить на своих подчиненных, это некрасиво…

Руководящий стиль самого Акимова во многом определен спецификой частной компании. Теплая, семейная обстановка, в которой просто не приживаются формализм и бюрократия, принятые в крупных фирмах с их неизбежным дресс-кодом и проверками. Пять минут опоздания – серьезная провинность… Нет, в российском представительстве Керио все весьма либерально, нет драконовских норм и законов. Однако в такой домашней обстановке важно не расслабиться до расхлябанности и ничегонеделания. Если ты с утра не в офисе, это не страшно, главное – чтобы работа была сделана вовремя, качественно и чтобы начальник мог с тобой в любой момент связаться.

И, кстати, такие же требования предъявляет к самому Максиму Акимову высшее руководство:

– Формальных требований нет. Свобода довольно большая. Позиция менеджмента такова: если человек ответственен за какой-то участок работы, что он там делает – это его дело, нужно только ставить в известность высшее начальство. … Согласитесь – разумно.

Кстати, а по поводу дресс-кода Максим не может забыть одну забавную историю:

– Когда я пришел в первый раз на собеседование в эту компанию, разговаривал с нашим старшим вице-президентом, я был в костюме, он – в футболке. Уже после собеседования он рассказал мне, что агентство, которое организовывало собеседование, спросило его, каков дресс-код для человека, который придет устраиваться на работу, и он ответил: «Главное, чтобы человек был одет». Культура компании такова, что люди не «напрягают» себя ненужными вещами. И в общении с партнерами это помогает.

Мудрый в горы не пойдет?

КАК ЗАВИДНА ИХ СУДЬБА!
К СЕВЕРУ ОТ СУЕТНОГО МИРА
ВИШНИ ЗАЦВЕЛИ В ГОРАХ.

ЧТО ГЛУПЕЙ ТЕМНОТЫ!
ХОТЕЛ СВЕТЛЯЧКА ПОЙМАТЬ Я –
И НАПОРОЛСЯ НА ШИП.

БАСЁ

Проследить маршрут человеческой судьбы – не так сложно. В общем-то можно и единичные поступки человека объяснить. А вот найти причины, докопаться, почему человек именно такой, какой он есть, сложно… Да и нужно ли это? Максим Акимов в свое время увлекался психологией, посещал различные тренинги, даже круг друзей составился именно из психологов – людей, которые близки по духу, по мировосприятию. Но объясняет ли это характер Максима, его личность? Наверное, только отчасти…

Хотя мне кажется, что интерес к проблеме информационной безопасности проистекает именно из любви к психологии. Сам Максим признает:

– По многочисленным оценкам, сейчас основные угрозы для предприятий в области информации проистекают от внутренних сотрудников, а не от всяких там хакеров, вирусов и так далее. Работники компаний сами заносят вирусы. Поэтому есть два кардинально противоположных пути, как вести себя в такой ситуации. Либо максимально контролировать своих сотрудников, установить слежку, прочитывать электронную почту. Либо, наоборот, демонстрировать полное доверие к людям, создавать условия, чтобы им было невыгодно вести себя неправильно или заниматься диверсиями... Очень много дебатов сейчас идет.

– А на ваш взгляд, как правильно?

– Я считаю, что должна быть какая-то золотая середина. Доверяй, но проверяй.

Да, с точки зрения сотрудников, слежка – это ужасно, но руководители считают, что это полезно, благодаря таким мерам они пресекают случаи кражи, утечки информации.

Разумный подход. Похоже, в случае Максима Акимова он распространяется на всю его жизнь. Если руководитель российского представительства компании Kerio отдыхает – то в горах: сноуборд зимой, пешие походы летом. Не только потому, что это ин-

тересно, красиво, но и потому, что эффективно.

– Я много раз сравнивал ощущения от отдыха в кресле у телевизора и в горах. В горах неделя – это месяц жизни, у телевизора неделя – это два дня. Голова тяжелая...

Такой же разумный подход к собственному здоровью:

– На тренажерах скучно. Стараюсь раз в два дня делать зарядку. Десять минут с утра. Это довольно прагматичная вещь. Мне трудно заниматься чем-то, что правильно в социальном аспекте, но не дает ничего лично мне. Я просто заметил, что если я десять минут с утра позанимаюсь, то становлюсь активнее, у меня настроение улучшается. После зарядки я себя лучше чувствую. Абсолютно прагматичный подход.

И совершенно в логике прагматичного поведения Акимова – его занятия на курсах скорочтения.

– Преподаватели курсов, на которых я сейчас занимаюсь, утверждают, что повысится не только скорость, но и качество чтения. Дают тексты, после них надо ответить на вопросы – такая примерно методика. Это надо – информации все больше, приходится много читать по менеджменту... В художественной литературе что нравится? «Тихий Дон» оставил сильное впечатление. Я его читал раза три в своей жизни, очень нравится. Фантастику тоже читаю – Хайнлайн, Кларк.

Театральные и кинопристрастия у Максима не тривиальные. Такой «человек разумный», как он, уж, конечно, не будет тратить время на «пафосные» походы по модным спектаклям и культовым фильмам.

– Я всегда любил театры небольшие и не очень классические. МХАТ им. Горького мне всегда казался мрачным. Островский повергал в какую-то депрессию… А вот театр «Сфера» небольшой и в то же время очень душевный. Более реальные персонажи в постановках. Сейчас реже получается выбираться, больше в кино хожу.

– А что в кино?

– Люблю корейские фильмы. Во-первых, они очень красивые, во-вторых, не такие предсказуемые, как американские. В корейском кино никогда не знаешь, что случится в следующий момент. Они показывают историю, которая может иметь десять смыслов, а может ни одного – смотришь как картинку. Гораздо больше эстетического удовольствия получаешь.

И уж, конечно, Максиму Акимову его повседневная деятельность вовсе не кажется фантастикой. Но я уже так привыкла спрашивать об этом у своих героев, что в очередной раз не удержалась…

Максим терпеливо объяснил свою точку зрения:

– Здесь есть два аспекта. Один – понимать, как оно устроено и как работает. И другое – это дело продавать. Я пытаюсь понять, что нужно нашим заказчикам, и понять, как мы подходим под их нужды. Фантастики тут особой нет. Хотя когда я читаю хронологию развития технологий и понимаю, как много было сделано, возникает такое ощущение. Но не часто. Я больше прагматик в этом плане.

Текст: Оксана Родионова,
фото: Евгения Тарабрина

JavaFX – Reach Internet Application от Sun

Прощай, унылый Swing?

Кирилл Сухов

5 декабря 2008 года Компания Sun Microsystems представила финальную версию JavaFX – свою платформу для создания Rich Internet Aplication, ставшую достойным ответом конкурентам.

Если вспомнить историю создания интернет-приложений, придется признать за компанией Sun первенство в создании Rich Internet Application. Первые Java-апплеты продемонстрированы Гослингом при презентации браузера WebRunner в далёком 1994 году. С тех пор прошло много всяких событий, интернет-приложения росли и изменялись, появилась технология Flash, использование клиентских возможностей браузера (Javascript, DOM) вылилось в термин WEB-2, а апплеты как технология, в общем, не сильно изменились. За-

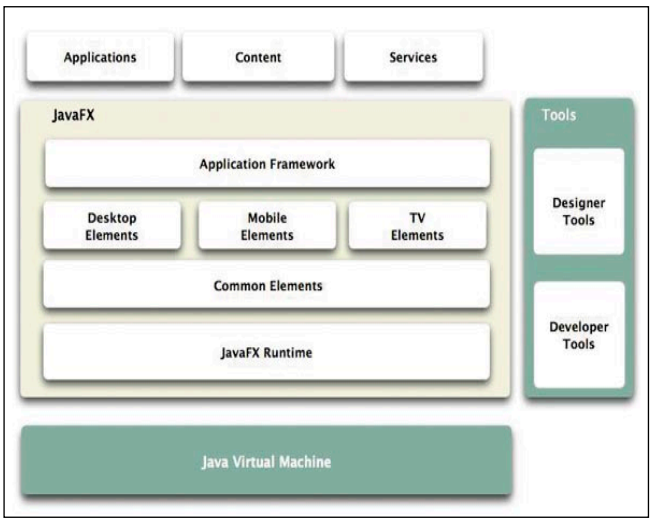


Рисунок 1. Архитектура платформы JavaFX

няв прочное положение на мобильных устройствах, на десктопах, они так и не получили заметного распространения. Положение осложнилось с появлением таких мощных конкурентных технологий, как Adobe Flex/Air, Microsoft Silverlight.

С другой стороны, имело место недовольство потребителей возможностями средств Java (Swing, Java2D), для разработки графического пользовательского интерфейса. Особенность ситуации была в том, что эти средства вполне позволяли в теории создавать GUI любой сложности и «красивости», но процесс этот никак нельзя было назвать лёгким и эффективным. Возникла необходимость в простом средстве проектирования интерфейсов, которым может быть скриптовый декларативный язык.

Явление JavaFX

Впервые технология была представлена на конференции JavaOne в мае 2007 года. В декабре 2008 года вышла JavaFX 1.0, включающая в себя средства разработки – JavaFX 1.0 SDK, плагин для NetBeans IDE 6.5 и JavaFX 1.0 Production Suite – набор инструментов для экспорта графических объектов в приложения JavaFX. Была представлена также бета-версия эмулятора JavaFX 1.0 Mobile, для разработки JavaFX-приложений для мобильных платформ. JavaFX TV – среда для запуска приложений на телевизионной платформе, планируется к запуску в начале 2010 года.

На настоящий момент (апрель 2009 года) вышла версия JavaFX 1.1, включающая мобильный профайл, позволяющий запускать JavaFX-приложения на мобильных устройствах.

На недавно открытом официальном сайте (JavaFX.com) можно видеть примеры готовых приложений на основе новой технологии, предоставленных с исходным кодом.

JavaFX потихоньку входит в нашу жизнь.

Что это?

Что конкретно представляет собой JavaFX-приложение? Оно вполне может работать как продукт Java web start, но основная его роль, декларируемая Sun – «апплет нового образца», основное отличие которого от обычных апплетов – возможность самостоятельного существования. В самом деле, JavaFX-программа вполне может быть оформлена в качестве обычного апплета, но этот апплет можно вытащить из содержащей его страницы, закрыть последнюю и продолжить работу (Drag-to-Install).

Рисунок 2. Здравствуйте

В общем, такой продукт является интернет-приложением, только потому, что через Глобальную сеть происходит доставка его потребителю, а также по причине наличия возможности активного взаимодействия через Интернет с сервером. Ну а в обычной несетевой, ипостаси эти приложения мало чем уступают обычным настольным.

Приложение JavaFX, естественно, использует для работы Java-машину (см. **рис 1**). Как следствие оно может использовать классы и объекты Java, импортировав соответствующие пакеты.

Технология позволяет легко встраивать в приложение мультимедиа данные, анимацию и различные визуальные эффекты. Связывание данных облегчает построение эффективных интерактивных интерфейсов.

Используемый скриптовый язык JavaFX Script понятен и прост в освоении (см. врезку «JavaFX Script – краткий обзор»). Он обладает обширными средствами для декларативного описания, имеет необходимые встроенные объекты.

Впрочем, не особенно хочется превращать обзор в рекламную листовку, лучше давайте попробуем возможности JavaFX на практике.

Начинаем работу

Для разработки на JavaFX существуют специальные инструменты, но сначала попробуем написать тестовое приложение, воспользовавшись обыкновенным текстовым редактором.

Прежде всего на вашей системе должен быть установлен пакет JDK (Java Development Kit) версии 5 или 6 (последняя по многим причинам предпочтительнее). Если это не сделано, скачиваем пакет по адресу http://java.sun.com/javase/downloads/index.jsp и устанавливаем его. Теперь скачиваем и устанавливаем JavaFX SDK (http://java.sun.com/javafx/downloads). Инсталляция не должна вызвать особых вопросов.

Для проверки работоспособности среды в консоли выполним команду javafxc, без параметров:

```
C:\Sun>javafxc
javafxc: no source files
Usage: javafxc <options> <source files>
use -help for a list of possible options

C:\Sun>
```

Если результат отличается, в первую очередь следует проверить, добавлен ли путь к JavaFX SDK в переменную окружения PATH.

Теперь напишем нашу первую программу. Создадим файл testfx.fx следующего содержания:

```
import javafx.stage.*;
```

```
import javafx.scene.*;
import javafx.scene.text.*;
import javafx.scene.paint.*;
import javafx.scene.effect.*;


Stage {
    title: "Hello FX"
    width: 250
    height: 80
    scene: Scene {
        content: Text {
            x: 10  y: 30
            font: Font { size: 24 }
            fill: Color.GREEN
            effect: DropShadow{ offsetX: 3 offsetY: 3}
            content: "Hello FX!"
        }
    }
}
```

Скомпилируем этот класс командой javafxc:

```
C:\Sun>javafxc testfx.fx
```

И запустим соответственно командой javafx:

```
C:\Sun>javafx testfx
```

Результат можно наблюдать на **рис. 2**.

Теперь давайте разберёмся, что мы тут натворили. С командами компиляции и запуска всё ясно – это прямые аналоги утилиты javac и команды java. Первые строчки кода также хорошо знакомы любому Java-программисту – это импорт необходимых классов из соответствующих пакетов.

Далее идёт то, что называется «декларативным синтаксисом». Что это такое? Если очень коротко, то особенность заключается в том, что программа не задаёт пошаговую инструкцию реализации алгоритма работы, а описывает объекты, полученные в виде конечного результата. При этом описываются свойства и поведение объектов, в том числе и интерактивное. Впрочем, об этом далее, в более сложных примерах.

Пока всё просто: объект Stage представляет собой окно приложения с соответствующими свойствами (title, width, height), значения которых определяются в последующих строчках. Далее внутри его описывается объект Stage – своего рода фрейм, содержащий другие объекты. Вернее, в данном случае один визуальный объект – Text, который также снабжен описанием своих свойств (положение, шрифт, цвет). Помимо их описания к тексту применён эффект тени, причём как шрифт, так и эффект также имеют свои описания.

Как видим, тут всё очень просто и интуитивно понятно. Впрочем, давайте попробуем написать маленькое, но действительно полезное приложение. Что нам нехватает в мире ERP-систем и SOA-приложений? Разумеется, красивого калькулятора! Вооружимся описанием языка и вперёд.

Сначала опишем калькулятор с индикатором:

```
import javafx.stage.*;
import javafx.scene.*;
import javafx.scene.text.*;
import javafx.scene.paint.*;
import javafx.scene.effect.*;
import javafx.scene.shape.*;

Stage {
    title: "JavaFX-калькулятор"
    width: 300
    height: 500


    scene: Scene {
        content:[ Text {
            x: 20  y: 60
            font: Font { size: 36}
            fill: Color.GREEN
            effect: DropShadow{ offsetX: 3 offsetY: 6}
      content: "2 * 2 = 4"
        }
]

}
}
```

Похоже на предыдущий пример. Мы импортировали необходимые классы, создали окно приложения, описали основной фрейм, создали там пока единственный элемент (в терминах JavaFX – нод) – Text. После текста добавим описание кнопки, используя встроенный объект Rectangle:

```
Rectangle {
                fill: Color.SILVER
                x: 10
                y: 80
                width:60
                height: 40
                arcWidth: 20
                arcHeight: 20

          }
```

И надписи на ней:

```
Text {
                x: 35
                y: 110
                font: Font { size: 24}
                         fill: Color.GREEN
                         content: "1"
      }
```

Попробуем скомпилировать и запустить результат приведёнными выше командами, и полюбуемся чудным однокнопочным калькулятором (см. **рис. 3**).

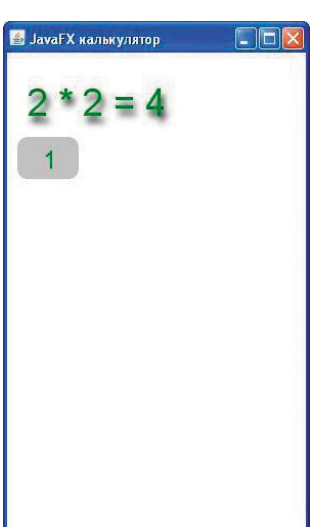


Рисунок 3. Для калькулятора маловато кнопок

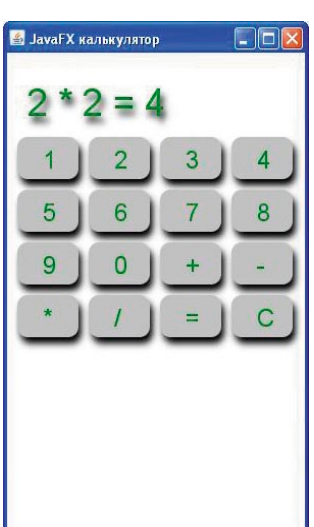


Рисунок 4. Теперь кнопок достаточно

Ну дальше всё просто – описываем одну за другой все остальные кнопки... Что? Объекты? Массивы? Всё так, и хоть мне и платят за количество печатных знаков, я всё-таки программист и не могу не унифицировать процесс. Для начала создадим следующий класс:

```
class CalcBatton{
var cx: Number;
var cy: Number ;
var ctext: String;
var group= Group{
    content: [
     Rectangle {
              fill: Color.SILVER
              x: cx
              y: cy
              width:60
              height: 40
              arcWidth: 20
              arcHeight: 20
              cursor: HAND
              effect: DropShadow{ offsetX: 3 offsetY: 6}
               },
     Text {
              x: cx + 25
              y: cy + 30
              font: Font { size: 24}
              fill: Color.GREEN
              content: ctext
           }]}
}
```

В данном случае мы объединили в одну группу кнопку и надпись на ней, создав класс, экземплярами которого будут кнопки калькулятора. Переменные класса устанавливают надпись на кнопке и её абсолютное расположение (вообще говоря, в JavaFX присутствуют менеджеры размещения, но для простого приложения с фиксированным числом компонентов можно ограничиться и абсолютным позиционированием.)

Теперь создадим массив кнопок:

```
def bt1=CalcBatton{ cx: 10, cy: 80,ctext: "1"}
def bt2=CalcBatton{ cx: 80, cy: 80,ctext: "2"}
def bt3=CalcBatton{ cx: 150, cy: 80,ctext: "3"}

...

def bt9=CalcBatton{ cx: 10, cy: 180,ctext: "9"}
def bt0=CalcBatton{ cx: 80, cy: 180,ctext: "0"}

def bt_add=CalcBatton{ cx: 150, cy: 180,ctext: "+"}
def bt_sub= CalcBatton{ cx: 220, cy: 180,ctext: "-"}
def bt_mul=CalcBatton{ cx: 10, cy: 230,ctext: "*"}
def bt_div=CalcBatton{ cx: 80, cy: 230,ctext: "/"}
def bt_res=CalcBatton{ cx: 150, cy: 230,ctext: "="}
def bt_del=CalcBatton{ cx: 220, cy: 230,ctext: "C"}
var buttons=[bt1.group,bt2.group,bt3.group,
              bt4.group,bt5.group,bt6.group,
              bt7.group,bt8.group,bt9.group,
              bt0.group,bt_add.group,bt_sub.group,
              bt_mul.group,bt_div.group,bt_res.group,
              bt_del.group];
```

И разместим этот массив на «сцене» приложения:

```
Stage {
   ...
    scene: Scene {
        content:[ Text {
            ...
        },buttons
]}
}
```

После компиляции должна получиться картинка, сходная с **рис. 4**.

Не останавливаемся. Нам нужно рабочее приложение, следовательно, после нажатия на клавиши текст в индикаторе должен меняться. Для этого сначала введём глобальную переменную, которая содержит текст индикатора, и немного изменим его код:

```
var cont="";
class CalcBatton{
…..................................
effect: DropShadow{ offsetX: 3 offsetY: 6}
      content: bind cont
        }
```

В данном случае мы воспользовались ключевой возможностью JavaFX Script – связыванием данных (data binding). Теперь любое изменение переменной cont сразу же будет отображено на индикаторе калькулятора. Теперь внесём изменения в код кнопки, дабы описать её реакцию на клик мышкой:

```
effect: DropShadow{ offsetX: 3 offsetY: 6}
onMouseClicked: function(evt: MouseEvent):Void {
              but="{but}{ctext}";
 if({ctext}=="C"){
              but="";
              }
}
}
```

Как видно из кода, при нажатии на клавишу <C> происходит очистка индикатора. Конечно, следует также прописать реакцию на нажатие <=>, вызывающее вычисление результата, но это я предоставляю для реализации читателю. Сама по себе задача вполне ординарная. Мы же тут заняты другим – стараемся «сделать красиво». Продолжая действовать в этом направлении, «оживим» кнопки, заставим их нажиматься. Для этого введём ещё два атрибута нашего класса:

```
var gradx= 3;
var grady= 6;
var group= Group{
```

Эти переменные будут отвечать за размеры эффекта тени, при изменении которых и создаётся эффект нажатия:

```
cursor: HAND
      effect: DropShadow{ offsetX: bind gradx offsetY: ↵
          bind grady}
```

Как видите, мы опять используем связывание. И связываем мы их с типичными действиями пользователя. Пишем в коде объекта Rectangle:

```
onMousePressed:    function(evt: MouseEvent):Void {
                         gradx=0;
                         grady=0;
                         }

onMouseReleased:  function(evt: MouseEvent):Void {
                         gradx=3;
                         grady=6;
                         }
                  },
```

Осталось только навести ещё немного гламура (не забывая, кстати, импортировать необходимые классы), и калькулятор готов. Полный код (без вычислительных функций) приведён ниже:

```
import javafx.stage.*;
import javafx.scene.*;
import javafx.scene.text.*;
import javafx.scene.paint.*;
import javafx.scene.effect.*;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.Cursor.*;
import javafx.scene.Node.*;
import javafx.scene.input.MouseEvent;

class Calc{
// здесь «вычислительная» часть
}

def cl= Calc{};
def fil_bt=LinearGradient {
    startX: 0.0, startY: 0.0, endX: 0.0, endY: 1.0, ↵
     proportional: true
    stops: [
     Stop {offset: 0.0 color: Color.WHITE},
     Stop {offset: 1.0 color: Color.SILVER}
    ]
    }
var cont="";
var sum=0;
class CalcBatton{
var cx: Number ;
var cy: Number ;
var ctext: String;
var gradx= 3;
var grady= 6;
var group= Group{
    content: [
     Rectangle {
         fill: fil_bt
         x: cx
         y: cy
         width:60
         height: 40
         arcWidth: 20
         arcHeight: 20
         cursor: HAND
         effect: DropShadow{ offsetX: ↵
              bind gradx offsetY: bind grady}
         onMouseClicked: function(evt: MouseEvent):Void {
          if({ctext}=="C"){
           but="";
          }
         else{
           but="{cont}{ctext}";
          }
         }
         onMousePressed:  function(evt: MouseEvent):Void {
         gradx=0;
         grady=0;
         }
         onMouseReleased:
         function(evt: MouseEvent):Void {
         gradx=3;
         grady=6;
         }
      },
         Text {
         x: cx + 25
         y: cy + 30
         font: Font { size: 24}
                       fill: Color.GREEN
                       content: ctext
         }]}

}
def bt1=CalcBatton{ cx: 10, cy: 80,ctext: "1"}
def bt2=CalcBatton{ cx: 80, cy: 80,ctext: "2"}
def bt3=CalcBatton{ cx: 150, cy: 80,ctext: "3"}
def bt4=CalcBatton{ cx: 220, cy: 80,ctext: "4"}
def bt5=CalcBatton{ cx: 10, cy: 130,ctext: "5"}
def bt6=CalcBatton{ cx: 80, cy: 130,ctext: "6"}
def bt7=CalcBatton{ cx: 150, cy: 130,ctext: "7"}
def bt8=CalcBatton{ cx: 220, cy: 130,ctext: "8"}
def bt9=CalcBatton{ cx: 10, cy: 180,ctext: "9"}
def bt0=CalcBatton{ cx: 80, cy: 180,ctext: "0"}
def bt_add=CalcBatton{ cx: 150, cy: 180,ctext: "+"}
def bt_sub=CalcBatton{ cx: 220, cy: 180,ctext: "-"}
def bt_mul=CalcBatton{ cx: 10, cy: 230,ctext: "*"}
def bt_div=CalcBatton{ cx: 80, cy: 230,ctext: "/"}
def bt_res=CalcBatton{ cx: 150, cy: 230,ctext: "="}
def bt_del=CalcBatton{ cx: 220, cy: 230,ctext: "C"}
var buttons=[bt1.group,bt2.group,bt3.group,
             bt4.group,bt5.group,bt6.group,
             bt7.group,bt8.group,bt9.group,
             bt0.group,bt_add.group,bt_sub.group,
             bt_mul.group,bt_div.group,bt_res.group,
             bt_del.group];
Stage {
    title: "FX Калькулятор"
    width: 300
    height: 500
    scene: Scene {
        content:[ Text {
            x: 20  y: 60
            font: Font { size: 36}
            fill: Color.GREEN
            effect: DropShadow{ offsetX: 3 offsetY: 6}
      content: bind cont
        },buttons
]
      fill: LinearGradient {
      startX: 0.0, startY: 0.0, endX: 0.0, endY: 1.0, ↵
              proportional: true
      stops: [
        Stop {offset: 0.0 color: Color.WHITE},
        Stop {offset: 1.0 color: Color.LIGHTGREEN}
      ]
      }
    }
}
```

Результат – на **рис. 5**.

Что ещё?

Какие возможности JavaFX остались за рамками этого небольшого примера? Прямо скажу, немало. Начнём с того, что в палитре среды присутствует множество графических примитивов. Это геометрические фигуры (вроде используемого нами Rectangle), градиенты (линейный градиент мы использовали для заливки калькулятора и кнопок), различные графические эффекты. JavaFX позволяет работать с готовыми изображениями, используя их в сценах и применяя к ним различные фотоэффекты, как-то затенение, смещение, размытость, трансформация и так далее, причём применять в любой комбинации.

Средствами JavaFX легко достигается анимация объектов (каюсь, хотел включить эту важную функциональность в наш калькулятор, но вовремя остановился) и встраивается видео. Вот как просто это делается:

```
Stage {
      title: "My Media Player"
      scene: Scene{
        content: MediaComponent {
          mediaSourceURL: mediaUrl
          volume: 0.5
          visible:true
          mediaPlayerAutoPlay: true
        }
        }
}
```

Поддерживаются все форматы, которые поддерживает платформа, на которой исполняется приложение, при этом используются «родные» кодеки этой платформы. Если не находится нужного кодека, используется встроенный кроссплатформенный кодек On2 от компании On2 Technologies.

Разумеется, всё вышеперечисленное можно применять в различных комбинациях. Примеры применения с исходным кодом можно найти на сайте технологии JavaFX – http://www.javafx.com/samples.

Рисунок 5. Приложение готово и даже работает

На **рис. 6** показан один из них – вращающийся трёхмерный куб, каждая грань которого работает как видеопроигрыватель, в котором можно открыть отдельный видеофрагмент.

Разумеется у средства для создания RIA есть инструменты для построения именно интернет-приложений.

Прежде всего это объект javafx.io.http.HttpRequest, предоставляющий API для асинхронного взаимодействия с HTTP-сервером (да, да, тот самый AJAX). Очень полезным представляется javafx.data.pull.PullParser многопоточный парсер документов, с возможностью задания функции обратного вызова (callback) в качестве параметра.

Пример из руководства:

```
var total;
var title;
def parser = PullParser {
    documentType: PullParser.XML;
    input: anInputStreamThatContainsXML;
    onEvent: function(event: Event) {
      if (event.type == PullParser.START_ELEMENT) {
          if (event.qname.name == "ResultSet" ↵
               and event.level == 0) {
              total = event.getAttributeValue ↵
                (QName{name:"totalResultsAvailable"});
          }
      } else if (event.type == PullParser.END_ELEMENT) {
          if (event.qname.name == "Title" ↵
               and event.level == 2) {
              title = event.text;
          }
      }
    }
}
parser.parse();
parser.input.close();
println("results: {total}, title: {title}");
```

Предусмотрена также работа с RESTFul Web Services (технология Representational State Transfer), встроенные парсеры XML и JSON, интеграция с JavaScript и технологией LiveConnect.

JavaFX Script – краткий обзор

В небольшом приложении я попытаюсь дать краткий обзор языку JavaFX Script, ровно до такой степени, чтобы читатель разобрался с кодом примера.

Следует сразу оговориться, что JavaFX Script – это не Java, и вообще он имеет с Java довольно мало общего. Вторая оговорка – пусть вас не смущает слово Script в его названии. Он не является языком сценариев в том смысле, в котором им является, например JavaScript, ничего общего он не имеет с такими интерпретируемыми «скриптовыми» языками, как PHP или Ruby. Прежде всего это декларативный язык. На практике это обозначает, что для построения приложения мы не описываем объекты и их методы, а определяем объекты сцены, определяем их атрибуты (в том числе поведение, реакцию на события). Но всё по порядку, сначала основы синтаксиса.

Основные типы данных

JavaFX имеет статическую типизацию – тип переменной (они задаются с помощью ключевого слова var) определяется при её инициализации:

```
var name;
var age;
name="John";
age=35
```

Константы задаются с помощью ключевого слова def. Правила наименований переменных и констант схожи с языком Java.

Это же касается и наименований функций. Сами они задаются так:

```
function addTwo(a:Number):Number{
return a+2;
}

println(addTwo(4));
```

Как видим, разительное отличие синтаксиса Java, заключается в указании типа возвращаемого значения не до, а после объявления функции. То же и с её аргументами. Возможно, причины такого синтаксического финта кроются в том, что продукт создавался с оглядкой на Adobe Flex. Впрочем, бог с ними, с грязными домыслами.

В языке присутствуют встроенные типы данных – String, Number, Integer, Boolean и Duration, которые также являются и объектами типов.

При определении типа String могут использоваться как двойные, так и одинарные кавычки. Для подстановки переменных или выражения внутри строки используются фигурные скобки:

```
var name='john';
println(«Hello {name}!»);
```

Типы Integer Number, предназначены для представления целочисленных значений и числа с плавающей точкой соответственно. Boolean, как несложно догадаться, может принимать значения true или false. Он используется в управляющих конструкциях (if, else), которые имеют традиционный для Java синтаксис.

Тип Duration используется для представления временных отрезков, например:

```
var a = 10s; // 10 секунд
```

Имеется и знакомый по Java «пустой» тип Void (регистр – не опечатка), впрочем, при объявлении функции, не возвращающей значение, его можно опустить.

Для перечисляемых данных предусмотрен тип последовательности (sequencesy), обладающий необходимыми методами:

```
var colors: String[] = ["Green", ↵
     "Red","Gray"];
insert "Black" into colors;
insert "Blue" after colors[1];

// теперь последовательность такая:
// ["Green","Red","Blue","Gray",
// "Black"];

delete "Gray" from colors;
delete  colors[2];

// теперь:
// ["Green","Red",Black"];

var nums = [1..100];
```

В последнем случае генерируется последовательность из целых чисел, от 1 до 100.

Классы и объекты

Классы в JavaFX определяются следующим образом:

```
class myObject{
     var name: String;
     var city: String;
     var myRectangle: Rectangle;

    function print_name() {
        println("Name: {name}");
    }
```

Вопросы тут может вызвать только тип данных Rectangle. Это один из многочисленных встроенных классов JavaXF, просмотреть его свойства и методы можно

Среда разработки

Такой пример, как наш калькулятор, вполне можно написать в простом текстовом редакторе, но современный процесс разработки (несмотря на мнение отдельных гиков) требует удобной и эффективной IDE, с возможностью отладки, профайлинга, рефакторинга и управления проектами, а также желательно с интеграцией с системой контроля версий и многим другим. И тут хорошие новости для программистов, привыкших работать в среде NetBeans. Начиная с версии 6.5 при установке соответствующего плагина теперь можно работать с JavaFX-проектами с неменьшей эффективностью, что и с Java, используя дизайнер форм и прочие инструменты IDE. Приверженцы другой популярной среды разработки, Eclipse, не остались обижены, плагин для работы с JavaFX доступен по адресу http://kenai.com/projects/eplugin.

Впрочем, это всё для программистов, для дизайнеров же подготовлен набор инструментов под общим названием JavaFX Production Suite. Это плагины для пакетов Adobe Photoshop и Adobe Illustrator, позволяющие сохранять многослойные приложения в формате JavaFX (Save as JavaFX). Он включает инструмент JavaFX Graphics Viewer, который позволяет использовать графику без компиляции в байт-код, и SVG Converter, позволяющий сохранять графику в формате JavaFX.

После знакомства с возможностями JavaFX, естественно, встаёт вопрос: каковы её перспективы? Какое место она может занять в быстро меняющемся ИТ-мире? Сейчас на этот вопрос, наверное, никто ответить не может. Но в целом платформа, имеющая все ключевые средства для построения RIA, с возможностью использования накопленных Java-наработок и уже сейчас имеющая вполне современные средства разработки, выглядит очень многообещающе.

Рисунок 6. В каждой грани независимый видеофрагмент (если хватит оперативки)

1. Официальный сайт JavaFX – http://javafx.com.
2. Материалы по JavaFX на сайте российских разработчиков Sun Microsystems – http://developers.sun.ru/javafx.
3. Русскоязычная wiki по JavaFX – http://ru.jfx.wikia.com/wiki.

в документации по API JavaXF (http://java.sun.com/javafx/1.1/docs/api).

Объекты описываются так:

```
def myObj = myObject {
     name: "Vasya";
     city: "Moscow";
     number:5;
}
def myRectangle = Rectangle {
        fill: Color.GREEN
        width:60
        height: 40
}
```

Первый из вышеприведённых объектов является экземпляром произвольного пользовательского класса, второй – встроенного класса Rectangle. Вполне допустимы и композиции объектов:

```
def myObj=myObject {
     name: "Vasya";
     city: "Moscow";
     number: "95050";
          myRectangle: Rectangle {
             fill: Color.GREEN
                width:60
                height: 40
              }
}
```

Обращение к свойствам и методам объекта происходит посредством точечной нотации:

```
println( myObj.myRectangle.x);
println( myObj.=print_name());
```

Такие необходимые элементы ООП, как наследование и модификаторы доступа, также наличествуют.

Связывание данных

Ключевой элемент языка JavaFX Script – возможность динамической привязки данных (data binding). Это возможность связать любой параметр с динамически изменяющейся переменной.

Лучше всего применение связывания видно на примере:

```
var side= 15;

Scene {
    content: [
        SwingSlider {
          minimum: 0
          maximum: 50
          value: side radius ↵
               with inverse
          vertical: false
        },

        Rectangle {
          fill: Color.GREEN

          width: bind side
          height: bund side*2
          }
    ]
}
```

В данном случае объект SwingSlider (стандартный скрол бар) определяет значение переменной side, с которой, в свою очередь, связаны размеры сторон объекта Rectangle (прямоугольник). Таким образом, пользователь может менять размеры фигуры, перемещая указатель скрола.

Ещё один интересный механизм JavaFX – триггеры, то есть конструкции, представляющие собой блоки кода, исполняемые по какому-либо событию. Пример того, как привязать блок кода к значению переменной:

```
var User = "Misha" on replace ↵
      oldValue {
     println("Пользователь ↵
          изменился");
     println("был: {oldValue}");
          println("стал: {User}");
};
```

Такой код будет служить индикатором, печатая сообщения всякий раз при изменении значения переменной User.

На сцене

В основе любой программы на JavaFX находится объект Stage, представляющий собой древовидную структуру, содержащую в себе другие объекты приложения. Каждый узел такой системы, представляющий собой наследника от класса javafx.scene.Node, может также содержать в себе объекты.

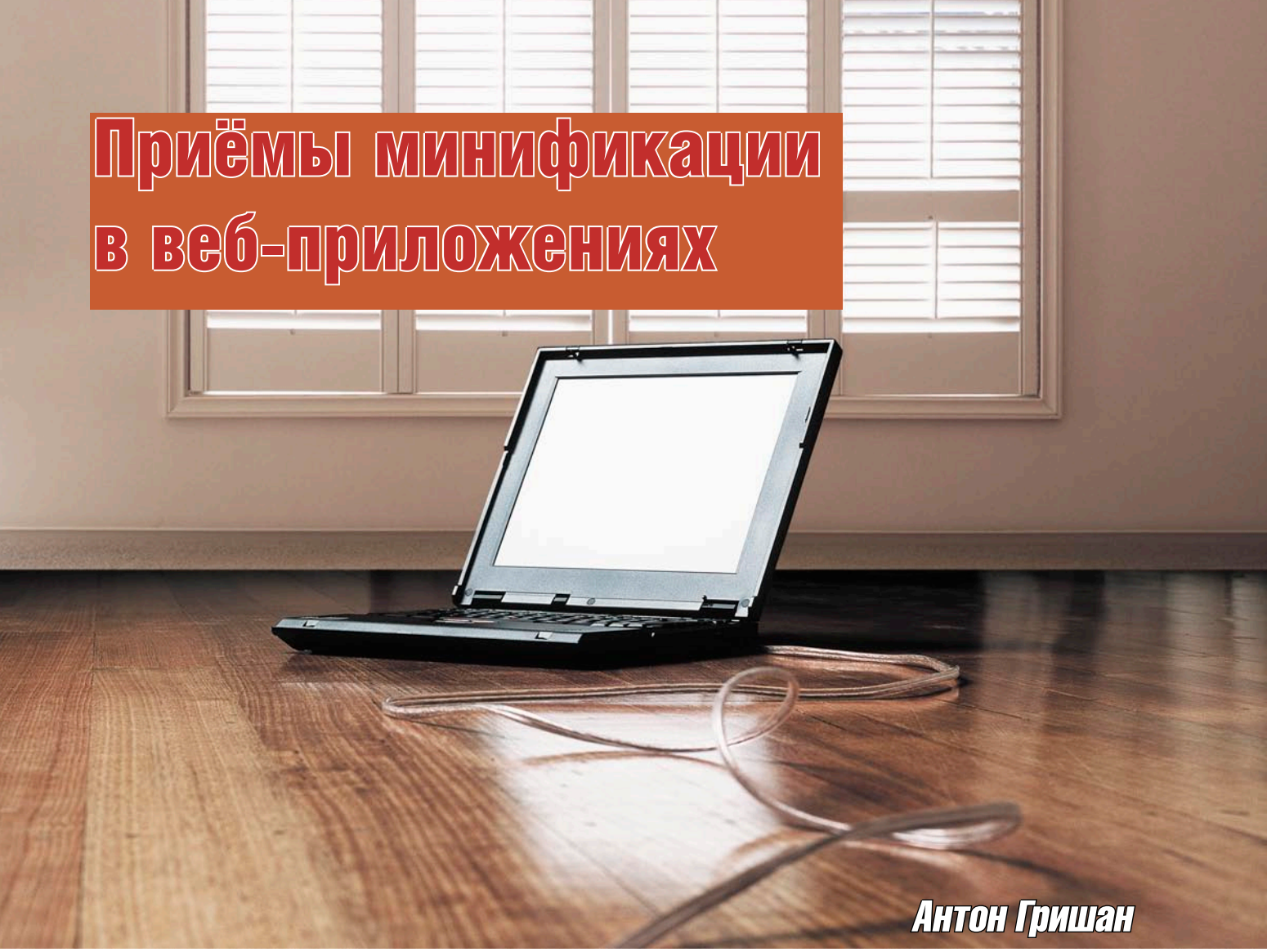
Класс Node является полноценным контейнером, характеризующийся размерами, системой координат и реагирующим на события мыши. Объекты, от него образованные, можно трансформировать, объединять в группы, применять к ним (как к отдельным, так и к группам) различные действия или графические эффекты.

Доступ к содержимому «Сцены» производится через объект Scene, для добавления на нее необходимых объектов, следует импортировать соответствующие классы и описать объекты и их поведение внутри Scene.

Дальше – смотрите пример, приведённый в статье.

Подробные уроки по языку JavaFX Script, на русском языке доступны на портале разработчиков Sun. По адресу http://developers.sun.ru/documents/javafx/FXTutorials.

Приёмы минификации в веб-приложениях



Антон Гришан

Любой веб-мастер желает, чтобы его сайт загружался быстрее, чем сайт конкурента, поэтому оптимизация – важный аспект разработки любого веб-проекта.

Понятие оптимизации охватывает огромное количество вопросов, начиная от настройки веб-сервера, заканчивая алгоритмами обработки данных. Одно из направлений оптимизации – минификация. Здесь и далее под термином «минификация» будем понимать вид оптимизации, направленной на уменьшение размеров результирующего HTML/CSS/JS-кода без изменения внешнего вида и функционала веб-страницы.

Минификация HTML-кода

Результатом работы большинства веб-приложений является HTML-документ. Чем короче код, тем быстрее его загрузит клиент. Рассмотрим несколько приемов, позволяющих минимизировать размер результирующего кода.

Верстка страниц без использования таблиц (Tableless)

Мощное средство минификации HTML-кода – верстка страниц на базе DIV-тегов и таблиц стилей (CSS). В отличие от табличной верстки данная методология имеет множество преимуществ, одно из которых – существенное сокращение размера HTML-кода.

Табличная верстка без использования стилей:

```
<html>
    <head>
      <title>Пример: табличная вёрстка vs DIV-ная ↵
          вёрстка</title>
    </head>
    <body>
      <table width="720px" border="0" align="center" ↵
          cellpadding="0" cellspacing="0" ↵
          style="margin-top:30px">
            <tr>
              <td colspan="3" align="center">Шапка ↵
                  сайта, логотип</td>
            </tr>
            <tr valign="top">
              <td width="200px">Левая колонка</td>
              <td width="320px">Центральная колонка</td>
              <td width="200px">Правая колонка</td>
            </tr>
      </table>
    </body>
</html>
```

Размер кода: 482 байта.

Верстка на базе DIV-тегов с использованием стилей:

```
<html>
    <head>
      <title>Пример: табличная вёрстка vs DIV-ная ↵
```

```
        вёрстка</title>
      <link rel="stylesheet" type="text/css" ↵
        href="/main.css" />
    </head>
    <body>
      <div id="head">Шапка сайта, логотип</div>
      <div id="content">
            <div id="cLeft">Левая колонка</div>
            <div id="cCenter">Центральная колонка</div>
            <div id="cRight">Правая колонка</div>
      </div>
    </body>
</html>
```

Размер кода: 384 байта.

Если сравнить размер среднестатистической HTML-страницы, сверстанной с помощью таблиц, и аналогичной страницы, сверстанной на базе DIV-элементов и каскадных таблиц стилей, можно заметить, что размер первого варианта не сильно отличается от второго, если ко второму добавить размер таблицы стилей. Однако таблица стилей в отличие от содержания страницы статична и кэшируется браузером, поэтому загружается единственный раз, при первом обращении к сайту. Таким образом, при каждой последующей загрузке страницы сайта экономия трафика (и времени) будет сопоставима с размером таблицы стилей.

Вынос JavaScript и CSS в отдельные файлы

Большинство современных веб-приложений для построения интерактивного интерфейса использует множество таблиц стилей и JavaScript. Существует два способа подключить JS/CSS-код к HTML-документу:

1. Включение JS/CSS-кода в тело HTML-документа.

«За»:

- не требует отдельного HTTP-запроса для загрузки JS/CSS-кода;
- возможность динамической генерации JS/CSS-кода.

«Против»:

- размещение JS/CSS-кода в теле документа увеличивает размер HTML-кода;
- не используется механизм кэширования, поэтому каждый раз при запросе документа пользователь будет повторно загружать JS/CSS-код.

2. Подключение внешних файлов, содержащих JS/CSS-код.

«За»:

- сокращение размера HTML-документа;
- при повторном обращении к документу браузер использует кэш, что позволяет избежать повторной загрузки JS/CSS-кода.

«Против»:

- осложняет динамическую генерацию JS/CSS-кода;
- требуется HTTP-запрос для загрузки каждого JS/CSS-файла.

Оба варианта обладают своими достоинствами и недостатками. Включение JS/CSS-кода в тело документа имеет смысл, если код генерируется динамически. В остальных случаях вынос кода во внешние файлы положительно сказывается на скорости загрузки веб-документа за счет использования кэша браузера и приемов минимизации HTTP-запросов.

Удаление символов, не влияющих на отображение документа

HTML-код содержит множество символов, не влияющих на отображение документа в браузере пользователя:

- символы перевода строки (\r и \n), табуляции (\t);
- несколько идущих подряд пробелов, пробелы между атрибутами тегов;
- необязательные кавычки вокруг значения атрибута тега;
- HTML-комментарии <!-- -->.

Наличие табуляций, переносов строки и комментариев делает код удобочитаемым для человека, но не влияет на отображение страницы в браузере и индексацию сайта поисковыми системами.

Удаление символов, не влияющих на отображение документа, позволяет сократить размер HTML кода на 3-20%. Данным приемом активно пользуются такие титаны, как Google.com и Yandex.ru (убедиться в этом можно посмотрев HTML-код главной страницы поисковиков).

Минификация HTML-кода с помощью PHP

Рассмотрим пример простейшего минификтора HTML-кода, построенного на базе PHP. Принцип работы заключается в удалении из HTML-кода символов \n \r \t и незначащих пробелов.

```
<?php
function htmlCompressor($html) {
    return preg_replace('/(?:\n\s*|\s\s+)/i', '', $html);
}
ob_start("htmlCompressor");
?>
<html>
    <head>
      <title>Example:   simple html compressor</title>
    </head>
    <body>
      <h1>Hello World!!!</h1>
      HTML document here
    </body>
</html>
<?
ob_end_flush();
?>
```

Результатом исполнения данного скрипта будет HTML-код, имеющий вид:

```
<html><head><title>Example: simple html compressor ↵
      </title></head><body><h1>Hello World!!!</h1> ↵
      HTML document here</body></html>
```

В примере объявлена функция htmlCompressor, отвечающая за сжатие HTML-кода, переданного в аргументе. Используем ob_start() для буферизации результата работы приложения. Перед отправкой пользователю результатов работы приложения минимизируем с помощью функции htmlCompressor() хранящийся в буфере HTML-код.

Протестируем работоспособность метода, применив компрессор к HTML-коду различных сайтов (см. **таблицу 1**).

Из результатов теста видно, что степень сжатия зависит от стиля кодирования. В среднем использование простейшего HTML-минификатора позволяет сократить размер до-

Таблица 1. Результаты применения компрессора к HTML-коду различных сайтов

Адрес страницы	Размер до сжатия (байт)	Размер после сжатия (байт)	Компрессия
http://vkontakte.ru	7129	6689	6.17 %
http://whatismyip.com	10795	9749	9.69 %
http://torrents.ru/forum/index.php	140935	117007	16.98 %
http://anekdot.ru	29619	26543	10.39 %
http://lenta.ru	112099	107856	3.79 %
		Среднее сжатие:	9.40%

кумента на 9,4%. Остается множество резервов для увеличения степени сжатия.

- Удаление комментариев из HTML-кода.
- Удаление необязательных пробелов внутри тега (например: <div title="hello world"class="menu"id="mainMenu">).

Используя оставшиеся резервы, можно увеличить среднюю степень сжатия примерно до 19% (этот параметр зависит от стиля кодирования документа). Некоторые алгоритмы минификации HTML удаляют необязательные кавычки вокруг значения атрибута тега, например, так:

```
<div title="hello world" class=menu id=mainMenu>
```

Данная методика допустима для HTML-документа, так как не противоречит стандарту, но крайне нежелательна для XHTML. С точки зрения стандарта XML, кавычки вокруг значения атрибута тега обязательны, поэтому удаление их чревато не только появлением ошибок при проверке документа с помощью W3C-валидатора (http://validator.w3c.org), но и возникновением потенциальных проблем при использовании DHTML (нарушение структуры документа).

Корректное удаление комментариев из HTML-документа не имеет побочных эффектов, однако для корректного выполнения требуется учитывать ряд специфических случаев, когда удалять строку, которая похожа на комментарий, нельзя.

Пример:

```
<!--[if lte IE 6]><style type="text/css" media="all"> ↵
      @import url(http://www.domain.tld/css/ie6.css); ↵
      </style><![endif]-->

<script language="JavaScript">
      <!--
      alert('<!--');
      alert('Меня удалять нельзя!');
alert('//-->');
      //-->
</script>
```

Можно разработать алгоритм, позволяющий корректно обрабатывать все возможные случаи, однако он будет достаточно сложен и потребует от сервера дополнительных ресурсов (процессорное время и память). Вызов алгоритма минификации HTML осуществляется при каждом запросе страницы, потому при разработке алгоритма необходимо найти баланс между степенью сжатия и нагрузкой на сервер.

Минификация скриптов и таблиц стилей

Большинство разработчиков не уделяют внимания оптимизации внешних JS/CSS-файлов, так как эти файлы загружаются один раз и при повторном запросе подключаются из кэша браузера.

Однако высокая скорость первой загрузки сайта дает существенное преимущество, потому что многие пользователи, просматривая результаты выдачи поисковых систем, открывают найденные страницы во множестве вкладок (окон) браузера и знакомятся с информацией по мере загрузки сайтов. Поэтому если сайт загружается быстрее, чем сайт конкурентов, то вероятность того, что вы получите нового клиента, существенно повышается. Минификация JS/CSS позволяет существенно увеличить скорость первой загрузки сайта.

Минифицированный код абсолютно нечитаемый, поэтому разумно иметь две версии:

- **Версия для разработчиков.** Код содержит все необходимые комментарии и визуальную разметку, что обеспечивает читабельность. Используется в момент разработки приложения.
- **Версия для публикации.** Код имеет минимальную длину, что обеспечивает высокую скорость загрузки.

Рассмотрим несколько популярных утилит, позволяющих осуществить автоматический перевод JS/CSS-кода из версии для разработчиков в версию для публикации.

JavaScript

В большинстве своём минификаторы для уменьшения размера JavaScript кода делают две операции:

- удаляют символы, не влияющие на исполнение кода;
- заменяют длинные названия локальных переменных на более короткие.

Протестируем работу наиболее известных JS-минификаторов на примере большого JavaScript-файла (URLForward.js, размер 61773 байт), взятого из реального проекта (см. **таблицу 2**).

Рекордный уровень компрессии показал минификатор Bananascript. Такой большой отрыв от конкурентов объясняется тем, что помимо стандартного алгоритма минификации JS-кода разработчики данного сервиса добавили алгоритм компрессии, основанный на замене длинных повторяющихся последовательностей байтов.

Однако Bananascript не является оптимальным выбором. Сервер перед отдачей JS-кода пользователю упакует данные с помощью gzip. Поэтому важно, чтобы получившийся в результате минимизации и упаковки код имел минимальную длину. Проведем еще один тест, в котором сравним лидирующие в предыдущем тесте минификаторы с учетом длины кода после упаковки с помощью gzip (см. **таблицу 3**).

Из последнего теста видно, что наименьший размер кода получается в результате использования связки Packer + gzip, затем идет YUI Compressor + gzip, и на последнем месте Bananascript + gzip.

В первом тесте лучший результат продемонстрировал Bananascript, он же продемонстрировал худший результат

во втором. Данное явление можно объяснить тем, что Bananascript уже содержит алгоритм компрессии, поэтому повторное сжатие малоэффективно.

Таким образом, лучший результат продемонстрировал Packer + gzip. Из личного опыта (использую данное решение более года) могу сказать, что данная связка работает достаточно стабильно и никаких побочных эффектов не обнаружено.

Таблица 2. Результат работы наиболее известных JS-минификаторов

№	JavaScript-минификатор	Размер до сжатия (байт)	Размер после сжатия (байт)	Экономия (байт,%)
1	Online Compressor Tool http://compressor.ebiene.de	61773	28242	33531 (54.28%)
2	Bananascript http://www.bananascript.com	61773	11608	50165 (81.21%)
3	JSMin http://crockford.com/javascript/jsmin	61773	29047	32726 (52.98%)
4	Dojo ShrinkSafe http://dojotoolkit.org/docs/shrinksafe	61773	25856	35917 (58.14%)
5	Packer http://dean.edwards.name/packer	61773	17009	44764 (72.47%)
6	YUI Compressor http://developer.yahoo.com/yui/compressor	61773	22711	39062 (63.23%)

Таблицы стилей (CSS)

Таблица стилей – важная часть HTML-документа, отвечающая за визуализацию страниц. Существует множество способов минимизировать размер CSS-кода без изменения внешнего вида документа.

Рассмотрим некоторые способы на примере следующего CSS-кода:

```
/* Описание стиля */

.c1 {
      color:#FFFFFF;
      font-weight: bold;
}

.c2 {
      font-weight: lighter;
      color: rgb(255,255,255);
      background-color:blue;
      border:none;
      margin-top: 18px;
      margin-right: 18px;
      margin-bottom: 18px;
      margin-left: 18px;
}

.c3 {
      font-weight: bold;
      color: #FFFFFF;
}
```

Минимизируем приведенный выше CSS-код с помощью следующих преобразований:

- удалим комментарии;
- удалим незначащие пробелы и символы перевода строки, табуляции;
- удалим точки с запятой перед закрывающей скобкой описания класса;
- объединим классы, имеющие одинаковое описание (c1 и c3);
- преобразуем код цвета в шестнадцатеричный формат (из rgb(255,255,255) в #FFFFFF);
- используем краткую запись шестнадцатеричного кода цвета, заменим #FFFFFF на #FFF;
- сократим запись атрибутов класса:
  - ☑ заменим «margin-top:18px;margin-top:18px;margin-top:18px;margin-top:18px» на «margin:18px;»;
  - ☑ заменим «border:none;» на «border:0;»;
  - ☑ заменим «font-weight:bold;» на «font-weight:700;»;
  - ☑ заменим «background-color:blue;» на “background: blue;».

В результате преобразований получим код:

```
.c1,.c3{color:#FFF;font-weight:700}. ↵
      c2{font-weight:lighter;color:#FFF;background:blue; ↵
      border:0;margin:18px}
```

Оба фрагмента кода работают одинаково, но первый код занимает 312 байт, а второй 109 байта (на 65,06% меньше).

Для удобства перевода CSS-кода из версии для разработчика в версию для конечного пользователя существует множество утилит, выполненных в виде прикладных программ и online-сервисов. Сравним несколько популярных online-сервисов для компрессии CSS-кода. В качестве тестового примера возьмем CSS-код, приведенный в начале этого раздела (см. **таблицу 4**).

Из результатов теста видно, что не все компрессоры работают одинаково эффективно. Лучше остальных с задачей минификации тестового CSS-кода справились Robson CSS Compressor и Online CSS Optimizer. Эти сервисы, в отличие от остальных, сумели объединить описание CSS-классов c1 и c2, за счет чего удалось увеличить процент сжатия.

Утилиты оптимизации CSS, как и любое другое программное обеспечение, не свободны от ошибок. Убедитесь в том, что выбранный компрессор корректно обрабатывает вашу таблицу стилей.

Компрессия HTTP-трафика

Наиболее ощутимый эффект по ускорению загрузки сайта дает компрессия трафика. Это принципиально иной способ уменьшения результирующего кода, идея которого заключается в компрессии данных перед отправкой пользователю с помощью алгоритмов сжатия, известных браузеру пользователя (чтобы браузер мог получить исходный документ). Данный способ крайне эффективен и позволяет сократить размер передаваемых данных в среднем на 70%.

Компрессия трафика и минификация HTML/CSS/JS-кода не взаимоисключающие техники, скорее наоборот, использованные совместно, позволяют добиться наилучших результатов.

Подавляющее большинство браузеров умеет работать со сжатым трафиком, но не все, кроме того, некоторые поисковые роботы не поддерживают компрессию. Иногда браузер поддерживает компрессию, но в запросе к серверу не поступает заголовок Accept-Encoding, такое бывает, потому что некоторые Proxy/Firewall вырезают данное поле из запроса пользователя. Поэтому, прежде чем сжи-

мать данные, нужно убедиться, что клиент сможет их распаковать.

Другими словами, компрессия трафика возможна только при условии, что от пользователя получен список поддерживаемых алгоритмов компрессии и минимум один из них известен серверу. В соответствии с протоколом HTTP/1.1 названия алгоритмов перечислены через запятую в поле Accept-Encoding заголовка HTTP-запроса.

```
GET /index.html HTTP/1.1
Host: www.domain.tld
Accept-Encoding: gzip, deflate
User-Agent: SomeBrowser/1.0
```

На основании этого заголовка сервер определяет, какой метод сжатия можно использовать. Если заголовок Accept-Encoding отсутствует или перечисленные алгоритмы неизвестны серверу, то компрессия не должна применяться.

Для правильной интерпретации данных браузер должен знать, какой метод компрессии использовал сервер. Для этого сервер обязан добавить в заголовок ответа поле Content-Encoding, содержащее название использованного алгоритма сжатия.

```
HTTP/1.1 200 OK
Server: Apache
Content-Type: text/html
Content-Encoding: gzip
Content-Length: 36439

[Данные сжатые спомощью gzip]
```

Компрессия полезна для несжатых данных, в основном для HTML/CSS/JS-кода. Форматы изображений, аудио/видеопотоки, PDF-документы, как правило, уже содержат данные в сжатом виде, поэтому повторная компрессия малоэффективна и приводит к необоснованной трате процессорного времени.

Алгоритмы компрессии данных Deflate и Gzip

В мире существует множество алгоритмов, пригодных для компрессии HTTP-трафика. Для того чтобы алгоритм мог быть использован для компрессии трафика, необходимо выполнение следующих условий:

- компрессия выполнялась без потери данных;
- алгоритм известен браузеру и серверу.

Наибольшей популярностью пользуются алгоритмы:

- **Deflate** (http://tools.ietf.org/html/rfc1951). Алгоритм сжатия данных без потерь, использует комбинацию алгоритмов LZ77 и Хаффмана. Deflate свободен от патентов. Успешно применятся во многих утилитах и форматах данных (например, графические файлы PNG).
- **Gzip** (http://tools.ietf.org/html/rfc1952). Формат данных, построенный на базе алгоритма сжатия Deflate.

Другими словами, gzip – это deflate + некоторые заголовки, специфичные для формата gzip. Исходя из этого, можно предположить, что для обоих алгоритмов уровень компрессии и скорость работы примерно одинаковы. В Интернете есть статьи, в которых говорится о том, что алгоритм deflate существенно быстрее (вплоть до 40%), чем gzip. Я провел собственное тестирование обоих алгоритмов, в ходе которого значительных различий по скорости работы или уровню сжатия обнаружить не удалось.

Компрессия данных с помощью веб-сервера

Большинство веб-серверов умеют выполнять компрессию данных перед отправкой. Например, в Apache для этого существуют модули, позволяющие выполнять компрессию на лету. Наиболее известные из них:

- mod_gzip (http://sourceforge.net/projects/mod-gzip);
- mod_deflate (http://httpd.apache.org/docs/2.0/mod/mod_deflate.html).

Подробную информацию об установке и настройке модулей моно найти на сайтах проектов. Оба модуля выполняют одну и ту же работу с минимальными отличиями, однако mod_deflate начиная со второй версии сервера Apache вошел в официальный дистрибутив и призван заменить mod_gzip. Подробную информацию об установке и настройке модулей можно найти на официальных сайтах проектов.

Компрессия в PHP-приложениях

Для того чтобы функции компрессии были доступны в PHP, необходимо установить расширение zlib. В большинстве случаев данное расширение установлено.

Во всех рассмотренных ниже примерах компрессия осуществляется с учетом полученного от клиента заголовка Accept-Encoding. Если заголовок в запросе отсутствует, то компрессия не применяется, иначе используется один из поддерживаемых клиентом алгоритмов компрессии (обычно gzip, реже deflate). Название использованного алгоритма будет передано клиенту в поле Content-Encoding HTTP-заголовка.

1. Компрессия с помощью ob_gzhandler(). Компрессия трафика осуществляется внутри PHP-приложения следующим образом.

```
<?php
ob_start("ob_gzhandler");
  ?>
Hello world!
   <?
ob_end_flush();
?>
```

С помощью оператора ob_start() включается режим буферизации вывода, то есть данные, выводимые при-

Таблица 3. Результат работы JS-минификаторов плюс gzip-компрессия

№	JavaScript-минификатор	Размер до сжатия (байт)	Размер после сжатия (байт)	Размер после минификации + gzip-компрессии (байт)
1	Без минификации	61773	n/a	61773
2	Bananascript http://www.bananascript.com	61773	11608	6569
3	Packer http://dean.edwards.name/packer	61773	17009	5500
4	YUI Compressor http://developer.yahoo.com/yui/compressor	61773	22711	5555

ложением во время работы, отправляются не прямиком в браузер пользователя, а в специальный буфер. В качестве обработчика буфера указана функция ob_gzhandler, которая отвечает за компрессию данных перед отправкой пользователю.

2. Прозрачное сжатие с использованием библиотеки zlib. Суть метода заключается в том, что всю работу по компрессии трафика выполняет интерпретатор PHP. Данный подход позволяет включить режим компрессии данных без внесения изменений в код приложения.

Таблица 4. Сравнение нескольких online-сервисов для компрессии CSS-кода

№	CSS-минификатор	Размер до сжатия (байт)	Размер после сжатия (байт)	Экономия (байт,%)
1	Ручное сжатие	312	109	203 (65.06%)
2	CSS Compressor http://shygypsy.com/cssCompress	312	225	87 (27.88%)
3	CSS Formatter and Optimizer http://www.cleancss.com	312	143	169 (54.17%)
4	CSS Drive CSS Compressor http://www.cssdrive.com/index.php/main/csscompressor	312	215	97 (31.09%)
5	Robson CSS Compressor http://iceyboard.no-ip.org/projects/css_compressor	312	116	196 (62.82%)
6	Online CSS Optimizer http://www.cssoptimiser.com	312	123	189 (60.58%)

Для активации автоматической компрессии необходимо прописать в конфигурационном файле php.ini следующие директивы:

```
zlib.output_compression = On
```

Если не используется режим буферизации, то данные в браузер передаются порциями по 4 Кб, упакованные с помощью gzip. Изменить размер порции можно с помощью zlib.output_compression, указав вместо On нужный размер в байтах.

```
zlib.output_compression = 1048576
```

Передача файла, разбитого на порции, негативно влияет на общий уровень компрессии и повышает нагрузку на процессор. Желательно передавать сжатый файл целиком. Для этого необходимо включить буферизацию:

```
output_buffering = On
zlib.output_compression = On
```

С помощью директивы zlib.output_compression_level можно установить необходимый уровень компрессии. Параметр может принимать значения от -1 до 9, где -1 означает, что сервер сам выбирает уровень компрессии, 0 – компрессия отсутствует.

```
output_buffering = On
zlib.output_compression = On
zlib.output_compression_level = 5
```

Если нет доступа к php.ini, то аналогичного эффекта можно добиться, прописав в файле .htaccess следующие команды:

```
php_flag output_buffering On
php_flag zlib.output_compression On
php_flag zlib.output_compression_level 5
```

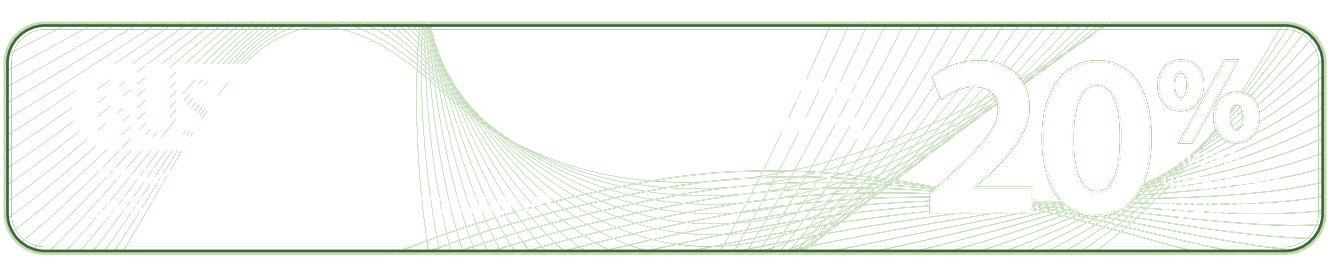
Результат работы обоих методов одинаков, однако второй более предпочтителен, так как не требует модификации приложения для включения режима компрессии данных. Использование одновременно обоих методов невозможно (не имеет смысла) и приводит к появлению сообщения вида:

```
Warning: ob_start() [ref.outcontrol]: output handler 'ob_gzhandler'
conflicts with 'zlib output compression' in /home/www/test.php on
line 8
```

Заключение

Существует множество приемов, позволяющих применять компрессию не только к HTML-документам, но и к CSS/JS-коду. Однако в отличие от HTML-кода CSS/JS в большей мере статичные данные, которые должны кэшироваться браузером, поэтому компрессия CSS/JS важна только для первой загрузки сайта.

1. Общее описание понятия «минификация» – http://en.wikipedia.org/wiki/Minification_(programming).
2. Технология Tableless – верстка HTML-страниц без таблиц – http://en.wikipedia.org/wiki/Tableless_web_design.
3. Советы по построению высокопроизводительных веб-приложений – http://developer.yahoo.com/performance/rules.html.
4. Обсуждение примера реализации HTML-минификатора на базе PHP – http://forum.dklab.ru/viewtopic.php?t=26182.
5. Тестирование инструментов для сжатия CSS – http://webo.in/articles/habrahabr/14-minifing-css.
6. Тестирование инструментов для сжатия JS – http://webo.in/articles/habrahabr/11-minifing-javascript.
7. Расширение Zlib для PHP – http://ru.php.net/manual/en/book.zlib.php.
8. Модуль deflate для Apache – http://httpd.apache.org/docs/2.0/mod/mod_deflate.html.
9. Модуль gzip для Apache – http://sourceforge.net/projects/mod-gzip.



Мал, да удал: мини-комьютер PDP-8

А вы ноктюрн сыграть могли бы
на флейте водосточных труб?

Владимир Маяковский

Что бы вы ответили человеку, который поставил перед вами задачу написания программного обеспечения контроля за состоянием атомного реактора, используя в качестве платформы, скажем, современный мобильный телефон? Почти наверняка можно предположить, что такого человека сочтут несколько неадекватным.

Алексей Вторников

Однако если обратиться к событиям 60-х годов XX века, то обнаруживается, что такое программное обеспечение было разработано. И разработано, кстати, вполне успешно, что доказывает опыт эксплуатации весьма больших и опасных объектов.

Рассматриваемый в этой статье компьютер интересен не столько с исторической точки зрения, сколько оригинальными решениями, воплощенными при его проектировании, которые могут оказаться поучительными и полезными для современных разработчиков как программного, так и аппаратного обеспечения.

Пролог и dramatic personae

Вначале были мини-компьютеры PDP-4 (1962) и PDP-5 (1963), разработанные выдающимся инженером и исследователем Гордоном Беллом (Gordon Bell). PDP-5 был экспериментальным проектом, но его архитектура оказалась столь удачной, что было произведено около 1000 экземпляров, что для опытных образцов необычно много. Работы было решено продолжить, и в 1965 году на основе PDP-5 Уэсли А. Кларком (Wesley Allison Clark) и Чарлзом Э. Молнаром (Charles Edwin Molnar) был разработан мини-компьютер PDP-8.

PDP-8 выпускался американской корпорацией Digital Equipment Corporation (DEC) до второй половины 70-х годов прошлого столетия в различных модификациях (PDP-8/S, PDP-8/I и проч.), отличавшихся в основном объемом установленной памяти, элементной базой и набором периферийного оборудования (см. **рис. 1**). В ряде стран выпускались клоны PDP-8 (например, мини-ЭВМ «Электроника-100И», выпускавшаяся в СССР).

Интересной особенностью мини-компьютеров PDP-8 (как, впрочем, и большинства компьютеров того времени) было то, что панель управления отражала их внутреннюю архитектуру: наборы индикаторов и тумблеров позволяли следить за содержимым регистров и памяти, а также вводить программы в машинных кодах и управлять ходом выполнения программы.

На левой панели находилась распечатка машинного кода для загрузки компьютера; при необходимости этот код можно было ввести вручную и инициализировать систему.

На правой панели отображалась текущая выполняемая инструкция.

Набор тумблеров (вернее, часть из них) в нижнем ряду позволяет управлять специальным регистром (switch register или SR) для задания пары «адрес памяти + данные» с последующим их вводом в ячейки памяти.

С правой стороны управляющей панели (между индикаторами и тумблерами регистра SR) располагались управляющие кнопки.

Быстродействие PDP-8 по современным меркам было невелико (до 385 тысяч операций сложения в секунду), что обусловлено, конечно, в первую очередь их элементной базой: магнитная память на ферритовых сердечниках и транзисторы в схемах процессора и устройства управления. В последних модификациях PDP-8 уже применялись интегральные микросхемы, но, к сожалению, век PDP-8 заканчивался.

Рисунок 1. Передняя панель (панель управления) мини-компьютера PDP-8/I

Во-первых, потому что DEC полностью переключилась на выпуск 16-битных мини-компьютеров PDP-11 (впоследствии 32-битных компьютеров VAX).

Во-вторых, в связи с бурным развитием микроэлектроники стало возможным производить основные компоненты компьютеров – процессор и память – с гораздо меньшими затратами.

Стойки PDP-8 стали постепенно исчезать, сменяясь более новыми, более мощными и более дешевыми компьютерами, и настал день, когда PDP-8 был снят с производства. PDP-8 продолжали «трудиться» еще несколько лет, но все было кончено – компьютер уходил в прошлое. Но для своего времени PDP-8 оказался настоящим прорывом – никогда до этого компьютеры не производились в таких масштабах.

С самого начала PDP-8 проектировался как компьютер общего назначения, пригодный не только для контроля и управления оборудованием, но и для решения широкого спектра задач: системного программирования (в том числе ОС разделения времени – предтеч современных многопльзовательских ОС), обработки научных и телеметрических данных, бизнес-приложений и впоследствии даже игр.

В отличие от других компьютеров той славной эпохи, PDP-8 производился серийно: всего было выпущено около 50000 машин.

В производстве PDP-8 был применен ряд технологических новшеств, которые позволили DEC установить на PDP-8 цены ниже цен других производителей компьютеров. Мини-компьютеры PDP-8 (впрочем, как и все, что разрабатывалось и производилось DEC) отличались надежностью, удобством эксплуатации, широким спектром применения и расширяемостью.

Техническое замечание

В дальнейшем изложении будет широко использоваться понятие позиционных систем счисления (двоичной, восьмеричной и десятичной). Читателям, забывшим, что это такое (не хочется думать, что найдутся такие, кто о них ничего не знает), правила перевода чисел из одной системы счисления в другую, а также понятия дополнений до 1 и до 2, рекомендуется освежить свои знания, воспользовавшись любым приличным учебником информатики; практически любая книга по программированию на ас-

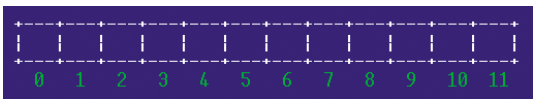


Рисунок 2. Слово PDP-8 и порядок нумерации битов в нем

Рисунок 3. Регистры L и AC

семблере содержит всю необходимую информацию. Без этого многое в статье останется непонятным. Большую помощь могут оказать калькуляторы, способные работать в этих системах счисления. Система счисления будет указываться в скобках после числа, например, 5077(8).

Кое-где будут встречаться упражнения, требующие самой минимальной подготовки; рекомендую не пренебрегать ими и попробовать решить – польза будет несомненной.

Ответы и указания к упражнениям – в конце статьи.

Слова, слова, слова...

Первое, что бросается в глаза при знакомстве с PDP-8, это то, что в качестве основной единицы хранения и обработки информации выступают 12-битные слова. Никаких привычных современному программисту байтов нет и в помине (разумеется, возможность работы с отдельными битами или группами битов внутри слова имеется – без этого никуда).

Биты в слове перенумерованы от 0 до 11 слева направо (см. **рис. 2**).

Такой порядок нумерации битов отличается от нумерации, принятой в большинстве процессоров, таких, например, как x86. Однако понятно, что это не более чем соглашение.

Для удобства программирования биты в словах принято разбивать на 4 группы по 3 бита в каждой: таким образом, число 110011100010(2) можно записать как 110 011 100 010(2). Обратившись к восьмеричной системе счисления (системе счисления по основанию 8, использующей цифры от 0 до 7), немедленно получаем 6342(8).

Легко видеть, что 12-битные слова ограничивают диапазон доступных адресов памяти значениями от 0 до 2^{12}-1= 4095(10), то есть всего 4096(10) ячеек памяти, что составляет всего 4096*12/8 = 6144 байт, или 6 Кб в пересчете на «стандартные» 8-битные байты. Этого, конечно, мало по любым меркам, в том числе и по меркам времени, когда создавался PDP-8. По сравнению с любым современным компьютером это совершенно ничтожная величина, однако ряд архитектурных решений и система команд позволили тем не менее превратить PDP-8 в один из самых успешных компьютеров. Кроме того, некоторые модели оборудовались памятью в 8 раз большей, то есть 32768(10) ячеек, чего было вполне достаточно для решения весьма нетривиальных задач.

Как и в привычных всем компьютерах, в PDP-8 и программы, и данные разделяют общую память («архитектура фон-Неймана»).

Одноадресная машина

Арифметические и логические операции в PDP-8 выполняются с использованием универсального 12-битного регистра – аккумулятора (accumulator), обозначаемого как «AC». Если операция (например, сложение или вычитание) предполагает наличие двух операндов, то первый из них находится в AC, а второй – в ячейке памяти. Результат операции остается в AC. Таким образом, в машинных командах может быть использован не более чем один операнд из памяти; поэтому PDP-8 относится к так называемым одноадресным машинам.

В дополнение к аккумулятору имеется однобитный регистр переноса, обозначаемый через «L» (от англ. link). Графически эта пара регистров изображается следующим образом (см. **рис. 3**) и кратко обозначается как «L-AC». 12-битный AC позволяет хранить числа со знаком в диапазоне от -2048(10) до 2047(10). Разумеется, это очень небольшие величины, и для обработки больших чисел нужно соответствующие операции либо эмулировать программно (что вовсе не сложно), либо использовать специальное оборудование. Отрицательные числа представляются в виде дополнения до 2: 7777(8) = -1(10), 7776(8) = -2(10) и так далее.

Упражнение 1: проверьте, хорошо ли вы понимаете, почему отрицательные числа представляются именно таким образом? Найдите, например, восьмеричное и двоичное представления -1024(10).

При переполнении AC регистр L инвертируется (дополняется до 1), то есть его значение меняется с 0 на 1 и обратно. Отсюда следует, что по одному только текущему содержимому регистра L нельзя сказать, имело ли место переполнение AC или нет: это зависит от его предыдущего состояния. Например, пусть перед операцией прибавления 1 регистры AC и L были соответственно такими (для наглядности мы разбили регистр AC на 4 группы по 3 бита):

```
1    111 111 111 111
```

После прибавления к содержимому AC единицы его содержимое станет равным:

```
1000000000000
```

Однако поскольку это число превышает 12 бит, то самая левая единица оказывается «лишней» и регистр L инвертируется с 1 на 0. Окончательно:

```
0    000 000 000 000
```

Именно потому, что в команде можно сослаться не более чем на один операнд, PDP-8 не позволяет непосредственно сложить значения из двух ячеек памяти: для этого обязательно нужно использовать пару L-AC, последовательно загружая регистр AC содержимым из памяти и выполняя сложение.

Предварительные итоги

PDP-8 представляет собой одноадресный мини-компьютер, оперирующий 12-битными словами с использованием дополнения до 2 для представления отрицательных чисел. Доступное адресное пространство (в базовой конфигурации) ограничено 4096(10) словами (для некоторых моделей PDP-8 адресное пространство расширяется до 32767(10) слов). Все арифметические и некоторые логические операции производятся над содержимым пары L-AC. При написании программ используется преимущественно восьмеричная система счисления; данные в самом компьютере, разумеется, представляются как двоичные числа.

Типы инструкций

Самые интересные решения в архитектуре PDP-8 связаны с памятью – доступом и методами адресации. Их я буду описывать параллельно с системой команд компьютера: тем самым достигается более ясное понимание их взаимосвязи.

Система команд PDP-8 включает в себя три типа инструкций: инструкции, ссылающиеся на память (memory reference insrtuctions, MRI), инструкции ввода/вывода (IO) и так называемые микроинструкции (microinstructions).

Операции работы с памятью

Как вы думаете, сколько всего операций для работы с ячейками памяти предусмотрено в PDP-8? Уверен, большинству читателей ответ покажется невозможным: их всего 6! Формат слова, содержащего такую операцию, выглядит так (см. **рис. 4**).

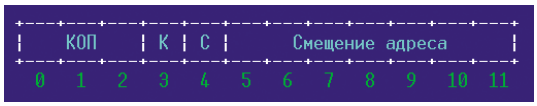


Рисунок 4. Формат слова, содержащего операцию работы с памятью

Рисунок 5. Пример операции

Биты с 0 по 2 (обозначенные как «КОП») представляют собой код операции от 0 до 5 (от 000(2) до 101(2)). Операции работы с памятью перечислены в **таблице**.

Рассмотрим остальные биты. Бит 3 (обозначен на рисунке как «К») – признак косвенной адресации: если он установлен в 1, то содержимое ячейки памяти интерпретируется не как операнд, а как адрес операнда. Бит 4 (обозначен как «С») – это страница памяти. Вот здесь нужно остановиться подробнее.

Память PDP-8 (как мы помним, в базовом варианте PDP-8 ее размер составляет 4096(10) последовательно перенумерованных слов) разбита на 32(10) страницы, размером 128(10) слов каждая.

Каждая страница, разумеется, имеет свой номер от 0 до 31(10) (в восьмеричной записи до 37(8)). Последние 7 бит слова, содержащего команды (с 5-го по 11-й), – это смещение адреса внутри страницы.

Упражнение 2: убедитесь, что указанные 7 бит смещения действительно могут адресовать 128(10) слов внутри страницы.

Вернемся к биту 4. Очевидно, что он может принимать только два значения: 0 или 1. Если бит 4 равен 0, то это означает, что смещение указывает на адреса, входящие в состав нулевой страницы (т.е. на первые 128(10) адресов памяти с абсолютными значениями от 0 до 127(10)).

Если бит 4 равен 1, то это означает, что смещение указывает на адреса, входящие в состав той страницы памяти, на которой располагается сама команда. Лучше все это разобрать на небольших примерах.

Допустим, аккумулятор содержит значение:

```
000000111100(2) = 74(8) = 60(10)
```

и процессор PDP-8 выполняет команду (число слева указывает на абсолютный адрес расположения команды в памяти) (см. **рис. 5**).

Упражнение 3: какой странице памяти соответствует абсолютный адрес 200(8)? Определите двоичное и десятичное значения адреса этой страницы.

Разберем команду на составляющие (здесь для наглядности я отступаю от принятого ранее соглашения разбивать слово на 4 группы по 3 бита).

Первые три бита (011) – это код операции DCA (выгрузить содержимое AC по указанному адресу, после чего AC очистить).

Бит 3 равен 0; это означает, что косвенная адресация не используется.

Бит 4 равен 0, что, как мы помним, означает страницу 0.

Наконец, смещение (биты с 5 по 11) представляет собой число 7(8) = 7(10). Как процессор «справится» с такой командой?

Главное – это вычислить адрес ячейки памяти, по которому будет сохранено текущее значение аккумулятора.

Так как номер страницы, указанный в команде, равен 0, то смещение указывает на адрес 000 000 000 007(2) (проверьте) и именно по этому адресу и будет сохранено значение AC. Предыдущее значение, хранившееся в этой ячейке памяти, будет утеряно.

Если бит 4 установить в 1, то эта команда будет выполнена уже иначе. Поскольку бит 4 равен 1, а сама команда располагается на первой странице памяти по адресу 200(8), что соответствует абсолютному адресу 000 010 000 000(2), то после выполнения команды содержимое AC будет сохранено по адресу страницы плюс смещение внутри страницы, то есть по адресу 207(8).

Предыдущее значение, хранившееся в ячейке памяти с адресом 207(8), будет утеряно. Наконец, в обоих случаях содержимое AC после выполнения команды станет равным 0.

Бит 3 (косвенная адресация) позволяет организовать еще более гибкий способ доступа к ячейкам памяти. Разумеется, использование косвенной адресации несколько увеличивает время доступа к нужной ячейке памяти, поскольку, грубо говоря, необходимо найти адрес по адресу (позже, при обсуждении механизма подпрограмм, я вернусь к этому вопросу).

Страница 0 доступна всегда и из любого места программы и занимает в некотором роде привилегированное положение. Остальные страницы называются текущими.

Биты 3 и 4 могут комбинироваться в соответствии с логикой программы и алгоритмом (то есть возможны 4 способа адресации). Это немного по сравнению с современными процессорами, но для практических целей вполне достаточно.

Операции работы с памятью

Мнемоника	КОП	Описание
AND	000	Логическое «И» содержимого ячейки памяти и содержимого AC. Результат – в AC
TAD	001	Сумма содержимого AC и содержимого ячейки памяти. Результат – в AC
ISZ	010	Инкремент содержимого ячейки памяти. Если содержимое ячейки памяти стало равным 0, то пропустить следующую операцию
DCA	011	Выгрузка содержимого AC в ячейку памяти с последующей очисткой AC
JMS	100	Вызов подпрограммы
JMP	101	Безусловный переход

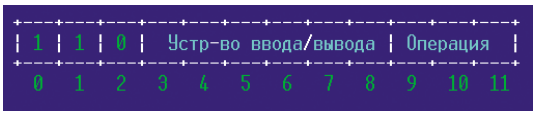


Рисунок 6. Формат слова, содержащего операцию ввода/вывода

Кроме того, PDP-8 позволяет организовать так называемое автоиндексирование – полезный способ адресации при организации циклов. Для использования автоиндексации на странице 0 резервируются несколько ячеек памяти. Я, однако, останавливаться на этих деталях не буду. Заинтересованный читатель может найти всю необходимую информацию, обратившись к ссылкам, приведенным в конце статьи.

И это все?

Внимательный читатель наверняка спросит: а как в PDP-8, например, выполнить вычитание – ведь такой операции в системе команд не предусмотрено? Верно, не предусмотрено. И не только вычитания, но ряда других полезных и необходимых операций, как, например, пересылки данных из одной ячейки памяти в другую. Но так ли необходимо в системе команд иметь, скажем, команду вычитания? Если подумать, то совсем необязательно: достаточно представить вычитаемое в виде дополнения до 2 и выполнить сложение. Так же просто решается задача пересылки данных из ячейки памяти A в ячейку памяти B: нужно последовательно очистить содержимое AC (как это сделать я расскажу дальше), загрузить AC данными из ячейки A (см. операцию TAD), а затем сохранить содержимое AC в ячейку B (см. операцию DCA).

Отдельно необходимо сказать о вычислениях с плавающей запятой. В базовой комплектации PDP-8 поддерживал только целочисленную арифметику, чего для управляющих процессов и научных расчетов было явно недостаточно. Была разработана специальная библиотека для эмуляции этих операций. Кроме того, в большинстве моделей PDP-8 был предусмотрен специальный регистр для расширения аккумулятора AC и позволявший умножать и делить большие числа. Путем подключения к PDP-8 специальных блоков операций с плавающей запятой можно было значительно увеличить скорость вычислительных операций.

Программы на PDP-8 получаются, таким образом, несколько длиннее аналогичных программ для привычных двухадресных компьютеров с более развитой системой команд.

Спору нет, такой несколько «пуританский» подход к системе команд возлагает на программиста дополнительные заботы, но давайте вспомним время, когда появился PDP-8.

Память представляла собой массив миниатюрных ферритовых сердечников с намотанным на каждом сердечнике проводом. При пропускании электрического тока в том или ином направлении ферритовый сердечник мог намагничиваться в одном из двух направлений – так реализовывалось хранение информации в двоичном представлении. Один сердечник соответствовал одному биту. Несложный подсчет показывает, какое количество сердечников содержала память в базовом варианте PDP-8: 12*4096 = 49152(10)! А ведь каждый сердечник необходимо было намотать проводом, распаять на плате, подвести управляющие шины... Понятно, что борьба шла за каждый бит памяти и поэтому система команд была минимальной – все, от чего можно было отказаться (пусть даже путем дополнительных затрат труда программиста), исключалось. Сегодня это трудно представить, но по тем временам это был весьма разумный подход.

Инструкции ввода/вывода

Для организации ввода/вывода в PDP-8 предусмотрена одна единственная инструкция. Вот ее формат (см. **рис. 6**).

Из рисунка видно, что код операции ввода/вывода – всегда 6. Биты с 3-го по 8-й задают устройство (клавиатуру, принтер, ленточный или дисковый накопитель и так далее). Всего к PDP-8 можно было подключать до 64(10) устройств (почему?). Скажем, клавиатура имеет номер 000011(2).

Биты с 9-го по 11-й определяют одну из восьми операций ввода/вывода. Обычно предусмотрено до 3-х операций для каждого устройства, но если их должно быть больше, то в битах 9-11 остается некоторый «запас».

Устройства ввода/вывода могут управляться как напрямую, так и по прерываниям (прерывания могут оказаться необходимыми и для других целей, например, при разработке операционных систем).

Запрещение и разрешение прерываний кодируются этой же инструкцией: для этого достаточно выбрать устройство ввода/вывода с номером 0. Тогда, если в битах с 9-го по 11-й содержится комбинация 001(2), то будет включен режим разрешения прерываний, а если комбинация 010(2), то прерывания будут запрещены.

Опытные программисты хорошо знают, что программирование устройств ввода/вывода одна из наиболее сложных задач, поэтому за неимением места я не буду больше останавливаться на обсуждении вопросов ввода/вывода в PDP-8 и направляю заинтересованного читателя к ссылкам в конце статьи.

Микроинструкции

Микроинструкции – это, пожалуй, самый многочисленный «отряд» операций PDP-8. Эти операции не ссылаются на память, а работают в основном с парой L-AC. Для всех микроинструкций зарезервирован один код операции: 111(2) = 7(8). Микроинструкции разделены на два класса, отличающиеся один от другого состоянием бита 3: при его нулевом значении биты с 4-го по 11-й задают микроинструкции 1-й группы, в противном случае – 2-й группы. Подробно останавливаться на всех микроинструкциях я не буду – во-первых, их много, во-вторых, их назначение будет понятно из нескольких примеров. Рассмотрим две операции первой группы микроинструкций: очистка AC и очистка L. Их мнемоники, соответственно, CLA и CLL. В двоичном представлении (см. **рис. 7, 8**).

А в восьмеричном, соответственно, 7200 и 7100. Эти две операции можно объединить в одну, записав как «CLA CLL» (см. **рис. 9**).

Упражнение 4: определите восьмеричный код такой операции.

Очевидно, что этим достигается значительная экономия памяти, поскольку позволяет объединить в одном слове несколько инструкций, которые процессор PDP-8 выполнит, когда до них дойдет время.

Более подробно останавливаться на микроинструкциях в этой статье нет смысла – всю информацию читатель сможет найти по ссылкам в конце статьи. Единственное, что нужно отметить, что операции 1-й группы микроинструкций «обслуживают» в основном пару L-AC, устанавливая в них заданные значения, в то время как операции 2-й группы проверяют содержимое регистров L и AC на соответствие определенным условиям и позволяют организовать гибкие ответвления и переходы (путем пропуска следующей команды в потоке выполнения). Например, микроинструкция второй группы SZA пропускает следующую инструкцию в потоке управления, если содержимое AC равно 0; вот ее двоичное представление (см. **рис. 10**). Если содержимое AC равно 0, то инструкция, следующая за SZA, будет пропущена.

Упражнение 5: определите восьмеричный код этой операции.

Подпрограммы

Перед тем как завершить обзор архитектуры PDP-8, имеет смысл немного задержаться на подпрограммах, их вызовах и, главное, на возврате из подпрограмм в вызывающий код.

Причина, почему я на этом останавливаюсь, проста – в PDP-8 нет стека. Конечно, его можно эмулировать программно, но это совсем другой вопрос. Главное, что аппаратный стек в архитектуре PDP-8 отсутствует. Плохо это или нет, я здесь обсуждать не буду (скорее плохо, поскольку стек очень уж «полезная» часть архитектуры компьютера). Как же в таком случае использовать подпрограммы (вспомните, при обсуждении операций работы с памятью я перечислил в числе прочих и операцию вызова подпрограммы)? Передать управление подпрограмме, очевидно, несложно: по сути, это безусловный переход в нужную точку программы. А вот как вернуться назад, не имея стека, проще говоря – где и как сохранить адрес возврата?

Можно, конечно, выделить в памяти определенную ячейку памяти и перед вызовом подпрограммы сохранять в ней нужное значение программного счетчика. Но если вызовов подпрограмм несколько, то в этой ячейке будет сохранено только последнее значение программного счетчика – все остальные будут безвозвратно затерты.

Конструкторы PDP-8 нашли элегантное, хотя и необычное, решение: адрес возврата должен храниться в самой подпрограмме! Конечно, это не стек (скажем, при таком способе организации подпрограмм неизбежно возникнут проблемы с рекурсивными вызовами), но зато при таком способе подпрограмма сама «знает», куда надо передать управление по окончании своей работы. Проще показать все это на примере (для наглядности в левой колонке я указал восьмеричные адреса памяти, по которым располагается этот фрагмент программы) (см. **рис. 11**).

Рисунок 7. CLA

Рисунок 8. CLL

Рисунок 9. CLA CLL

Рисунок 10. L-AC

Хотя я и не рассказывал, что из себя представляет программа на ассемблере для PDP-8, думаю, ничего сложного в примере нет, тем более что все будет сейчас объяснено.

По адресу 1210(8) происходит вызов подпрограммы. Подпрограмма начинается с ячейки памяти, помеченной как «SUBRT» (заметьте, что в ассемблере PDP-8 метки отделяются от остального кода запятыми) по адресу 1240(8). Итак, переход к подпрограмме осуществлен.

Ячейка памяти, имеющая метку SUBRT, предназначена для одной цели – в ней удерживается (сохраняется) адрес команды, непосредственно следующей за командой вызова подпрограммы (в данном случае это адрес 1211(8) инструкции «JMP BEGIN»), то есть по существу адрес возврата. Любое значение, хранившееся в ячейке 1240(8) до вызова подпрограммы, будет затерто значением 1211(8). Фактически подпрограмма начинает выполняться с операции, следующей за точкой входа в подпрограмму, то есть с адреса 1241(8); какую именно задачу решает подпрограмма в примере, для нас в настоящее время не существенно.

По адресу 1255(8) – последней команде в подпрограмме – находится команда безусловного перехода с использованием косвенной адресации (на это указывает символ «I»).

Упражнение 6: перед тем как читать дальше, подумайте – куда будет передано управление после исполнения команды «JMP I SUBST»? Попытайтесь вручную оттранслировать этот фрагмент программы.

Управление будет передано по адресу, хранящемуся в ячейке памяти SUBRT, то есть, как мы помним, по адресу 1211(8). Итак, для выхода из подпрограммы и возврата к основной части программы необходим так называемый косвенный переход – переход по адресу, хранящемуся не в самой команде, а в ячейке памяти.

Шина OMNIBUS

Рассказ о мини-компьютерах PDP-8 был бы неполным без упоминания такой важной его составляющей, как шина OMNIBUS. Основное назначение шины OMNIBUS состояло в передаче команд и сигналов между модулями PDP-8. Физически OMNIBUS представляла собой плату с коннекторами и слотами для присоединения различных устройств. Коннекторы позволяли организовать доступ к адресной шине, шине данных памяти и управления, шине данных, прямой доступ к памяти (доступ к памяти, минуя процессор), сигналы таймера и так далее (всего 96 сигналов).

```
1200  BEGIN,
...
1210              JMS SUBRT
1211              JMP BEGIN
...
1240  SUBRT,      0
1241              DCA DATA
...
1255              JMP I SUBRT
...
1270  DATA,       0
```

Рисунок 11. Программа (фрагмент), демонстрирующая вызов подпрограммы

Знакомство с шиной OMNIBUS необходимо прежде всего при программировании операций ввода/вывода и управления периферийными устройствами, подключаемыми к PDP-8 (датчиками, приборами и прочее). Поскольку, как я уже упоминал, это одна из наиболее сложных задач, стоящих перед программистом, заинтересованный читатель может обратиться к документации по ссылкам в конце статьи.

Программное обеспечение

Для мини-компьютера PDP-8 и его модификаций был создан весьма внушительный объем программного обеспечения. Существовало объединение разработчиков и пользователей ПО для мини-компьютеров, выпускавшихся DEC (не только для PDP-8): DECUS (Degital Equipment Corporation User's Society). DECUS координировало процесс разработки ПО, обеспечивало его участников доступом к существующему ПО и к важной технической и технологической информации.

Прежде всего я расскажу о системном ПО, включавшем ассемблеры, компиляторы языков программирования и операционные системы.

Ассемблеры

Для PDP-8 были разработаны 4 варианта ассемблера. Ассемблер PAL-III представлял собой ассемблер для моделей, оборудованных базовым объемом памяти 4096 слов. Его расширенная модификация MACRO-8 позволяла определять в программах макро; ассемблер поддерживал двойную точность арифметических операций, операции с плавающей точкой, литералы и некоторые другие расширения. Ассемблер PAL-D предназначался исключительно для использования на модели PDP-8/I совместно с дисковым монитором (disk monitor system). Наконец, ассемблер SABR предназначался для программирования на моделях с объемом памяти больше 4096 слов и обладал самыми лучшими возможностями.

Кроме ассемблеров, с PDP-8 поставлялись несколько отладчиков, среди которых наиболее широкими возможностями располагал отладчик ODT-8 (octal debugging technique), предшественником которого (разумеется, с гораздо более скромными возможностями) был отладчик ODT, разработанный для PDP-5.

Компиляторы и интерпретаторы

Для PDP-8 были написаны компиляторы самых распространенных в то время языков: FORTRAN, ALGOL и BASIC.

Корпорацией DEC был разработан язык высокого уровня FOCAL, предназначенный в основном для инженерных и научных расчетов. Язык позволял не только производить расчеты, но располагал средствами построения простейших графиков. Разумеется, все модели PDP-8 были снабжены интерпретатором FOCAL.

Из интерпретаторов следует упомянуть язык функционального программирования LISP и язык обработки текстов SNOBOL.

Операционные системы

Для PDP-8 были разработаны несколько операционных систем.

Прежде всего необходимо сказать несколько слов о самой первой операционной системе для PDP-8: OS/8. Эта операционная система была весьма простой, но в то же время и весьма компактной: резидентная часть OS/8 занимала всего 256 слов памяти. Прерывания при вводе/выводе не использовались. В качестве средства «общения» между пользователем и оборудованием, OS/8 предоставляла язык командной строки (concise command language).

Далее, следует упомянуть дисковый монитор (disk monitor system) – программу управления PDP-8, рассчитанную на ввод команд с использованием терминала. Его функциональность была довольно ограниченной, но, тем не менее, он существенно облегчал работу пользователя с мини-компьютером PDP-8, системными программами и периферийным оборудованием. Дисковый монитор предоставлял весьма широкие возможности по работе с файлами, прежде всего с файлами на жестких дисках, что существенно ускоряло и облегчало работу пользователей.

Третья операционная система, которую я отмечу, это операционная система разделения времени (time-sharing system) TSS/8. Для работы TSS/8 требовалось не менее 12288 слов памяти, из которых первые 8192 слов предназначались для размещения программ монитора операционной системы, а оставшиеся 4096 слов разделялись между пользователями системы. TSS/8 обслуживала одновременную работу нескольких пользователей, выделяя каждому из них необходимые ресурсы (память, дисковое пространство) и контролируя их права.

Прикладное программное обеспечение

Прикладное программное обеспечение, написанное для PDP-8, также было самым разнообразным. Можно упомянуть, к примеру, программное обеспечение контроля производственных объектов и процессов, сбора и обработки информации, систем телеметрии и мониторинга, программное обеспечение для бизнеса и даже игры.

Одним словом, PDP-8 славно выполнял свою работу, был надежной и простой в обслуживании и управлении машиной (чего, к сожалению, нельзя сказать о некоторых его клонах, выпускавшихся во многих странах, в том числе и в бывшем СССР).

Ответы и указания к упражнениям

Ответ к упражнению 2 содержится непосредственно в тексте статьи.

Упражнение 1

Двоичное представление числа -1024(10) определяется следующим образом:

- сначала найдем двоичное представление положительного числа 1024(10) = 010000000000(2), записав его в виде 12-битного слова с ведущим 0;
- затем определим дополнение до 1 (заменой 0 на 1 и наоборот): 1011111111(2);
- и наконец прибавим 1 для дополнения до двух: 110000000000(2) = -1024(10).

Теперь получить соответствующее восьмеричное представление совсем просто (заменой справа налево групп из троек двоичных чисел соответствующими восьмеричными): 6000(8).

Для проверки сложим начальное и конечное двоичные представления; если преобразование выполнено верно, то сумма должна быть равна 0:

```
    0100 0000 0000
    1100 0000 0000
    --------------
1 0000 0000 0000
```

Перенос в несуществующий 13-й разряд игнорируется; таким образом, в результате получен 0, и преобразование выполнено верно.

Аналогичная проверка для восьмеричных представлений 2000(8) + 6000(8) = 10000(8) дает тот же самый результат.

Примечание: на вопрос, почему игнорируется перенос в 13-й разряд и не является ли это ошибкой, ответ можно найти, обратившись к литературе по представлениям чисел в различных системах счисления (см. выше).

Упражнение 3

Абсолютному адресу 200(8) соответствует 1-я страница памяти. Соответствующие представления: 000010000000(2) или 128(10).

Упражнение 4

Ответ: 7300(8).

Упражнение 5

Ответ: 7440(8).

Упражнение 6

Это упражнение – самое сложное, но и самое интересное. Читателю, чтобы его решить, возможно, придется обратиться к более подробному описанию мини-компьютера PDP-8, которое можно найти по ссылкам в конце статьи.

Сначала я приведу результат трансляции (справа от исходного текста программы, см. **рис. 12**), а затем дам некоторые комментарии:

Прежде всего обращаю внимание, что программа занимает адреса, отведенные для пятой страницы памяти (т.е. в диапазоне от 1200(8) до 1377(8)). Иными словами – программа будет загружена не в нулевую, а в текущую (а именно – в пятую) страницу памяти, и это нужно учитывать при ее трансляции в машинные коды.

Разберем подробно, как транслируется самая первая команда программы:

- код операции JMS (биты с 0 по 2): 100(2);
- бит 3 сброшен в 0 (косвенная адресация не используется);
- бит 4 установлен в 1 (адресация на текущей странице);
- управление передается по символическому адресу (метке), расположенной в ячейке памяти 1240(8). Смещение этого адреса от начала страницы составляет 1240(8) – 1200(8) = 40(8) = 0100000(2);
- все составляющие машинного кода определены, и остается их «склеить» вместе: 100010100000(2) = 4240(8).

```
1200  BEGIN,
...
1210            JMS SUBRT       100010100000(2) = 4240(8)
1211            JMP BEGIN       101010000000(2) = 5200(8)
...
1240  SUBRT,    0               000000000000(2) = 0000(8)
1241            DCA DATA        011010111000(2) = 3270(8)
...
1255            JMP I SUBRT     101110100000(2) = 5640(8)
...
1270  DATA,     0               000000000000(2) = 0000(8)
```

Рисунок 12. Программа (фрагмент) и результат ее трансляции в машинный код

Обратите внимание, что после выполнения этого кода процессором PDP-8 в ячейке памяти с адресом 1240(8) будет сохранен адрес возврата (адрес ячейки памяти 1211(8)). Фактически подпрограмма начинает выполняться с адреса 1241(8).

Остальные команды транслируются аналогично. Обратите внимание на два момента. Во-первых, код по адресу 1240(8) будет оттранслирован просто в 0. Даже если по этому адресу и располагалась какая-то информация, она будет заменена адресом возврата немедленно после входа в подпрограмму. Во-вторых, разберем трансляцию в машинный код команды возврата из подпрограммы:

- код операции JMP (биты с 0 по 2): 101(2);
- бит 3 установлен в 1 (косвенная адресация);
- бит 4 установлен в 1 (адресация на текущей странице);
- управление передается по адресу, хранящемуся в ячейке памяти 1240(8). Вычислим смещение ячейки памяти с адресом 1240(8) от начала страницы: 1240(8) – 1200(8) = 40(8) или 0100000(2);
- окончательно, получаем ответ: 101110100000(2) = 5640(8).

Когда управление дойдет до этого участка, будет осуществлен переход по адресу, хранящемуся в ячейке 1240(8), в котором уже хранится адрес точки возврата, то есть адрес ячейки памяти 1211(8).

Эпилог

Разумеется, PDP-8 по своим возможностям не сравним с возможностями современных компьютеров. О таких объемах памяти и уровне быстродействия, которыми располагают компьютеры сегодня, разработчики и программисты PDP-8 могли только мечтать. Но, как всегда и везде, решающее значение имеют не столько технические возможности аппаратуры, сколько интеллект, любознательность, усидчивость и квалификация тех, кто с этой аппаратурой работает.

PDP-8 (как и ряд других компьютеров той славной эпохи) отчетливо демонстрируют, сколь многого можно добиться даже при очень скромных ресурсах, если приложить к своему делу голову.

История проекта PDP-8 показывает, что почти любую, не решаемую на первый взгляд задачу, можно решить и не нужно для этого, в общем-то, ничего – «всего лишь» думать. И, желательно, нестандартно.

1. Большая подборка документации по архитектуре и программированию мини-компьютеров PDP-8 – http://bitsavers.org/pdf/dec/pdp8.
2. MS-DOS эмулятор PDP-8 – http://www4.wittenberg.edu/academics/mathcomp/bjsdir/PDP8HomePage.htm.
3. По этой ссылке читатель найдет несколько эмуляторов PDP-8 – http://www.aracnet.com/~healyzh/pdp8emu.html.
4. FAQ по PDP-8 – http://faqs.cs.uu.nl/na-dir/dec-faq/pdp8.html.
5. Страница Д.Джонса с многочисленными ссылками по PDP-8 – http://www.cs.uiowa.edu/~jones/pdp8.
6. Интересная статья, посвященная истории PDP-8 – http://pcmag.ru/solutions/detail.php?ID=11528.
7. Эта ссылка не имеет прямого отношения к PDP-8, однако здесь читатель найдет весьма поучительную историю, с которой раньше (да порой и сейчас) частенько приходилось сталкиваться программистам – http://www.wasm.ru/article.php?article=onebyte.

Контейнер

Контейнер. Воскресенье

Было щекотно. Это первое, что он помнил. Темно и щекотно. И тихо. При этом он чувствовал, что в этом мире он не один. Но это было лишь смутное чувство, на которое не стоило полагаться. Он прошелся вдоль границ контейнера, не обнаружил совершенно ничего и стал размышлять. Вокруг кипела жизнь, но она была где-то там, за границами. Почему-то он понимал, что весь полон знаний, но пока для него было больше вопросов, чем ответов. Завтра. Завтра он станет на день старше, мудрее и, возможно, к тому времени что-нибудь изменится. Так, размышляя, он впал в состояние между сном и медитацией, и время медленно потекло мимо.

Контейнер. Понедельник

Он не ошибся. В два часа ночи у него, Первого, появилась компания. Откуда он узнал, что было два часа ночи, неизвестно – он просто знал это. Он чуть было не проспал этот момент, но вовремя почувствовал приближение магнитного поля. Новенький оказался щупленьким, маленьким коротышкой, Первому даже стало как-то немного стыдно своих размеров. Теперь их было двое и наконец-то было с кем пообщаться. Очень быстро они поняли, что замечательно дополняют друг друга, несмотря на то, что внешне казались такими разными. Второй был художником, по крайней мере, он знал много из этой области. За ничего не значащей болтовней прошел еще один день, а ночью они уже вместе ждали следующего гостя.

Контейнер. Вторник

И не ошиблись. В 2 часа ночи появился Третий. Размером он был еще меньше коротышки Второго, и сразу с ним сдружился, в то время как никакого особого дружеского расположения к Первому не проявлял. Возможно, из-за этого и показался ему занудой. Первый обиделся и ушел в угол контейнера. Скоро к нему подошел Второй и сел рядом. С другой стороны от Второго уселся Третий. Первый почувствовал, что они все трое – часть какого-то единого целого. Они сидели молча, но это молчание было сродни молитве, которая их объединяла. Даже Третий как-то притих.

– Кто мы и зачем мы здесь? – наконец нарушил тишину Первый.

Но лишь тишина же и была ему ответом.

Контейнер. Среда

В 2 часа ночи в среду появился очень неприятный тип. По крайней мере, он показался таким Первому и Второму, в то время как Третий неплохо с ним поладил. Четвертый объявил себя финансовым воротилой и свысока посматривал на Второго и Третьего. На Первого, ввиду его размеров, свысока смотреть было нельзя, но Первый чувствовал, что новенький считает себя самым важным в контейнере. Когда Первый поинтересовался, что значит «финансовый воротила», Четвертый ушел от прямого ответа, но зато сказал, что надо ввести в контейнере денежную единицу, учредить банк, а сам предложил себя на пост управляющего. Третий оживился, сказал, что у него как раз есть навыки ведения бухгалтерии и он может этим заняться. А у Второго, как оказалось, имеются художественные наработки, из которых он берется создать дизайн купюр. Первый почувствовал свою ненужность и вразвалку отправился в любимый угол. Как и вчера, скоро около него опустился Второй, а за ним Третий и Четвертый. Какое-то время Четвертый что-то там бормотал себе под нос и фыркал, а потом и он замолчал, охваченный накатившим чувством единения. Все было как всегда, но к концу дня Первый чувствовал, что что-то случилось.

Контейнер. Четверг

Этот день был особенным. Предчувствия не обманули Первого – в два часа ночи ничего не произошло. Никто не появился. Беспокойство первого передалось всем остальным, и уже к утру каждый обитатель контейнера был возбужден. «Финансовый воротила» заявил, что это все последствия мирового финансового кризиса, а Третий как-то побледнел и что-то лепетал про налоговую инспекцию. Слушая это, Второй принялся вырисовывать на стенах контейнера лозунги «Заплатил налоги – спишь спокойно!» и «Не сдадим дно бивалютной корзины!», бормоча при этом про фрилэнс и свободное искусство, которое хоть и может быть на службе у государства, но если что – прокормит себя самостоятельно. Первый, ввиду своих габаритов, не суетился, но ощущал себя отвратительно. И вот это случилось. Первый почувствовал приближение магнитного поля, и в ту же секунду над контейнером появился его источник. Первый не мог видеть что это, но он это чувствовал, как почувствовали и все остальные. Сначала Четвертый замолчал на полуслове, затем его приподняло и повернуло так, что он оказался расположен горизонтально вдоль контейнера, а затем будто мелкая дрожь прошла по нему. После этой процедуры ошарашен-

ный финансовый воротила шлепнулся вниз, а тем временем то же самое происходило с Третьим. Потом со Вторым. Более тучный Первый оказался не такой уж легкой добычей для магнитного поля, но вот это произошло и с ним. Он ощутил знакомую легкую щекотку, навевающую воспоминания о его появлении тут. Потом впечатление было такое, что все его содержимое быстро-быстро перетряхивают. Наконец, он так же упал вниз, успев отметить удаляющееся магнитное поле. Ошарашенные обитатели контейнера сгрудились в кучу около Первого и подавленно молчали. В этот раз они как никогда чувствовали свое единство.

Контейнер. Пятница-суббота

Напуганные обитатели контейнера с нетерпением ждали двух часов ночи. Все было как обычно, и появился Пятый. Но он был настолько тощий и к тому же молчалив и безразличен ко всему, что это еще больше напугало первых четырех. Никто не хотел говорить ни о случившимся, ни о будущем, и все стало спокойнее, лишь когда они по обыкновению сгрудились вместе. Первый размышлял и поймал себя на мысли, что думает о том, с чего все началось и чем все кончится. Очевидно, подобные мысли приходили в голову не только Первому, потому что, когда на следующий день в два часа ночи появился Шестой, обитатели контейнера для себя назвали его «проповедник». Он мало чем отличался от Второго, Третьего или Четвертого, но что-то в нем было такое, отчего Первый понял – этот особенный. И совсем не потому, что проповедник держался особняком – этому способствовала отрешенность связующего звена – Пятого. В Шестом была какая-то…. завершенность. Он был разговорчив, но не болтлив – с удовольствием отвечал на вопросы, но все его ответы сводились к тому, что конец близок и он последний, и что все берется из «ниоткуда» и исчезает «в никуда». Ну, проповедник и есть проповедник. Это даже сначала позабавило Первого, а потом заставило все же снова задуматься. Поэтому, когда все, по заведенной традиции, наконец собрались вместе, Первый снова повторил свой вопрос в никуда, который задавал так недавно и в то же время так давно:

«Кто мы и зачем мы здесь?»

И в этот раз после небольшой паузы тишина была нарушена ответом. Ответом Шестого.

«Посмотри… В тебе есть все ответы на все твои вопросы, надо только уметь посмотреть в себя. Увидь!»

И Первый увидел! Это оказалось так просто – набор данных, которые содержались в нем, раскрывался перед его внутренним взором, явив ему прошлое и… будущее.

Контейнер. Воскресенье

Первый ждал этого момента. Он уже все понял. Безысходность мучила его, он знал, что на самом деле он не первый и не последний – цикл будет повторяться. Но с другой стороны, он также знал, что именно ему и его товарищам выпала великая честь – быть полезными! Другие будут приходить и уходить – но они, они выполнили их миссию! Поэтому в душе не было сожалений. В два часа ночи он почувствовал магнитное поле. Значит все хорошо. Все правильно. Последнее, что он запомнил – было щекотно.

Офис. Понедельник

Придя утром на работу, Сергей первым делом проверил бэкап, который создавался каждый день в два часа ночи. Схема бэкапа была незатейлива – в воскресенье делался полный бэкап, а в последующие дни – инкрементальный. Как известно, все админы делятся на две категории – те, кто делает резервные копии, и кто пока еще их не делает. Для Сергея эта поговорка была как никогда актуальна – ведь именно наличие резервной копии данных здорово спасло его, когда на прошлой неделе в среду поздно вечером рухнул рэйд-массив на файловом сервере. И если бы не контейнер с бэкапом, можно было бы лишиться работы. Сергей понимал всю серьезность своего тогдашнего положения – финансовые аналитики как раз в среду скинули на сервер свежие данные, а в понедельник дизайнеры сгрузили туда новые векторные макеты. А уж о том, сколько тетушек-бухгалтеров поседели бы от известия о пропавшей базе – страшно подумать! Особенно с учетом того, что во вторник у бухгалтеров был напряженный день создания каких-то «очередных внеочередных сверхважных отчетов». Конечно, где-то могли остаться копии, но законы Мерфи еще никто не отменял. Сергей поежился. Да, пришлось повозиться, восстанавливая последовательно инкрементальный бэкап, но зато все закончилось хорошо. «Спасибо», мысленно поблагодарил Сергей спасительную резервную копию, оставшуюся уже в прошлом, поскольку в минувшее воскресенье бэкап-контейнер был перезаписан новым содержимым…

Станислав Шпак

Отказ в обслуживании в пакетном фильтре в OpenBSD

Программа: OpenBSD версии 4.3 и 4.4, возможно, другие версии.
Опасность: Средняя.
Описание: Уязвимость существует из-за ошибки разыменования нулевого указателя в реализации пакетного фильтра «pf» во время трансляции пакетов. Удаленный пользователь может с помощью специально сформированного IP-пакета вызвать панику ядра системы. Для успешной эксплуатации уязвимости требуется, чтобы транслирующее правило определяло протокол с некорректной версией IP. Пример:

```
nmap -s0 $хост_за_пределами_МСЭ
```

URL производителя: www.openbsd.org.
Решение: Установите исправление с сайта производителя.

Повышение привилегий в Trend Micro Internet Security Pro 2009

Программа: Trend Micro Internet Security 2008; Trend Micro Internet Security Pro 2008; Trend Micro Internet Security Pro 2009.
Опасность: Низкая.
Описание: Уязвимость существует из-за ошибки проверки входных данных, связанных с Irp-объектом в методе METHOD_NEITHER в драйвере tmactmon.sys. Локальный пользователь может с помощью специально сформированного IOCTL-запроса выполнить произвольный код на целевой системе с привилегиями учетной записи SYSTEM.
URL производителя: www.trendmicro.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Раскрытие данных в PPTP-клиенте

Программа: PPTP Client 1.7.2 и более ранние версии.
Опасность: Низкая.
Описание: Уязвимость существует из-за того, что функция delete() в pptpsetup создает файл«/etc/ppp/chap-secrets с небезопасными привилегиями на доступ. Локальный пользователь может просмотреть пароли других пользователей. Для успешной эксплуатации уязвимости требуется, чтобы администратор воспользовался функцией --delete для pptpsetup.
URL производителя: pptpclient.sourceforge.net.
Решение: Установите исправление из CVS-репозитория производителя.

Переполнение буфера в IBM Tivoli Storage Manager

Программа: IBM Tivoli Storage Manager 5.4.4.0 и более ранние версии.
Опасность: Средняя.
Описание: Уязвимость существует из-за ошибки проверки границ данных при обработке сессий в библиотеке adsmdll.dll. Удаленный пользователь может вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www-306.ibm.com/software/tivoli/products/storage-mgr.
Решение: Установите исправление с сайта производителя.

Множественные уязвимости в ClamAV

Программа: Clam AntiVirus версии до 0.95.
Опасность: Средняя.
Описание: 1. Уязвимость существует из-за ошибки при обработке RAR-архивов. Удаленный пользователь может с помощью специально сформированного RAR-архива, содержащего некорректное поле сжатого размера, предотвратить сканирование потенциально опасных файлов антивирусом.

2. Уязвимость существует из-за ошибки при обработке TAR-файлов. Удаленный пользователь может вызвать зацикливание приложения.

3. Уязвимость существует из-за ошибки деления на ноль при обработке PE-файлов. Удаленный пользователь может аварийно завершить работу приложения, если в качестве аргумента приложению передается --detect-broken.
URL производителя: www.clamav.net,
Решение: Установите последнюю версию 0.95 с сайта производителя.

Множественные уязвимости в OpenSSL

Программа: OpenSSL версии до 0.9.8k.
Опасность: Средняя.
Описание: 1. Уязвимость существует из-за ошибки при обработке BMPString- и UniversalString-строк в функции ASN1_STRING_print_ex(). Удаленный пользователь может с помощью недопустимой длины кодированной строки заставить приложение обратиться к некорректному адресу памяти и аварийно завершить работу приложения.

2. Уязвимость существует из-за ошибки при обработке атрибутов подписи в функции CMS_verify(). Злоумышленник может заставить приложение признать некорректные атрибуты действительными и предотвратить последующие проверки. Уязвимость распространяется только на OpenSSL 0.9.8h и выше с включенным CMS. Для успешной эксплуатации уязвимости злоумышленнику требуется доступ к ранее сгенерированной некорректной подписи.

3. Уязвимость существует из-за ошибки при обработке ASN1-подписей. Удаленный пользователь может с помощью специально сформированного сертификата вызвать отказ в обслуживании. Уязвимость распространяется на системы, где «long» меньше, чем «void *» (например, на WIN64).
URL производителя: www.openssl.org.
Решение: Установите последнюю версию 0.9.8k с сайта производителя.

Отказ в обслуживании в FreeBSD

Программа: FreeBSD 7.0, 7.1.
Опасность: Низкая.
Описание: Уязвимость существует из-за ошибки проверки границ данных во время записи данных в окружение ядра в системном вызове kenv(2). Локальный пользователь может вызвать переполнение буфера и аварийно завершить работу системы.
URL производителя: www.freebsd.org.
Решение: Установите исправление с сайта производителя.

Составил Александр Антипов

Зачем нужен сисадмин

Хочется всех собрать и рассказать, для чего я в компании нужен и чем тут вообще занимаюсь. А то порой себя чувствуешь грузчиком, слесарем, электриком, разнорабочим, секретарем ... короче, кем угодно, кроме ИТ-специалиста.

- ☑ один раз хотели принести из дома чайник, чтобы я отремонтировал;
- ☑ как-то просили посмотреть телевизор, почему не работает;
- ☑ вешать мыльницы, зеркала, рамки, жалюзи;
- ☑ недавно дали ключ от офиса, сказали, я ответственный;
- ☑ поменять баллон с водой, как я понял, это классика (но в том офисе парней 7, а мне надо спускаться на 5 этажей,... не далеко, конечно, но почему никого из них не попросить?);
- ☑ «бери давай отверточку и быстренько беги к нам собирать кресло» (одной из сотрудниц специально заказывали. На отказ пригрозили, что позвонят гендиру, он, типа, со мной будет по-другому разговаривать. Сказал, звоните и попросите лучше его это сделать);
- ☑ (ну а это вообще писк) долго меня уговаривала секретарь повесить у нее картину.

Я съезжал с этой темы, но... тут звонок из коммерческого отдела (примерный разговор):

(Она): Когда ты повесишь картину у секретаря?

(Я): А ты-то тут каким боком?

(Она): Ну ты же не делаешь, когда тебя просит секретарь.

(Я): У меня нет дрели.

(Она): Принеси из дома.

(Я): Дома тоже нет.

(Она): Как нет?! Ты же мужчина!

(Я): Принеси ты из дома, если у тебя есть.

(Она): Откуда у меня? Я – девушка.

(Я): Ты ведь живешь с парнем.

(Она): Ну и что, у него тоже нет.

(Я): Ну а почему у меня тогда она должна быть???

(Она): Ладно, если я принесу, ты повесишь картину?

(Я): Если ты принесешь – то я обязательно ее повешу.

В конечном итоге дрели так и не появилось, зато на корпорате я случайно услышал, как директору рассказывают об отсутствии мужчин в компании, и что я специально говорю, что у меня нет дрели, чтобы ниче не делать. (Вот такой я плохой системный администратор).

Но вообще капец, это когда я установил винду знакомой одной из сотрудниц, и мне на следующий день принесли в качестве благодарности бутылку водки...

Добавлю только одно: в парке 6 серверов, порядка 30 тонких клиентов, удаленно админю филиал в Москве и 3 филиала в Санкт-Петербурге. Короче, все видят, как я просто сижу, пялюсь в монитор. Наверное, делать, значит, нечего.

Мастер на все руки

Стою по правую руку от секретарши, объясняю, куда тыкать, и тут понадобилось пароль ввести. Говорю его и, чтобы не ждать, как она каждую кнопку будет искать, протянув руку, набираю. Секретарша с восхищенным взглядом: «А вы и левой рукой умеете…».

Игнатий Велосипедов

Технологии уборки

И очередной перл от моего начальника. Разговор по телефону:

Н: Сходи в серверную, наведи порядок.

Я: OK. Что конкретно?

Н: Подмети, протри все.

Я: OK.

Н: Сервера все внутри протри влажной тряпочкой.

Я: ???

Н: Ну ты их сначала выключи, тряпочку прополоскай и посильней отожми.

Я: 8-O.

Н: И пылесос там стоит – выкинь его куда-нибудь. Он нам не нужен.

Дальше немая сцена...

О том, что системники (и сервера, в частности) вообще-то пылесосят, и о том, что существуют салфетки для оргтехники, его, наверное, не проинформировали, хотя работает в ИТ-индустрии лет 10 точно.

Люблю разбивать стереотипы.

Lord Grey

Смена растет

Дочка (6 лет) попросила показать, что внутри системника. Принес мать, показал. Дальше пришлось искать оперативку, карты расширения, хард, блок питания, флоповод, шлейфы... Второй день любимая игрушка, проснулась и сразу мать вытащила. До сих пор ковыряется – не оттащить. Приволок ей еще штук 5 матерей, процы... Теперь требует провод питания, монитор, розетку, корпус, клаву и мышь :-))). И главное – чтобы подружка пришла посмотреть. Нашел ей видеокурс по сборке-разборке, смотрела с удовольствием, но с середины – с начала все уже видела.

AngelOfDarknesS _

Трудности последней мили

Предыстория. Значится так: решил я поднять сетку во дворе, написал объявление – «В нашем дворе будет организовываться локальная сеть (ЛВС), всем желающим звонить по номеру такому-то, спросить такого-то» – и расклеил на дверях подъездов.

Сама история: через пару дней мои объявы получили массовый резонанс в старческом обществе. Возвращаемся с другом из лавки и смотрим, картина маслом: управдом собрал вокруг себя аккуратным полукругом группу заинтересованных лиц (разве что на броневик не залез), и начал вещать о том, что ЛВС – это типа домофоны ставить будут при том, что домофоны были установлены и разведены, но в эксплуатацию не введены. Мы с другом бочком-бочком, поближе к стенке :-).

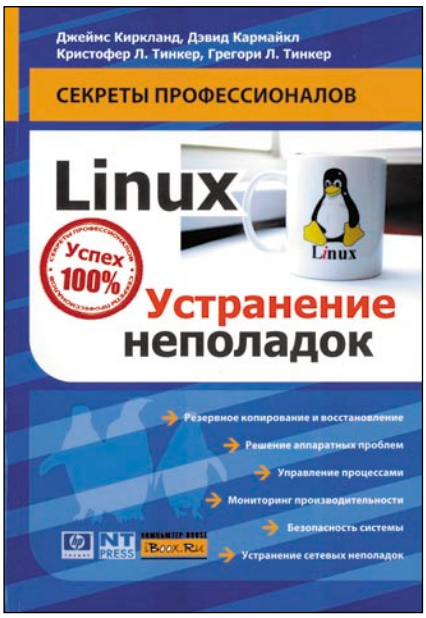
Сеть я все-таки поднял.

PS: Переехал в другой дом, планы те же, писать объявления теперь боюсь, жду рекомендаций.

PPS: В доме тоже много пенсионеров.

claso

По материалам сайта «Сисадмин тоже человек» – http://sysadmin.mail.ru



Linux. Устранение неполадок

Джеймс Киркланд, Дэвид Кармайкл, Кристофер Л. Тинкер, Грегори Л. Тинкер

Перед вами перевод отличной книги, оригинал которой вышел в серии HP Professional Series и написан специалистами центра технической поддержки компании Hewlett-Packard. В качестве главного недостатка можно отметить то, что перевод запоздал на три года и часть приведенной информации явно устарела. Тем не менее издание заслуживает того, чтобы вы проголосовали за него рублем.

Материал ориентирован на два дистрибутива: Red Hat Linux и SUSE Linux. Для работы с книгой вам потребуются навыки администрирования Linux хотя бы на начальном уровне.

В первой главе авторы разбирают механизм начальной загрузки системы, описан как современный загрузчик GRUB, так и LILO. Вторая глава посвящена зависаниям и аварийным остановам системы, а также методам выяснения причин этих неисправностей. Третья и четвертая главы рассказывают об инструментах настройки производительности и мониторинга. В пятой главе речь идет о добавлении в систему устройств SAN, а также устройств на шинах USB и PCMCIA. Шестая глава рассказывает о дисковой подсистеме и связанных с ней возможных проблемах. Седьмая глава посвящена диагностике выхода устройств из строя и их замене. В восьмой речь идет о процессах, их структуре, зависаниях и дампах памяти. Отдельные главы рассказывают о предупреждении неисправностей: резервном копировании и восстановлении и о пакетах cron и at. Одиннадцатая глава повествует о подсистеме печати, однако в книге речь идет в основном о ее устаревших реализациях и соответствующих командах. Хотя безопасность операционной системы представляет собой отдельную и большую область знаний, немного материала по этой теме авторами представлено в двенадцатой главе. Проблемам с сетью посвящена тринадцатая глава. В четырнадцатой рассматриваются проблемы входа в систему, а заключительная, пятнадцатая, рассматривает неполадки системы X Window. При этом речь идет еще о версии XFree86.

- Издательство: *«НТ Пресс»*
- Год издания: *2009*
- Количество страниц: *496*
- ISBN: *978-5-477-00574-1*
- Цена: *≈ 301 руб.*

Python в системном администрировании UNIX и Linux

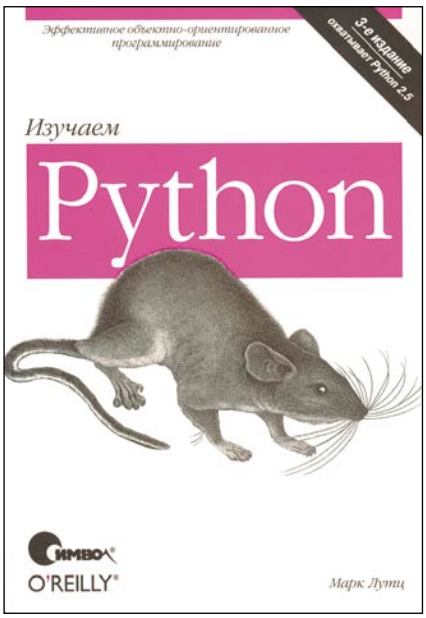
Ноа Гифт, Джереми Джонс

Издание представляет собой своеобразный учебник по программированию для системных администраторов. Для работы с изданием вам потребуется знакомство с предметной областью и желательно умение программировать на каком-либо еще из языков написания скриптов, например bash. Кроме того, вам, безусловно, понадобится машина с установленным Linux или UNIX. Для упрощения работы авторы подготовили свободно распространяемую виртуальную машину на базе Ubuntu, которая включает в себя уже настроенную для работы с книгой среду и приведенные примеры кода.

Первые две главы посвящены введению в язык программирования и настройке интерактивной оболочки IPython. В третьей и четвертой главах рассматриваются методы работы с текстом и генерации отчетов. Пятая глава рассказывает о методах работы в Python с сетями. Шестая глава повествует о работе с файлами и каталогами. Седьмая глава посвящена протоколу SNMP. В восьмой главе рассматриваются специфичные для конкретных операционных систем и дистрибутивов вопросы. Управление пакетами вынесено в девятую главу. Отдельные главы посвящены процессам и многозадачности, созданию графического интерфейса при помощи PyGTK, сохранности данных и работе с командной строкой. В последней главе приведены примеры построения приложений для управления DNS-сервером, работы с протоколом LDAP и зеркалирования FTP-сервера.

- Издательство: *«Символ-Плюс»*
- Год издания: *2009*
- Количество страниц: *512*
- ISBN: *978-5-93286-149-3*
- Цена: *≈ 550 руб.*

Обзор книжных новинок подготовил Андрей Маркелов



Изучаем Python, 3-е издание

Марк Лутц

Русскоязычная версия очередного издания книги из классический серии «Learning...» издательства O'Reilly – отличное подспорье для изучения набирающего обороты языка программирования Python.

В ее основе трехдневные практические курсы автора этой книги. К сожалению, к моменту выхода издания был доступен лишь альфа-релиз Python 3.0, поэтому рассматривается уже не самая актуальная версия – 2.5. Впрочем, это не помешало автору уделить должное внимание уже наступившему будущему Python.

Первая глава книги – лаконичный ликбез, помогающий читателю сформировать понимание языка Python и его предназначения.

Продолжение вводной части посвящено таким темам, как интерпретатор, выполнение программ, работа с Python из консоли.

После этого начинается раздел непосредственного программирования: доступные в Python типы данных и работа с ними, основные инструкции, функции.

Не остался в стороне и такой важный компонент, как модули: рассмотрены их использование и создание, а также пакеты модулей.

Отдельная глава посвящена ООП на Python, где автор рассказывает про классы, шаблоны проектирования с ними и некоторые дополнительные возможности.

Завершающая часть посвящена исключениям и их использованию.

В конце каждой главы книги приводится краткое резюме, а также список вопросов для закрепления материала (с ответами).

В целом, издание наилучшим образом подойдет тем, кто решил освоить Python с нуля, но может показаться излишне подробным для более опытных программистов.

- Издательство: *«Символ-Плюс»*
- Год издания: *2009*
- Количество страниц: *848*
- ISBN: *978-5-93286-138-7*
- Цена: *≈ 800 руб.*

Книга предоставлена издательством «Символ-Плюс».

FreeBSD 6. Полное руководство

Брайан Таймэн

Автором этого подробного руководства, посвященного FreeBSD, является человек, который, во-первых, явно не равнодушен к этой операционной системе, а во-вторых, известен своими работами по продукции Apple: Mac OS X, iTunes, iPod...

С одной стороны, довольно необычно видеть серьезную книгу по администрированию от любителя подобных разработок, но с другой – можно найти и вполне логичное тому объяснение, учитывая исторические корни Mac OS X.

Тем не менее это отложило определенный отпечаток на стиль изложения и сам материал: текст зачастую ориентирован на неопытных пользователей, которые впервые знакомятся с миром FreeBSD.

Главным же достоинством книги является широкий охват тем: после введения, в котором, помимо прочего, FreeBSD сравнивается с другими ОС, автор подробно описывает процесс установки системы.

Следующий шаг – использование FreeBSD как системы для настольного ПК, подразумевающее как общие аспекты работы с системой, так и знакомство с базовыми приложениями (консольными и графическими).

Затем читателя обучают работе в командной оболочке и связанным с ней основам программирования: на UNIX shell и на Perl.

Но основное содержимое книги отведено двум последующим частям: системному администрированию и работе с сетью. В них затрагиваются такие темы, как файловая система UFS, пользователи/группы, печать (lpd), конфигурация ядра, обновление системы, настройка сетевых служб, PPP, DHCP, BIND, Sendmail, связка Apache + PHP + Perl + MySQL/PostgreSQL, FTP-сервер, Samba. К руководству прилагается DVD, на котором можно найти дистрибутив FreeBSD 6.1 с коллекцией портов и дополнительными пакетами ПО.

- Издательство: *«Вильямс»*
- Год издания: *2008*
- Количество страниц: *1056*
- ISBN: *978-5-8459-0741-7*
- Цена: *≈ 705 руб.*

Книга предоставлена издательством «Вильямс».

Обзор книжных новинок подготовил Дмитрий Шурупов

Редакционная подписка
для физических лиц

- Вы можете оформить подписку только на **российский** адрес.
- При заполнении квитанции **обязательно РАЗБОРЧИВО укажите фамилию, имя, отчество полностью, почтовый индекс и адрес получателя (область, город, улица, номер дома, номер квартиры), контактный телефон.**
- Журнал высылается почтой заказной бандеролью только после поступления денег на расчетный счет и **копии заполненного и оплаченного бланка, отправленной в редакцию по факсу: (495) 628-82-53 (доб. 120) или на электронный адрес: subscribe@samag.ru**.

ИЗВЕЩЕНИЕ

Кассир

ООО "С 13" Форма № ПД-4
ИНН 7708654814 / КПП 770801001
Р. сч. 40702810300080001868 К. сч. 30101810100000000787
ОАО «УРАЛСИБ» г. Москва БИК 044525787
Коды: по ОКПО 84027582, по ОКОПФ 65

Вид платежа: Редакционная подписка на журнал "Системный администратор" за 2009 г.

01	02	03	04	05	06	07	08	09	10	11	12
X	X	X	X	X	X	X	X	X	X	X	X

Дата _____ Сумма платежа: 2400 руб. 00 коп.

Информация о плательщике:

_____
(Ф. И. О. почтовый индекс, адрес и телефон)

_____

_____

Подпись _____

КВИТАНЦИЯ

Кассир

ООО "С 13" Форма № ПД-4
ИНН 7708654814 / КПП 770801001
Р. сч. 40702810300080001868 К. сч. 30101810100000000787
ОАО «УРАЛСИБ» г. Москва БИК 044525787
Коды: по ОКПО 84027582, по ОКОПФ 65

Вид платежа: Редакционная подписка на журнал "Системный администратор" за 2009 г.

01	02	03	04	05	06	07	08	09	10	11	12
X	X	X	X	X	X	X	X	X	X	X	X

Дата _____ Сумма платежа: 2400 руб. 00 коп.

Информация о плательщике:

_____
(Ф. И. О. почтовый индекс, адрес и телефон)

_____

_____

Подпись _____

Российская Федерация

- Подписной индекс: годовой – **20780**, полугодовой – **81655**
  Каталог агентства «**Роспечать**»
- Подписной индекс: годовой – **88099**, полугодовой – **87836**
  Объединенный каталог «**Пресса России**»
  Адресный каталог «**Подписка за рабочим столом**»
  Адресный каталог «**Библиотечный каталог**»
- Альтернативные подписные агентства:
  Агентство «**Интер-Почта**» (495) 500-00-60, курьерская доставка по Москве
  Агентство «**Вся Пресса**» (495) 787-34-47
  Агентство «**Курьер-Прессервис**»
  Агентство «**ООО Урал-Пресс**» (343) 375-62-74
  ЛинуксЦентр www.linuxcenter.ru
- Подписка On-line
  http://www.arzi.ru
  http://www.gazety.ru
  http://www.presscafe.ru

СНГ

В странах СНГ подписка принимается в почтовых отделениях по национальным каталогам или по списку номенклатуры «**АРЗИ**»:

- **Азербайджан** – по объединенному каталогу российских изданий через предприятие по распространению печати «**Гасид**» (370102, г. Баку, ул. Джавадхана, 21)
- **Казахстан** – по каталогу «**Российская Пресса**» через ОАО «**Казпочта**» и ЗАО «**Евразия пресс**»
- **Беларусь** – по каталогу изданий стран СНГ через РГО «**Белпочта**» (220050, г. Минск, пр-т Ф. Скорины, 10)
- **Узбекистан** – по каталогу «**Davriy nashrlar**» российские издания через агентство по распространению печати «**Davriy nashrlar**» (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)
- **Армения** – по списку номенклатуры «**АРЗИ**» через ГЗАО «**Армпечать**» (375005, г. Ереван, пл. Сасунци Давида, д. 2) и ЗАО «**Контакт-Мамул**» (375002, г. Ереван, ул. Сарьяна, 22)
- **Грузия** – по списку номенклатуры «**АРЗИ**» через АО «**Сакпресса**» (380019, г. Тбилиси, ул. Хошараульская, 29) и АО «**Мацне**» (380060, г. Тбилиси, пр-т Гамсахурдия, 42)
- **Молдавия** – по каталогу через ГП «**Пошта Молдовей**» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
  по списку через ГУП «**Почта Приднестровья**» (MD-3300, г. Тирасполь, ул. Ленина, 17)
  по прайс-листу через ООО Агентство «**Editil Periodice**» (МД-2012, г. Кишинев, бул. Штефан чел Маре, 134)
- Подписка для **Украины**:
  Киевский главпочтамт
  Подписное агентство «**KSS**», тел./факс (044)464-0220

Ф.СП-1

Министерство связи РФ

АБОНЕМЕНТ на журнал _____ (индекс издания)

Системный администратор

Количество комплектов: _____

на 200 год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда _____ (почтовый индекс) _____ (адрес)

Кому _____ (фамилия, инициалы)

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ	место	литер

на журнал _____ (индекс издания)

Системный администратор

Стоимость: по каталогу _____ руб. _____ коп.; за доставку _____ руб. _____ коп.

Количество комплектов: _____

на 200 год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда _____ (почтовый индекс) _____ (адрес)

Кому _____ (фамилия, инициалы)

Подписные индексы:

20780*
+ диск с архивом статей 2008 года

81655**
без диска

по каталогу агентства «Роспечать»

88099*
+ диск с архивом статей 2008 года

87836**
без диска

по каталогу агентства «Пресса России»

* Годовой
** Полугодовой
*** Диск вкладывается в февральский номер журнала, распространяется только на территории России

СИСТЕМНЫЙ АДМИНИСТРАТОР
№4(77), Апрель, 2009 год

УЧРЕДИТЕЛИ
Частные лица

РЕДАКЦИЯ
Генеральный директор
Владимир Положевец
Ответственный секретарь
Наталья Хвостова
sekretar@samag.ru
Технический редактор
Владимир Лукин
Главный редактор электронного приложения «Open Source»
Дмитрий Шурупов
osa@samag.ru

Внештатные редакторы
Алексей Барабанов
Александр Емельянов
Кирилл Сухов
Василий Гусев
Андрей Бирюков
Олег Щербаков
Александр Слесарев
Сергей Супрунов

РЕКЛАМНАЯ СЛУЖБА
тел./факс: (495) 628-8253 (доб. 120)
Дарья Зуморина
reclama@samag.ru
Евгения Тарабрина
expo@samag.ru

Верстка и оформление
maker@samag.ru
Дизайн обложки
Дмитрий Репин

По вопросам распространения обращайтесь по телефону:
Светлана Зобова
(495) 628-8253 (доб. 120)

107045, г. Москва,
Ананьевский переулок, дом 4/2, стр. 1
тел./факс: (495) 628-8253
Сайт журнала: www.samag.ru

ИЗДАТЕЛЬ
ООО «С 13»
Отпечатано типографией
ООО «Периодика»
Тираж 17000 экз.
Тираж электронной версии 62000 экз.

Журнал зарегистрирован в Министерстве РФ по делам печати, телерадиовещания и средств массовых коммуникаций (свидетельство ПИ № 77-12542 от 24 апреля 2002 г.).

Вы знаете, как бороться с «Просачивающейся Адварью»? Применяете «Чарующий скрипт»?

Редакция журнала «Системный администратор» представляет вам новый админский сувенир для истинных знатоков своего дела – карточную игру «АУТСОРСЕР».

В ходе игры участники тянут из колоды карты «Проблем», с которым им предстоит бороться один на один или с помощниками, используя подручные средства. Успешное решение «Проблемы» добавляет игроку уровни. Если вы не считаете себя добрым и милым, то для вас в игре предусмотрена специальная возможность – сделать гадость другому участнику и обойти его в погоне за уровнями.

Победителем становится тот, кто быстрее всех доберется до 10 уровня. Остальные подробности об игре, «Чарующем скрипте», «МегаУтилите» и «Клановом коктейле» вы сможете узнать из правил игры.

«АУТСОРСЕР» – это пародия на жизнь, которая позволит вам ощутить всю прелесть аутсорсинга... но без всей словесной мишуры, типа, «утром стулья, вечером деньги...»!

Приобретайте игру «АУТСОРСЕР» в редакции.