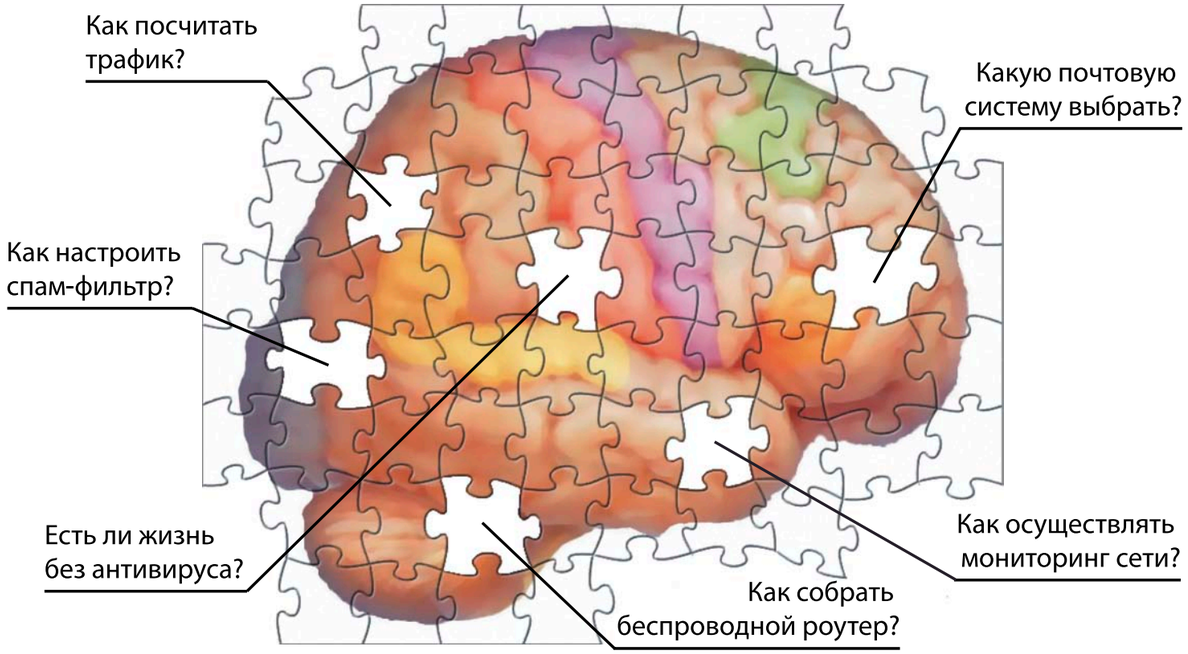
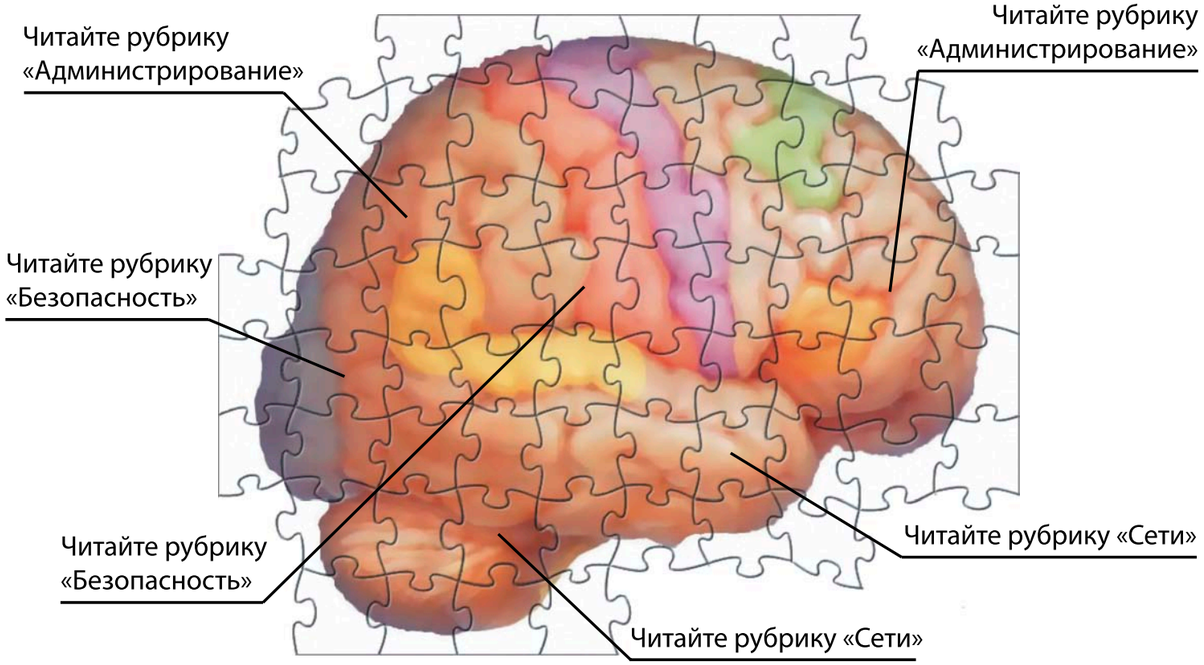
Какими мыслями занят системный администратор, который не читает свой журнал?



Остальные системные администраторы знают, где искать ответы на вопросы!



Журнал «Системный администратор»
www.samag.ru

№9(82) сентябрь 2009 Системный администратор

Системный администратор

ежемесячный журнал www.samag.ru

№9(82) сентябрь 2009

Аутсорсинг: ищем сотрудников или партнеров?

Терминальные службы в Windows Server 2008

Комплексное решение: виртуализация + отказоустойчивый кластер

Домашний хостинг: используем сервисы динамического DNS

Новые возможности Nmap 5.00 – исследуем безопасность сетей

Как стать системным администратором?

Альтернативы MS Project: пробуем Open Source-решения

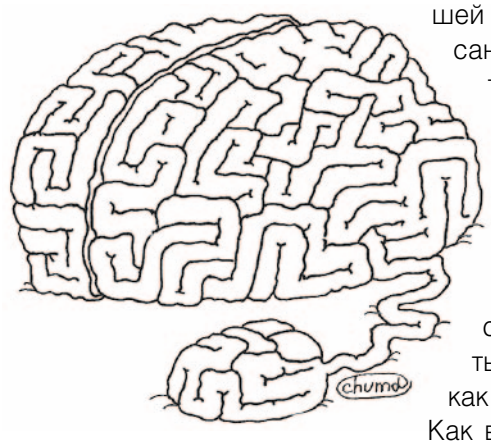
Джефф Безос – гений интуиции

ISSN 1813-5579

Linux center www.linuxcenter.ru

Свой среди чужих

Как стать системным администратором? На этот простой вопрос, оказалось, нет однозначного ответа. В России уже трудятся десятки тысяч сисадминов, однако нигде – ни в школах, ни в вузах, ни в колледжах – нет такой специализации. Овладевать азами и премудростями претенденту на эту работу приходится самостоятельно. Возможно, поэтому еще существует «кое-где у нас порой» пренебрежительное отношение к сисадмину как необязательному приложению к компьютерной сети. Ну ходит, ну чего-то там ковыряется… А в общем-то, непонятно, чем занимается весь день. И если наступает кризис, такой горе-начальник в поисках резервов для сокращения недрогнувшей рукой вычеркивает из штатного расписания должность системного администратора. Мол, обойдемся. Справимся сами.



Но нет, не обходятся, не справляются. «Железо» мстит за неуважение к своему хозяину и другу. И тогда сисадмины вновь возвращаются.

Так как же? Где научиться столь важному и нужному делу? Мы попросили рассказать об этом наших читателей. Их ответы вы найдете в дискуссии номера о том, как стать системным администратором.

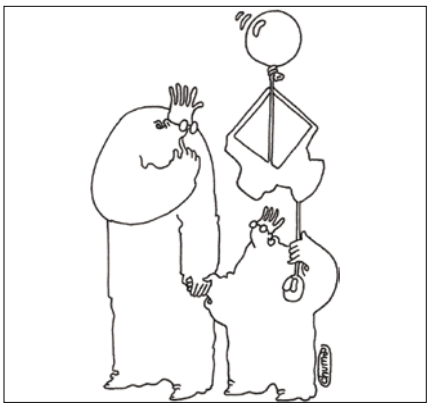
Как в любой профессии, кроме знаний, вам потребуются определенные качества характера: терпение, креативность, умение не пасовать перед трудностями, находить решение в, казалось бы, безвыходной ситуации, интуиция, наконец. И порядочность, наверное. Потому что, как писали мы в августовском номере, от сисадмина до хакера или до лидера компании – всего один шаг. Все решает собственный выбор человека.

Кстати, среди читателей «СА» очень много руководителей среднего и высшего звена, которые начинали свой карьерный рост с позиции системного администратора.

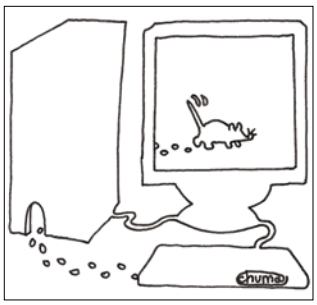
Когда я узнаю о том, что гендиректор известной компании был когда-то сисадмином, я доверяю ему больше. Значит, он знает досконально то, о чем говорит. Потому что системный администратор – это человек творческий, увлеченный, многогранный. Он прошел сам свои университеты, докопался до сути профессии и не задержался на нижних позициях.

Но даже если кто-то из них решил остаться просто системным администратором, он в конце концов становится таким суперпрофессионалом, о котором мечтает каждый начальник.

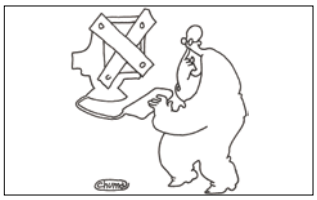
Галина Положевец,
главный редактор



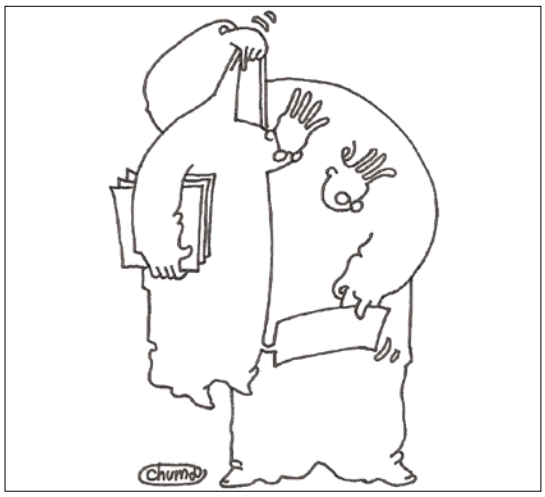
06



20



54



68

В Рунете появился сервис IT4All, разработанный компанией Softkey для участников софтверного рынка

Визитка

ДМИТРИЙ ШУРУПОВ,
ведущий рубрики

Решение для контроля доступа в Сеть

Компания Entensys выпустила новую версию популярного прокси-сервера UserGate Prosy & Firewall. UserGate предназначен для решения задач, актуальных для большинства компаний: снижение нагрузки на сеть и регулирование доступа в Интернет. UserGate является эффективной альтернативой дорогостоящему программному и аппаратному обеспечению и предназначен для использования в компаниях малого и среднего бизнеса.

Программа позволяет контролировать доступ в Интернет пользователей локальной сети. Модуль фильтрации трафика BrightCloud блокирует доступ к нежелательным ресурсам как в отдельности, так и по категориям сайтов. Это существенно сокращает затраты компании на нецелевой трафик, повышая производительность работы сотрудников и снижая риски попадания вредоносного ПО в локальную сеть. UserGate обеспечивает комплексную защиту локальной сети благодаря встроенному межсетевому экрану и наличию двух антивирусных модулей от «Лаборатории Касперского» и Panda Security, а также позволяет контролировать приложения, установленные на клиентских машинах, разрешая или запрещая приложениям выход в Интернет. Кроме того, продукт предоставляет администраторам полную статистику о посещаемости ресурсов пользователями. EOF

Softkey запустил социальную сеть для участников ИТ-рынка – IT4All

Недавно в Рунете появился уникальный сервис IT4All (http://it4all.ru), разработанный компанией Softkey для всех участников софтверного рынка. IT4All, онлайн-центр экспертизы и продуктового консалтинга, предлагает разработчикам ПО, дилерам, пользователям и VAR-ам единую площадку для общения и сотрудничества. Для корпоративных пользователей IT4All – это гарантия организации квалифицированной продажи ПО. Они смогут не только вести прямой диалог с разработчиками и поставщиками, но и получить помощь авторизованных специалистов в поиске решения под конкретную задачу, выбрать из ряда предложений наиболее оптимальное. Большая роль в проекте отводится независимым экспертам, которые могут не только советовать пользователю то или иное программное обеспечение, но также рекомендовать любого из продавцов софта. Работая в центре событий отрасли, участвуя в дискуссиях, предлагая свои варианты решений, эксперты повышают собственную ИТ-компетенцию. Лучшие из них станут обладателями сертификатов ведущих вендоров. EOF

VMware покупает SpringSource

VMware, крупный мировой производитель программных решений для виртуализации, объявил о достижении соглашения по поглощению компании SpringSource, специализирующейся в области корпоративных решений на базе Java. SpringSource хорошо известна в мировом Open Source-сообществе благодаря созданию фреймворка Spring Framework и существенной поддержке сервера приложений Apache Tomcat и популярного HTTP-сервера Apache. Кроме того, за последний год компания поглотила G2One, авторов Groovy и Grails, и Hyperic, создавшую ПО для управления ИТ-инфраструктурой предприятия.

Теперь, когда все это наследие SpringSource станет частью VMware, ожидается, что у компании появится возможность представить новые решения, «которые позволят корпоративным заказчикам более эффективно создавать, эксплуатировать приложения и управлять ими как в привычном внутреннем окружении, так и в рамках облачных вычислений».

Стоимость сделки составит в общей сложности около 420 миллионов американских долларов. Поглощение уже одобрено акционерами SpringSource, и его завершение ожидается в третьем квартале этого года.

Во второй половине августа также приблизилось время другого поглощения – куда более масштабного и заметного для всей ИТ-индустрии: министерство юстиции США одобрило сделку между Oracle и Sun Microsystems, о которой впервые стало известно еще в апреле этого года. EOF

Linux Foundation обновила статистику по разработке Linux-ядра

Организация Linux Foundation обнародовала обновленный документ «Разработка Linux-ядра: Как быстро она проходит, кто ей занимается и кто ее спонсирует?». В нем собраны различные статистические сведения по разработке Linux-ядра, которые помогают организации демонстрировать стремительное развитие свободной ОС.

В новый отчет была добавлена информация о шести последних релизах Linux-ядра (начиная с версии 2.6.24). Этот вклад дополнил картину статистикой за 500 последних дней разработки, подтвердившей продолжающийся рост активности разработки Linux-ядра. Так, например, если в ядре версии 2.6.11 (2 марта 2005 года) было представлено 3616 изменений (2,18 изменения в час), а в 2.6.21 (25 апреля 2007 года) – 5016 изменений (2,58 в час), то в 2.6.30 (9 июня 2009 года) их уже 11989 (6,4 в час).

Одновременно с этим заметно и увеличение числа компаний, работающих над Linux-ядром: в Linux 2.6.11 таковых было 68, в 2.6.21 – 143, а в 2.6.30 – 240. При этом небольшие изменения претерпел и список компаний/организаций, которые вносят свой вклад в развитие Linux-ядра. На первом месте – Red Hat (12,3%), на втором – IBM (7,6%), на третьем – Novell (7,6%), а далее следуют Intel (5,3%), Oracle (2,4%), Linux Foundation (1,6%), SGI (1,6%), Parallels (1,3%), Renesas Technology (1,3%).

Актуальная версия документа в электронном виде доступна по адресу http://www.linuxfoundation.org/publications/whowriteslinux.pdf. EOF

Участвуй сам и расскажи друзьям! Как получить Адмиский приз?

Редакция «СА» продолжает разыгрывать Адмиский приз, и приглашает к участию новых игроков. Вам необходимо зарегистрироваться на сайте www.samag.ru и активировать код, который можно получить, купив номера журнала (№№7-12, 2009). Чем больше у вас заветных кодов, тем выше шансы стать победителем розыгрыша. Дополнительные коды смогут получить самые активные участники форума на сайте www.samag.ru.

Успехов и удачи!

Адмиский Приз

Розыгрыш будет проходить в три этапа:

I – участвуют коды из журналов №7, 8, 9, полученные с июля по сентябрь 2009 г.

II – участвуют коды из журналов №10, 11, 12, полученные с октября по декабрь 2009 г.

III – участвуют коды из всех шести номеров журнала за 2-е полугодие 2009 г.

Адмиский Приз

Админский Приз

Призы:

I этап:
1 место — приз-сюрприз
2 место — учебные курсы
3 место — пакет программного обеспечения
4 место — почтовый сервер на 50 пользователей
5 место — виртуальные выделенные серверы

II этап:
1 место — приз-сюрприз
2 место — учебные курсы
3 место — пакет программного обеспечения
4 место — почтовый сервер на 50 пользователей
5 место — виртуальные выделенные серверы

III этап:
1 место — приз-сюрприз
2 место — учебные курсы
3 место — пакет программного обеспечения
4 место — почтовый сервер на 50 пользователей
5 место — виртуальные выделенные серверы

Специальный утешительный приз — электронная книга

Ваш код для участия
в розыгрыше призов:

Системный администратор

A-SYS.RU

RUSONYX
Скорость. Надежность. Поддержка.

ideco

allsoft.ru®
группа компаний Softline

KERIO

Админский Приз

Визитка

ОКСАНА РОДИОНОВА, журналист «Учительской газеты» специально для «Системного администратора»

Куда приводят мечты? В сети

Хороший системный администратор любит учиться, умеет искать информацию, обожает читать скучные книжки

Информационные технологии – по-прежнему популярная сфера деятельности? Пожалуй, что так. Например, в Германии программисты и сетевые администраторы входят в десятку самых престижных специальностей. Предложений на рынке труда для них с каждым годом все больше. И в России, заглянув на любой сайт, где размещаются вакансии, в любую газету объявлений о найме на работу, можно найти вакансии системных администраторов. Другое дело, что эта должность не приносит ощутимых доходов, да и особого карьерного роста сисадмину никто не обещает. Чем же все-таки привлекает эта специальность? Как становятся системными администраторами? Где учатся? Что нужно знать, чтобы стать хорошим сисадмином? На эти вопросы мы и попробуем ответить. Естественно, с помощью профессионального сообщества.

«Хочу стать сисадмином». И что дальше?

Вспомним день вчерашний. «Изначально не было такой профессии. Были инженеры-эксплуатационщики – скорее специалисты по электронике и системотехнике (и дипломы у них были соответствующие). Но, как оказалось, они совершенно ничего не понимали в операционных системах, без которых компьютер был не более чем грудой пыльного и шумного железа. Можно даже сказать, что именно развитие операционных систем создало профессиональную нишу для сисадминов.

«Меня интересовали как раз операционные системы. И я стал смещать сферу своих обязанностей с прикладного программирования на эксплуатацию ОС. Однажды я просто стал начальником ЭВМ, что в табели о рангах того времени соответствовало системному администратору». Так рассказывает о себе системный администратор и аутсорсер Алексей Барабанов. Очень многие нынешние сисадмины со стажем начинали работать в этой сфере просто потому, что было интересно. А хобби переросло в работу.

Другое дело – современные молодые люди. Может быть, для многих из них налаживать сети и является хобби, но заниматься любимым делом хотелось бы все-таки за нормальную плату. Знать, какие бонусы получишь на этой работе. Сколько придется вкалывать? И как скоро можно стать мастером, которого отрывают с руками и ногами компании?

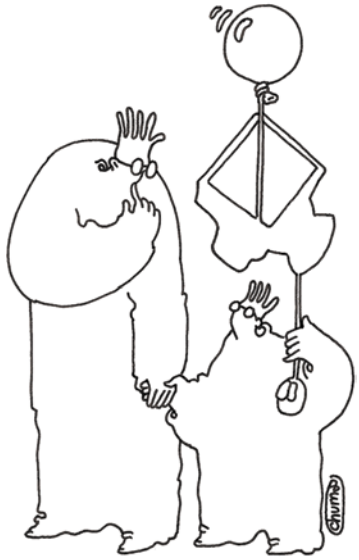
Опытные коллеги предпочитают сразу взять и отвадить мечтателей. Они советуют… выбрать другую профессию, потому как «технические специалисты в России вынуждены жить более чем скромно. При тех же самых интеллектуальных способностях и тех же трудозатратах в других сферах, например, в финансах, торговле, обслуживании, можно добиться гораздо большего материального благополучия». Лев Мышкин, системный администратор, суров, но реалистичен. Тем, кто упрямо желает добиться своего, он рекомендует прочитать фантастическую повесть Айзека Азимова «Профессия».

А вот еще одно предостережение от Андрея Бешкова, эксперта по инфраструктуре компании «Майкрософт Россия»: «Эта работа не предполагает особой благодарности и почета от пользователей и руководства – ведь лучший «админ» – это невидимый «админ».

И, наконец, пожелание от Сергея Супрунова, инженера электросвязи, – сохранить энтузиазм, ведь без него сисадминство – «не такая уж и привлекательная специальность».

Если вас не смущает, что в перечне профессий «системный администратор» не значится, что в вашей трудовой книжке появится запись «администратор вычислительной сети» или «инженер-программист», приступайте к воплощению своей мечты.

Если человек еще на школьной скамье решил стать сисадмином, пожалуй, у него неплохие шансы. Опытные товарищи советуют попробовать себя в деле, устроившись во время каникул на любую временную работу в сфере ИТ – в контору, где ремонтируют технику, к местному провайдеру и так далее. Можно начать даже с незавидной должности разносчика кофе операторам «коллдеска». Главное – внимательно приглядываться и прислушиваться.



Сожжешь пару компов, тебе по «кумполу» настучат — мигом всё выучишь за ночь

Другой важный совет – учиться! Читать книги по компьютерной тематике и пробовать воплощать на практике теорию. Какие книги читать и с кем обсуждать неизбежно возникающие вопросы – это очень важно. Начинающему сисадмину нужен свой гуру – знакомый компьютерщик, сообщество на форуме или irc-чате, учитель информатики в школе. Наконец, очень хороший педагог – жизнь. Знающие люди говорят: «Сожжешь пару компов, тебе по «кумполу» настучат – мигом всё выучишь за ночь».

А вот и литература, список взят с одного из популярных профильных форумов.

Что нужно прочитать:

- Питер В. Ален. «Оптимизация и защита Linux-сервера».
- Э. Таненбаум. «Компьютерные сети».
- А. Поляк-Брагинский. «Локальные сети».
- Немет Эви, Снайдер Гарт, Хейн Трент. «Руководство администратора Linux».

Начинать надо с принципов построения локальных и глобальных сетей, работы стека TCP\IP. Это базис. А дальше сужать область специализации...

Практиковаться можно с решения такой задачи – организовать сеть из двух компьютеров. Взять свеженькие дистрибутивы Linux или Windows, установить и настроить все программы, входящие в стандартный набор, понять, как они работают. Поднимать виртуальные сети и находить выход из реальных проблем, описанных на форумах.

Выпускнику школы или студенту прямая дорога устроиться к местному провайдеру или в какой-нибудь офис, где есть не слишком большое количество компьютеров (15-20), эникеем, то бишь специалистом среднего звена в отделе ИТ. Он решает мелкие технические проблемы – от подключения компьютерной периферии и настройки интерфейса ПО пользователя до замены картриджей в принтерах и объяснений альтернативно одаренным пользователям, какую кнопку надо давить в сложной ситуации. Вырасти до помощника сисадмина, а затем стать полноправным системным администратором.

Станислав Шпак, системный администратор, рассказывает, что на последних курсах дипломный руководитель предложил ему работу «железячником» в организации на полсотни компов (три подсети и домен). Привлекло, что параллельно можно будет ознакомиться с сетями и задавать вопросы тамошнему сисадмину. «Однако через два месяца сисадмин, не сумев уладить с руководством вопрос повышения зарплаты, уволился. И я остался и за железячника и за сисадмина. Поначалу было очень туго, рассчитывать не на кого, и пришлось много читать как специальной литературы, так и руководств администратора, но все равно это не спасало от «грабель». Спустя год, когда я уже достаточно хорошо освоился с вверенным мне доменом, сетевыми сервисами и аппаратно-кабельным хозяйством, я подрабатывал на четверть ставки преподавателем ИТ-технологий. Спустя два года я вырос из масштабов поддерживаемой сети и сменил работу, попав в крупную фирму с большим парком компьютеров, несколькими доменами и распределенным территориальным положением. Прошел сертификацию Microsoft и получил статус MCSE по Windows Server 2000 и 2003. Сейчас я отношу себя к тем людям, для кого работа в радость». Вот такая история.

А карьера… Она может выглядеть так: сисадмин – старший сисадмин (опционально) – аналитик (прикладные системы, сетевая безопасность) – эксперт – ведущий эксперт по информационной безопасности. Кто-то уезжает работать за рубеж. Это дает нормальные деньги, к тому же российских сисадминов, например в Соединенных Штатах, расхватывают, как горячий хлеб, они универсальны в отличие от местных узких специалистов.

«Хочу стать хорошим сисадмином». Что нужно знать и уметь?

Системный администратор – это не профессия, а целый список профессий.

Мои эксперты на вопрос, что нужно знать и уметь сисадмину, а также, где эти знания приобрести, откликнулись целым списком:

- Уметь читать скучные книжки.
- Общаться с более опытными коллегами.
- Важно умение добывать и фильтровать информацию. Сергей Супрунов считает: «Умение грамотно задать вопрос (в том числе и поисковику), а из полученных ответов выбрать то, что нужно, как раз и определяет успешность или неуспешность работы сисадмина. Чтобы правильно сформулировать вопрос, очень пригодятся базовые знания – основы файловых систем, сетевых протоколов, принципы сборки и установки пакетов, основы программирования...»
- Знать свою инфраструктуру как в лицо, так и изнутри. Информацию можно получить от коллег или из печатных материалов.
- Знать английский язык.

Павел Закляков, ИТ-специалист, предлагает следующий «скелет» нужных знаний: «Общее понимание схемотехники, алгоритмов, умение программировать, знание строения ОС и сетей. А дальше пойдёт и теория связи, и помехоустойчивое кодирование, и оптимальный приём, и теория вероятностей, и логика, комбинаторика, дискретная математика, и гуманитарные науки будут не на последнем месте».

Конечно, все эти знания и умения должны лечь на «правильную» почву. То есть человек, который решил стать системным администратором, конечно, должен иметь технический склад ума, быть готов к творчеству, получать безумное удовольствие от изучения чего-то нового, даже если это проблема, из-за которой «упал» сервер и у босса случился микроинфаркт. Сюда можно добавить стрессоустойчивость, упорство, терпение, умение грамотно распределять доступное время на все имеющиеся задачи.

Идеальный сисадмин не страдает от отсутствия внимания к своей персоне. «Чем меньше заметна его деятельность, тем лучше он выполняет свою работу», – уверен Алексей Барабанов. Он не жалеет о потраченном свободном времени, ведь дела могут отвлечь его и от праздников, и от отпуска, и от собственного дня рождения. В общем, надо любить свою работу, иначе ничего не получится.

Наконец, многие считают, что системный администратор все-таки должен иметь высшее образование.

«Хочу выучиться на сисадмина». Куда поступать?

Не будем долго приводить здесь аргументы «за» и «против» высшего образования. Много ли даст настоящему специалисту, да еще талантливому человеку, вуз? Но это с какой стороны посмотреть. Вуз (хороший) дает базу, позволяет систематизировать знания, помогает обрасти связями, обзавестись собственным кругом общения. Наконец, работодатели при приеме на работу требуют эту самую «корочку», особенно в крупных компаниях.

«Необходимо наличие высшего технического образования (приоритет будет отдан программистам) либо математического или авиационного», – говорит руководитель отдела управления персоналом компании APC by Schneider Electric в России и странах СНГ Галина Нарушева. И даже если берут человека без высшего образования, особо одаренного, распространяться об этом не станут…

«Диплом не является сколько-нибудь значимым плюсом. К нам пришел соискатель из МГТУ им. Баумана, у него красный диплом по специальности «Сети». В ответ на вопрос, что такое DNS, раздалось невнятное мычание…» – рассказал мне представитель фирмы, пожелавший остаться анонимным.

Однако есть и другое мнение: «Наличие высшего образования или то, что будущий системный администратор еще проходит обучение в вузе, это огромный плюс. Желательно, чтобы это был технический вуз. Не всегда важно, какая у претендента специальность – она может иметь косвенное отношение не то что к системному администрированию, но и вообще к информационным технологиям. Высшее образование служит своего рода индикатором того, что человек способен решать сложные неординарные задачи, пользоваться литературой, что, собственно, и нужно системному администратору», – уверен старший специалист группы системного администрирования ЗАО «Диалог-Наука» Александр Злобин.

Какой конкретно вуз лучше выбрать? Ответ на этот вопрос пару лет назад дал журнал «Куда пойти учиться», он актуален и сегодня: «Технические вузы предлагают четыре специальности, наиболее соответствующие профессии системного администратора.

Что в дипломе?

«Автоматизированные системы обработки информации и управления (АСОИУ)». Эту специальность ввели в свои учебные планы два десятка столичных вузов, среди них: МАТИ – Российский государственный технологический университет им. К.Э. Циолковского (РГТУ), Московский государственный институт стали и сплавов (МИСиС), Российский государственный социальный университет (РГСУ).

«Вычислительные машины, комплексы, системы и сети (ВМСС)». Профессионалов в этой области готовят около десяти вузов: Московский энергетический институт (МЭИ ТУ), Московский государственный институт электронной техники (МИЭТ), Московский инженерно-физический институт (государственный университет) (МИФИ) и др.

«Программное обеспечение вычислительной техники и автоматизированных систем». Обучение ведут восемь высших учебных заведений, в том числе МИЭТ, Российский государственный социальный университет (РГСУ), Московский государственный технологический университет «СТАНКИН».

«Математическое обеспечение и администрирование информационных систем». Это самая молодая специальность, поэтому существует пока только в семи вузах, например: в Московском городском психолого-педагогическом университете (МГППУ), Российском новом университете (РосНОУ), Московском государственном университете экономики, статистики и информатики (МЭСИ)».

Любопытный эксперимент однажды поставили на форуме sysadmins.su. Было предложено назвать дисциплины, которые могут войти в учебный план мифического факультета по подготовке системных администраторов. Всем миром составили такой план.

Где учиться в России

Город	Вуз	Специальность	Сайт
Москва	Московский государственный институт электронной техники	230101 Вычислительные машины, комплексы, системы и сети	www.miee.ru
		230100 Информатика и вычислительная техника (бакалавр, магистр)	
	Московский государственный технический университет им. Н.Э. Баумана	230101 Вычислительные машины, комплексы, системы и сети	www.bmstu.ru
		230200 Информационные системы (бакалавр)	
		230100 Информатика и вычислительная техника (магистр)	
	Московский энергетический институт (технический университет)	230101 Вычислительные машины, комплексы, системы и сети	www.mpei.ru
		230201 Информационные системы и технологии	
Санкт-Петербург	Санкт-Петербургский государственный университет информационных технологий, механики и оптики	230101 Вычислительные машины, комплексы, системы и сети	www.ifmo.ru
		230201 Информационные системы и технологии	
Екатеринбург	Уральский государственный технический университет	230200 Информационные системы (бакалавр)	www.ustu.ru
		230201 Информационные системы и технологии (инженер)	
Нижний Новгород	Нижегородский государственный университет им. Н.И. Лобачевского	010400 Информационные технологии (бакалавр, магистр)	www.unn.ru
		Курс «Системное администрирование» (2 семестра, ориентирован на студентов, аспирантов, выпускников вузов и средних специальных заведений)	
Самара	Самарский государственный аэрокосмический университет им. ак. С.П.Королева	010400 Информационные технологии	www.ssau.ru
		230102 Автоматизированные системы обработки информации и управления	
Омск	Омский государственный технический университет	230101 Вычислительные машины, комплексы, системы и сети	www.omgtu.ru
		230100 Информатика и вычислительная техника (Бакалавр, магистр)	
Казань	Казанский государственный технический университет им. А.Н.Туполева	230101 Вычислительные машины, комплексы, системы и сети	www.kai.ru
		230100 Информатика и вычислительная техника (Бакалавр, магистр)	
Магнитогорск	Магнитогорский государственный технический университет им. Г.И.Носова	230105 Программное обеспечение вычислительной техники и автоматизированных систем управления	www.magtu.ru
		230100 Информатика и вычислительная техника (Бакалавр, магистр)	
Краснодар	Кубанский государственный технологический университет	220201 Управление и информатика в технических системах	www.kubstu.ru
		230105 Программное обеспечение вычислительной техники и автоматизированных систем	
		230101 Вычислительные машины, комплексы, системы и сети	
Уфа	Уфимский государственный авиационный технический университет	230101 Вычислительные машины, комплексы, системы и сети	www.ugatu.ac.ru
		220400 Программное обеспечение вычислительной техники и автоматизированных систем	
Новосибирск	Новосибирский государственный технический университет	010503 Математическое обеспечение и администрирование информационных систем	www.nstu.ru
		220200 Автоматизация и управление (магистр)	
	Сибирский государственный университет телекоммуникаций и информатики	230100 Информатика и вычислительная техника (бакалавр)	www.neic.nsk.su
		230101 Вычислительные машины, комплексы, системы и сети	
		230105 Программное обеспечение вычислительной техники и автоматизированных систем	
Томск	Томский политехнический университет	230101 Вычислительные машины, комплексы, системы и сети	www.tpu.ru
		230201 Информационные системы и технологии	
Иркутск	Иркутский государственный технический университет	230101 Вычислительные машины, комплексы, системы и сети	www.istu.edu
		230201 Информационные системы и технологии	
Хабаровск	Тихоокеанский государственный университет	230101 Вычислительные машины, комплексы, системы и сети	www.khstu.ru
		230201 Информационные системы и технологии	
Архангельск	Архангельский государственный технический университет	230201 Информационные системы и технологии	www.agtu.ru
Волгоград	Волгоградский государственный технический университет	230101 Вычислительные машины, комплексы, системы и сети	www.vstu.ru
		230105 Программное обеспечение вычислительной техники и автоматизированных систем	
Якутск	Якутский государственный инженерно-технический институт	230105 Программное обеспечение вычислительной техники и автоматизированных систем	www.yseti.ru

Курс молодого сисадмина:

- Компьютерная аппаратура (архитектура компьютера, стандарты, совместимость, периферия).
- Сетевое оборудование (кабельные системы, телекоммуникационная аппаратура и программное обеспечение).
- Операционные системы (принципы работы ОС, файловые системы).
- Протоколы обмена данными (IP-протокол, прикладные протоколы).
- Документирование процессов и процедур.
- Автоматизация процессов (скриптование, планировщик задач).
- Основы управления базами данных (устройство, безопасность, резервное копирование).
- Маршрутизация (сегментирование, VPN, фильтрация трафика).
- Системная безопасность (аутентификация, шифрование, защита передачи данных, программный и аппаратный инструментарий).
- Time Management для системных администраторов.
- Финансовые вложения в компьютерные инфраструктуры (стоимость владения, оценки рисков, управление жизненным циклом).
- Централизованное управление сетями (доменные структуры, удалённый доступ, делегирование полномочий).
- Технологические решения (терминальный сервер, корпоративный файервол, веб-службы, системы документооборота и др.).
- Внедрение технологических решений (работа с человеческим фактором, работоспособность проектов, стандартизация).

Большинство опытных специалистов делают вывод: надо тщательно выбирать вуз, внимательно просматривать список предметов, которые преподаются на данном факультете, не ограничиваться только учебой, «добирать» курсы за счет других учебных заведений и самообразования. Замечательно, если ваша alma mater проповедует системный, комплексный подход. Важно, чтобы у вас была хорошая практика.

О практике для будущих профессионалов ИТ, в том числе системных администраторов, в последнее время заботятся и научный мир, и бизнес-сообщество. Это приводит к появлению новых проектов, таких как развитие сотрудничества с крупнейшими вендорами и работодателями в области ИКТ в рамках учебно-методического объединения вузов России по университетскому политехническому образованию на базе лаборатории проблем технического образования России МГТУ им. Н.Э. Баумана. Цели сотрудничества – внедрение учебно-методической базы авторизованного обучения в области ИКТ в систему высшего образования, взаимное признание академического, авторизованного и неформального обучения, обеспечение «всем миром» непрерывного образования. Еще можно назвать ряд успешных проектов, реализованных совместно с корпорацией Microsoft в рамках MS IT Academy.

«Хочу стать сисадмином в школе». Меня там ждут?

И средним, и средним специальным заведениям, и вузам системные администраторы нужны, и даже очень, особенно там, где создан большой парк машин. Сейчас, после массовой поставки компьютеров и другой техники во все школы страны, да еще в свете того, что техника успела износиться, это актуально, как никогда.

«На кухне должна быть одна кухарка», – таково мнение учителя информатики из Рамонского лицея Воронежской области Романа Наливкина. Роман считает, что их школе повезло, у них нормальный сисадмин, который вполне грамотно и дружелюбно взаимодействует с учителями информатики. А главное, он профессионал, у которого в хозяйстве все в порядке. В школе появляется один раз в четверть.

Кто обычно становится системным администратором в школе? Наиболее частые кандидаты, по мнению учителя информатики лицея города Кемерово Даниила Титорова, это:

Учителя информатики. Не самый лучший вариант. У учителя, как правило, нет знаний, как управлять компьютерными сетями и ремонтировать компьютеры. Учиться, черпать информацию из Интернета ему зачастую лень, а экспериментов учитель боится.

«Знакомый студент». Очень часто бывший выпускник школы. Молодой человек, как правило, что-то знает о серверах, комплектующих, но в силу возраста ему не хватает инициативы, понимания нужд образовательного учреждения. Кто-то должен ему сказать, что надо делать. Очень большим плюсом молодого человека будет нетребовательность к зарплате.

Специалист. Им может стать родитель или коллеги родителя. К услугам аутосорсинга практически не прибегают. Специалист все знает, особенно хорошо ему известно, сколько реально стоят его услуги (поддержка одного компьютера организации составляет около пятисот рублей в месяц).

Ученики. В последнее время стала популярна идея, что простейшие действия могут выполнять продвинутые ученики, из них формируют школьные бригады. Ребятам льстит, что они могут решать некоторые проблемы, что им известен пароль администратора. Очевидно, что ученикам требуется наставник.

«Я был свидетелем всех четырех вариантов, – пишет Даниил Титоров. – Несколько лет был сисадмином в своих школах. Но когда численность компьютеров перевалила за пятьдесят, понял, что физически не успеваю за всем смотреть, особенно за ту зарплату, которую школа может платить (около трех тысяч рублей в месяц). Сейчас роль сисадмина в нашем лицее выполняет студент, с которым я почти не встречаюсь, он приходит по вечерам. На мою территорию, мой класс он не претендует, там я все делаю сам. Очень часто коллеги обращаются ко мне с вопросами, и я решаю их «по дружбе», часто мне помогают мои ученики».

Что должен уметь системный администратор, пришедший в школу? Даниил Титоров перечисляет:

- устанавливать и настраивать программное обеспечение, в том числе переустанавливать ОС Windows;
- находить неисправность и заменять неисправности компьютера;
- прокладывать локальные сети;
- настраивать школьный сервер.

Школьному сисадмину так и придется работать с разнообразной мультимедийной техникой, которая есть в классах, – смарт-досками, проекторами и так далее. Наконец, от него потребуется знание не только Windows, но и Linux, так как с января 2011 года многие регионы переходят на свободное программное обеспечение. BOF

Звезды «СА»

Первые – среди равных

Наши герои – авторы и читатели «СА» – начинали когда-то свою карьеру с должности системного администратора. Но стремление знать больше, видеть дальше, покорять новые высоты заставило их двигаться вперед. Все они – интереснейшие люди, достойные отдельного рассказа. Запомните их! И дополните наш список своими именами.

ВЯЧЕСЛАВ КАЛОШИН, в 1995-2000 гг. сетевой администратор в Иркутском государственном университете. Ныне – технический директор PingWin Software (компания входит в группу компаний «АйТи»)

ВЛАДИСЛАВ ЛИСТРОВОЙ, в 2007 г. – помощник системного администратора в одной из коммерческих компаний Санкт-Петербурга. Ныне – начальник ИТ компании KLG Holding

АНДРЕЙ ПОГОДИН, был сисадмином в 70-е гг. XX века. Ныне – председатель Совета директоров компании «Эн Ди Групп». Лауреат премии «IT-ЛИДЕР» 2003 и 2006 гг.

ВЛАДИСЛАВ КОТУСОВ, в 2004 г. системный администратор ИТ-департамента компании Softline. Ныне – технический директор этой компании

РОМАН МАРКОВ, в 2001-2003 гг. – помощник, затем системный администратор ведущего «1С:Франчайзи» в Санкт-Петербурге. Ныне – генеральный директор ООО «Оптимум»

АЛЕКСАНДР САМОДУРОВ, в 2002-2004 гг. системный администратор в ГУП «Мосстройресурс». Ныне – генеральный директор компании «Мастертрэйд», входящей в холдинг «Прайм Бизнес Групп» (Москва)

Где учиться за рубежом?

Университеты. Образование в сфере Computer Science можно получить в огромном количестве университетов. В США и Европе ИТ-специальности не считаются особо престижными, конкурс в большинство технических вузов невысокий.

Конкретно на специальность «системный администратор» почти нигде не учат, как и у нас. Есть такие учебные программы, как «Программист», «Специалист по безопасности», «Специалист в области информационных технологий». Сегодня в США лишь в трех вузах можно получить степень бакалавра именно как «Системный администратор». Это Рочестерский институт технологий, Тафтский и Мичиганский технические университеты. Также на системного администратора учат в Нью-Йоркском техническом колледже. Ни одно из этих заведений нельзя назвать ведущим в своей области.

Если говорить собственно об образовании в сфере Computer Science, то для него характерны гибкий учебный план и небольшое количество обязательных дисциплин. Благодаря этому будущий сисадмин может составить свою программу обучения таким образом, чтобы не тратить время на дисциплины для других специалистов.

Несмотря на серьезный приток эмигрантов с техническим образованием и большое количество своих специалистов, и в Европе, и США, по-прежнему очень востребованы ИТ-эксперты. Поэтому компании тесно сотрудничают с вузами, и выпускник университета, как правило, точно знает место своей работы после защиты диплома.

С другой стороны, многие российские специалисты, работавшие за рубежом, достаточно критично отзываются о своих зарубежных коллегах. Узкая специализация плоха тем, что многие выпускники «научились нажимать красную кнопку, и работают с людьми, которые нажимают красную кнопку, а больше ничего не умеют». Так жестко охарактеризовал админов в США один из наших собеседников, несколько лет проработавший в Штатах. Происходит это потому, что уровень образования в топовых технических вузах (Беркли, Массачусетс) многократно выше уровня обычных университетов, но выпускники Беркли чаще работают инженерами в крупных компаниях, чем сисадминами.

В итоге даже у не очень больших фирм ИТ-отдел разрастается до нескольких десятков человек. Один сотрудник отвечает за безопасность, другой за сети, третий за ПО. Российский образ «админа, который придет и молча поправит все» в США не актуален.

Сертификаты. Для получения хорошей работы сертификат от крупной компании гораздо важнее, чем диплом вуза. Наиболее престижными являются сертификаты от Microsoft, Cisco, Oracle, свои сертификаты выдает Red Hat Linux, Sun Microsystems и многие другие ИТ-компании. Такой документ подтверждает наличие определенного уровня знаний у его обладателя. Получить его можно в авторизованном учебном центре, сдав экзамен. Экзамен сдают либо на основе самостоятельно полученных знаний, либо после прослушивания дорогого курса, цены в России не ниже цен в США.

Сертификаты стали такими ценными потому, что консервативная академическая система просто не успевает за развитием технологий. А сертификаты Microsoft и Cisco подтверждают актуальность знаний и умений.

Тренинги. Можно сколько угодно говорить про узкую специализацию и недостаток знаний, но образование на Западе – это главное, особенно в ИТ. Сотрудников постоянно отправляют на семинары, приглашаются специалисты для проведения тренингов. У крупных компаний есть собственные учебные центры, иногда они могут сойти за полноценный университет. Иметь в бюджете большую статью расходов на образование сотрудников считается хорошим тоном. Программы повышения квалификации достаточно стандартны – это лекции о разных аспектах сисадминской деятельности, базы данных, сетевая инфраструктура, безопасность, техническое обслуживание компьютеров.

Подготовил Илья Александров

Как вы стали системным администратором?

На вопрос «СА» отвечают ИТ-специалисты

Алексей Барабанов, г. Москва

Когда сгенерил систему

Если с точки зрения админской деятельности, то в первый раз, будучи студентом, когда сгенерил систему на диске для М4030. Так как в обычном порядке народ просто подкладывал колоды в лоток, или иногда что-то клевал на консольной клавиатуре, то разметка диска и перенос системы – это были явно операции сисадмина. Если с точки зрения заработка, то в первый раз, когда стал начальником ЭВМ СМ-4, будучи сотрудником научной лаборатории вуза. Если с точки зрения бизнеса, то в первый раз, когда поставил сетку на NetWare286 на торговом предприятии. К этому моменту я уже переработал в полудюжине предприятий, чужих и своих. Но как выделенный админ работал первый раз по приглашению однокашника – главного бухгалтера этого предприятия.

Алексей Тараненко, г. Таганрог

Хотел заработать и стать независимым

Администрировать начал курса со второго университета. Отчасти было желание заработать деньги и меньше зависеть от родителей, отчасти хотелось применить знания и навыки, которые давал университет. Начинал с администрирования небольшой компании в 10 компьютеров, постепенно росла комания, вместе с ней рос и я. В самом начале задавал очень много «ламерских» вопросов, в их решении очень помогли форумы sysfaq.ru и форум технет, журналы «Системный администратор» и IT Спец, MCP-клубы. Постепенно сам начинал не только задавать вопросы, но и отвечать другим пользователям. В итоге получился неплохой (я надеюсь, что неплохой!) системный администратор крупной российской компании, который отвечает уже не только на форумах, но и совместно с друзьями основал собственный проект ITband.ru для помощи новичкам.

Игорь Штомпель, г. Майкоп

Я понял, что могу помогать другим

Меня всегда интересовало, как устроена та или иная система. А однажды произошло знакомство с Linux, который многое перевернул в моем отношении к операционным системам. Затем я окончил СГА (www.muh.ru) по направлению – «Информатика и ВТ». На определенном этапе пришло понимание того, что я в состоянии оказывать помощь другим. Кроме того, расширению профессионального кругозора и приобретению необходимого опыта значительно способствовали порталы – www.linuxcenter.ru, www.opennet.ru, www.intuit.ru; журналы – «Системный администратор», Linux Format, Open Source.

Кирилл Хорошилов, г. Новосибирск

Знания брал из Интернета

После вуза ничего нормального из работы я не нашел. Немного поразмыслив, принял предложение областной библиотеки (НГОНБ), инженер-программист, ЗП – 2000 руб., решив набирать необходимый всем конторам опыт. Работу рассматривал как бесплатное обучение, за которое тебе еще платят деньги. 3 года упорно изучал Linux и Windows в библиотеке, потом удалось устроиться в Центр финансовых технологий. Знания брал из Интернета, google, citforum.ru, forum.ru-board.com (раздел Администрирование), www.networkdoc.ru. В процессе уже сформировалась большая тяга в UNIX-направлении.

Алексей Ставицкий, г. Новосибирск

В нашей библиотеке

Учась на факультете радиоэлектроники и физики Новосибирского государственного технического университета, увлекся автомобильной электроникой и системами электронного управления двигателем. Устроился работать на одно из новосибирских СТО, но мое желание развивать это направление не совпадало с желанием владельца сервиса. Выбрал системное администрирование, так как ситуация с разработкой электроники в 2005 году не радовала. Опыта работы сисадмином не было, поэтому пошел работать туда куда взяли, а взяли меня в Новосибирскую государственную областную научную библиотеку эникейщиком. Спустя год, попробовал попасть в ОАО «МТС». Взяли на должность эникейщика. Полтора года беготни по пользователям и «курение док» принесли плоды. Сейчас администрирую контактный центр ОАО «МТС» в «Макро-регионе» Сибирь».

Алексей Потапов, г. Москва

Искал свою дорожку в ИТ-индустрии

Путь к системному администрированию начался ещё в 14 лет, после того как я пытался увеличить производительность своего первого компьютера путём удаления лишнего ПО, чтобы на нём можно было играть в игрушки. К сожалению, первый опыт не увенчался успехом – игрушки так и не стали работать, но зато компьютер стал работать шустрее. К 16 годам я уже с лёгкостью разбирал компьютеры и менял комплектующие. В 18 лет я уже мог чинить сложные конфигурации дорогих компьютеров. Стремление увеличить объём знаний и найти свою дорожку в ИТ-индустрии привели меня к обслуживанию клиентов с серверами.Тут и начался мой путь системного администратора.

Андрей Захарченко, г. Иркутск

Меня научил выпускник университета

Компьютер у меня появился в девятом классе. Честно проиграв год, мне все надоело, и я стал разбираться в устройстве ОС. Ставил различные версии Windows/Linux. В 2001 году поступил в университет на кафедру «Экономической кибернетики», специальность «Математические методы в экономике». Проучившись год и проявив немного свои знания на факультете, у меня появилась большое желание обзавестись аськой в учебном классе. А такую возможность имели только инженеры вычислительной лаборатории и операторы классов. На втором курсе освободилась вакансия инженера в вычислительной лаборатории кафедры, и меня взяли на это место. В мое хозяйство поступило два сервера. Один веб-сервер, второй – учебный для студентов. Под крыло меня взял один из выпускников кафедры и для меня он и по сей день является Linux-гуру. Он меня научил самому важному делу – поиску информации. Дальше по мере получения опыта и знаний в *nix-системах и желая иметь скоростной Интернет, я начал прощупывать безопасность прокси-серверов главного информационного отдела вуза. Мне удалось найти ряд уязвимостей и через них получить желаемый быстрый Интернет. По мере общения с главным администратором вуза, я ему указал на данные уязвимости, и показал, что я не один такой умный. Он, оценив мои знания, предложил перейти работать в его отдел. Так я проработал еще полтора года, занимаясь только UNIX-серверами. Позже я обратил свой взор на другие ОС, такие как Windows. Когда я изъявил желание заняться и ими, начальник свесил на меня самые сложные и долго не реализуемые проекты. Как показала практика, лучше всего учиться, именно решая задачи. Задач было много, я постепенно их решал и сейчас знаю больше своего начальника, хотя стараюсь это скрывать. По мере решения задач знакомился с администраторами на форумах. После того как я написал ряд своих статей по настройке программ для FreeBSD, ко мне стали обращаться другие администраторы и тут уже я стал делиться опытом.

Визитка

ЮЛИЯ ШТОКАЛО, юрист, специализируется в области гражданского права, права информационных технологий, налогового права. Сфера интересов: юридическое сопровождение

Сложно, но можно

Как защитить персональные данные на предприятии?

С 1 января 2010 года все информационные системы персональных данных должны быть приведены в соответствие с требованиями Федерального закона от 27 июля 2006 г. № 152-ФЗ «О персональных данных»

> Было бы очень интересно увидеть в «СА» статью, касающуюся подразделений ИТ, в связи с вступлением в силу ФЗ №152 «О персональных данных».
>
> **Алексей, г. Санкт-Петербург**

Плюсы и минусы

В Законе «О персональных данных» есть ряд плюсов и минусов. К числу плюсов можно отнести то, что на законодательном уровне определено понятие «персональные данные». Установлены их защита, а также порядок обработки персональных данных. И наконец, закон должен сыграть положительную роль в борьбе с утечками информации в организациях и на предприятиях.

К числу минусов закона относится, во-первых, отсутствие какой бы то ни было практики по данному вопросу. Поэтому не исключается различное трактование положений закона, как операторами, так и государственным органом, уполномоченным осуществлять контрольные функции, а следовательно, неизбежно наличие споров. Во-вторых, необходимо наличие лицензии на техническую защиту конфиденциальной информации во всех случаях, включая те, когда обработка персональных данных осуществляется для собственных нужд (например, в рамках трудовых отношений). Это фактически распространяет действие закона на любое лицо, собирающее сведения не для личных, семейных нужд. В-третьих, понадобится модернизация информационно-технической инфраструктуры компании, а также внедрение новых продуктов информационно-технической безопасности, что потребует дополнительных затрат трудовых и финансовых ресурсов.

Кто есть who?

Согласно закону под персональными данными понимается любая информация, относящаяся к определенному или определяемому на основании такой информации физическому лицу (субъекту персональных данных), в том числе его фамилия, имя, отчество, год, месяц, дата и место рождения, адрес, семейное, социальное, имущественное положение, образование, профессия, доходы, другая информация.

В соответствии с положениями закона оператор – это государственный орган, муниципальный орган, юридическое и физическое лицо, организующие и (или) осуществляющие обработку персональных данных, а также определяющие цели и содержание обработки персональных данных, т.е. к числу операторов фактически можно отнести любое юридическое лицо и физическое лицо, обрабатывающее персональные данные.

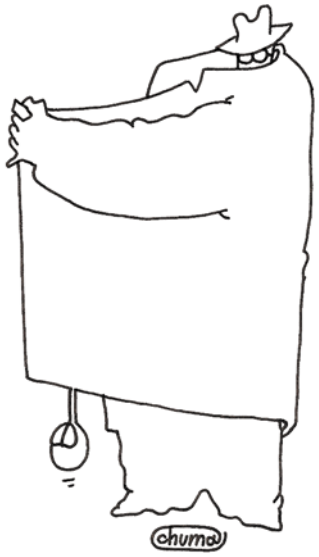
Действие закона не распространяется на отношения, возникающие при:

- обработке персональных данных физическими лицами для личных и семейных нужд;
- при обработке персональных данных в соответствии с законодательством об архивном деле в Российской Федерации;
- при обработке подлежащих включению в единый государственный реестр индивидуальных предпринимателей сведений о физическом лице в связи с деятельностью физического лица в качестве индивидуального предпринимателя;
- при обработке персональных данных, отнесенных к сведениям, составляющим государственную тайну.

Алгоритм действий организации

Теперь относительно приведения базы данных предприятия в соответствии с положениями закона. Организация должна:

Разработать и внедрить новые локальные акты, с которыми работники должны быть ознакомлены под роспись. (Например, Положение о персональных данных и их защите; Инструкция о порядке обеспечения конфиденциальности при обращении с информацией, содержащей персональные данные; Регламент допуска сотрудников к обработке персональных данных; Перечень допущенных сотрудников к обработке персональных данных.) В локальных актах необходимо определить



С введением Закона «О персональных данных» число пиратских баз должно сократиться

перечень, цели и порядок обработки персональных данных, назначить ответственных за работу с персональными данными, подготовить должностные инструкции сотрудников, обрабатывающих персональные данные.

Получить от каждого работника письменное согласие на обработку его персональных данных. Данное согласие должно включать в себя обязательные поля, перечень которых указан в пункте 4 статьи 9 Закона «О персональных данных».

Отнести предприятие к одному из четырех классов информационных систем, которые устанавливаются в соответствии с Приказом ФСТЭК России, ФСБ России и Мининформсвязи России от 13 февраля 2008 года № 55/86/20 «Об утверждении Порядка проведения классификации информационных систем персональных данных». Цель классификации – установить методы и способы защиты информации. Результаты классификации информационных систем оформляются соответствующим актом оператора.

Обеспечить техническую защиту персональных данных. Она должна осуществляться в соответствии с методическими документами ФСТЭК России (документы ДСП), которые можно получить в данной организации. Это «Основные мероприятия по организации и техническому обеспечению безопасности персональных данных, обрабатываемых в информационных системах персональных данных» от 15 февраля 2008 года. «Базовая модель угроз безопасности персональных данных при их обработке в информационных системах персональных данных» от 15 февраля 2008 года. «Методика определения актуальных угроз безопасности персональных данных при их обработке в информационных системах персональных данных» от 15 февраля 2008 года. «Рекомендации по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных» от 15 февраля 2008 года.

Провести аттестацию информационной системы персональных данных (сертификации системы защиты персональных данных). Проводить аттестацию информационной системы персональных данных вправе только организация, которая обладает лицензией ФСТЭК России на деятельность по технической защите конфиденциальной информации (в лицензии должно быть указано, что организация имеет право оказывать данные услуги). Для защиты персональных данных можно использовать только сертифицированные средства защиты персональных данных. На сайте ФСТЭК России (http://www.fstec.ru/_doc/reestr_sszi/_reestr_sszi.xls) можно ознакомиться с Реестром сертифицированных средств защиты информации. На сайте ФСБ России (http://www.fsb.ru/fsb/supplement/contact/lsz/perechen.htm) можно ознакомиться с Перечнем средств защиты информации, не содержащей сведений, составляющих государственную тайну.

Операторы, которые осуществляют обработку персональных данных, обязаны направить в уполномоченный орган по защите прав субъектов персональных данных. Уведомление об обработке персональных данных (за исключением случаев, перечисленных в п.2 ст. 22 Закона) – http://38.rsoc.ru/.cmsc/upload/documents/uvedomlenie_01.06.2009.doc.

В качестве альтернативного варианта вместо проведения процедуры получения аттестации и сертификации можно заключить договор аутсорсинговых услуг с компанией, обладающей необходимыми лицензиями и сертификатами.

За нарушение норм, регулирующих получение, обработку и защиту персональных данных, предусмотрена гражданско-правовая, административная или уголовная ответственность. Работодатель вправе привлечь к дисциплинарной ответственности сотрудника, который обязан отвечать за хранение персональной информации, однако способствовал ее распространению.

Несмотря на сложности в применении закона на практике, необходимость его принятия давно назрела. Везде можно встретить копии баз данных с адресами, телефонами частных лиц, украденных у владельцев соответствующих баз данных. С введением в силу Закона «О персональных данных» количество такого рода баз должно сократиться, а в идеале они должны исчезнуть. EOF

Визитка

АЛЕКСАНДР КОСИВЧЕНКО, технический специалист ЗАО «Компьювэй». Занимается внедрением серверных решений на базе ПО Microsoft

Комплексное решение:
виртуализация + отказоустойчивый кластер

Как можно повысить надежность информационной системы с помощью виртуализации. Какие возможности предоставляет для этого Windows Server 2008 R2?

Что же такое виртуализация?

Wikipedia дает такое определение: «Виртуализация в вычислениях – процесс представления набора вычислительных ресурсов или их логического объединения, который дает какие-либо преимущества перед оригинальной конфигурацией. Это новый виртуальный взгляд на ресурсы, не ограниченные реализацией, географическим положением или физической конфигурацией составных частей». Возможно, звучит слишком сложно для неподготовленного человека, поэтому попытаемся перевести на «человеческий язык».

Разработкой решений, основанных на виртуализации, занимаются многие вендоры. Ведь «логические диски», представляющие собой просто разделы на одном физическом жестком диске, – это тоже виртуализация. Технология SMP, позволяющая представлять для программ два или более физических процессора как один виртуальный – это тоже виртуализация. Мы же посмотрим, что под словом «виртуализация» понимает Microsoft. В настоящее время Microsoft различает три аспекта виртуализации:

- виртуализация серверов;
- виртуализация представлений;
- виртуализация приложений.

С виртуализацией представлений более-менее знакомы практически все системные администраторы: самый яркий ее пример – терминальные службы Microsoft Windows Server. Виртуализация приложений – создание особой, изолированной среды внутри ОС для запуска отдельных приложений.

Здесь же и далее под словом «виртуализация» мы будем иметь в виду виртуализацию серверов. В ОС Windows Server 2008 появилась встроенная система поддержки виртуализации (гипервизор) под названием Hyper-V. В Windows Server 2008 R2 гипервизор был существенно доработан и получил название Hyper-V 2.0.

Давайте рассмотрим подробнее виртуализацию серверов. Что же это такое? Говоря понятным языком, это создание программно-эмулируемой среды, полностью имитирующей аппаратное обеспечение физического компьютера: процессор, оперативную память, жесткий диск, устройства ввода-вывода.

На такой виртуальный сервер может быть установлена ОС (ее называют гостевой ОС, Guest OS) и некие приложения. Работать все это будет как на полноценном сервере, только он невидим: он существует виртуально, внутри ОС на физическом сервере (применительно к этой ОС используется термин хостовая ОС, Host OS). При этом внутри одного физического сервера могут одновременно работать два и более, а иногда даже десятки таких виртуальных серверов.

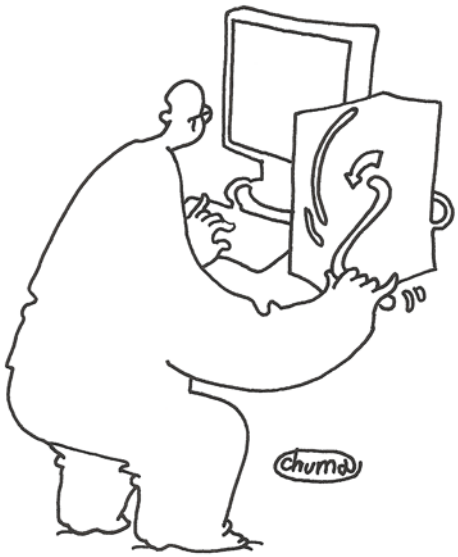
Для чего это может пригодиться? Вначале виртуальные машины использовались лишь в тестовых целях: проводить эксперименты с ними намного проще, быстрее и, главное, дешевле, чем с настоящими серверами. Наверняка многим сисадминам доводилось в своей практике испытывать что-либо на виртуальной машине. Ныне же виртуализация стала все больше и больше использоваться в промышленном применении. На то есть существенные причины, хотя, как и во всяком решении, имеют место быть и недостатки. Об этом – далее.

Достоинства и недостатки виртуализации

Самое главное: использование виртуализации позволяет более рационально распределять аппаратные ресурсы серверов. Действительно, ведь большинство серверов использует от силы 10% от своих ресурсов – процессорных мощностей, объемов памяти и т.д. Виртуализация позволяет вместо нескольких практически незагруженных серверов использовать один сервер, который будет загружен чуть сильнее. Понятно, что один сервер, пусть даже чуть более мощный, будет стоить дешевле, чем несколько отдельных.

Также вполне логично предположить, что один сервер будет потреблять намного меньше электроэнергии и занимать меньше места в стойке.

Еще одно очень важное преимущество – удобство администрирования. Любой администратор сталкивался с необходимостью идти в серверную и производить какие-то манипуляции непосредственно на консоли самого сервера в случае сбоя системы. Использование виртуализации позволяет получать доступ к консолям виртуальных серверов непосредственно с рабочего места администратора, и необходимость в экскурсиях в серверную практически отпадает.



Виртуализация позволяет вместо нескольких незагруженных серверов использовать один

Кроме этого, сильно упрощаются операции резервного копирования и аварийного восстановления серверов. Вы знаете, как бывает сложно сделать рабочую резервную копию системного раздела сервера: для этого часто приходится покупать дополнительный софт (такой, как Acronis Truelmage Server) и в некоторых случаях – перезагружать сервер. Использование виртуализации позволяет создавать резервные копии дисков серверов «на лету», незаметно для пользователей, а восстановление сводится всего лишь к копированию нескольких файлов.

Но, к сожалению, палка всегда о двух концах, и помимо всех достоинств, у решений на базе виртуализации есть существенный недостаток: понижение общей надежности системы. Действительно, так как на одном физическом сервере одновременно запущены несколько виртуальных машин, то выход из строя сервера (например, «сгоревший» процессор или RAID-контроллер) приведет к одновременному отказу всех виртуальных машин, на нем запущенных, и, соответственно, всех сервисов, которые они предоставляли.

Поэтому вместе с решениями на базе виртуализации целесообразно использовать отказоустойчивые решения, в частности – на базе отказоустойчивых кластеров. Подробнее речь об этом пойдет дальше.

Есть и еще один недостаток, касающийся лишь виртуализации на базе Windows Server 2008: аппаратные требования включают в себя 64-битный процессор с аппаратной поддержкой виртуализации и DEP. Так что множество старых серверов с 32-битными процессорами нам попросту не подходят. Тем не менее купить сервер, не удовлетворяющий техническим требованиям Hyper-V, в настоящее время затруднительно, поскольку серверы со старыми моделями процессоров были с недавнего времени сняты с производства всеми крупными вендорами.

Требования для использования виртуализации в Windows Server 2008

Как уже говорилось, основным и обязательным требованием для использования технологии виртуализации Windows Server 2008 Hyper-V является процессор, который обязательно должен иметь 64-битную архитектуру и обязательно аппаратную поддержку DEP и виртуализации (Intel VT или AMD-V).

Остальные требования зависят от задач, которые планируется выполнять. Мощность процессоров, объем оперативной памяти и дискового пространства необходимо подбирать, исходя из необходимых мощностей для запуска нужных виртуальных машин со всеми приложениями.

Отказоустойчивость

В газетах писали о случае, когда в США разбился истребитель-невидимка F-117A, стоимостью в 250 млн долларов. Расследование показало, что причиной катастрофы стал выход из строя одной микросхемы, стоимостью от силы в 20 долларов. Этот случай наглядно иллюстрирует необходимость повышения отказоустойчивости всех компонентов системы.

Пути здесь два:

- Первый – повышение надежности самих компонентов (например, жесткие диски, используемые в серверах, имеют намного больший срок наработки на отказ, чем те, что используются в домашних компьютерах).
- Второй путь – избыточное резервирование: все или особо критичные компоненты дублируются, например – жесткие диски работают в «зеркальном режиме» (RAID1), и при выходе из строя одного жесткого диска сервер продолжает работать на втором диске, и замечает это только системный администратор, но не пользователи системы.

Надо отметить, что эти два пути являются никак не взаимоисключающими, а наоборот – взаимодополняющими. Понятно, что любое повышение отказоустойчивости автоматически приводит к удорожанию всей системы, иногда – в разы, и поэтому главное здесь – найти «золотую середину».

В первую очередь, необходимо оценить, к какому ущербу в денежном эквиваленте может привести отказ системы, и повышать отказоустойчивость соразмерно этой сумме. К примеру, выход из строя жесткого диска на моем домашнем компьютере, где хранятся всего лишь какие-нибудь

фотографии и куча разного «информационного хлама», не приведет к большой катастрофе, максимум – я заплачу пару сотен долларов за новый жесткий диск. Мне будет достаточно периодически сохранять важную информацию, например, на другой жесткий диск или DVD-RW. А вот в самолет стоимостью 250 млн долларов совсем не помешало бы поставить не одну, а две микросхемы стоимостью 20 долларов каждая.

Отказоустойчивые кластеры

Помимо отдельных компонентов серверов – жестких дисков, модулей памяти и т.д. – резервироваться могут и целые серверы. В этом случае два или несколько серверов работают в группе и пользователю представляются как один сервер, обрабатывающий некие пользовательские приложения и отвечающие на запросы. Общая информация о конфигурации кластера хранится на некоем общем дисковом ресурсе, который именуется кворумом (Quorum). Для работы кластера необходим постоянный доступ к этому ресурсу всех узлов кластера. В качестве кворумного ресурса может использоваться система хранения данных с интерфейсами iSCSI, SAS или FibreChannel.

В случае выхода из строя одного из серверов (они называются «узлы кластера») пользовательские приложения автоматически перезапускаются на работоспособных узлах, и приложение либо не прекращает работу, либо прекращает на достаточно короткое время, чтобы простой не повлек за собой больших убытков. Процесс перехода приложения со сбойного узла на работоспособный называется Failover.

Для того чтобы вовремя определить сбойные узлы, все узлы кластера периодически обмениваются между собой информацией под названием heartbeat. Если один из узлов не отсылает heartbeat – это означает, что произошел сбой, и запускается процесс Failover.

Процесс Failover на примере кластера из двух узлов показан на рис. 1.

В некоторых случаях, в зависимости от настройки, при восстановлении работоспособности сбойного узла приложения могут перемещаться обратно на него – этот процесс называется Failback (см. рис. 2).

Требования для создания отказоустойчивого кластера в Windows Server 2008

Итак, что же нам нужно для создания кластера в Windows Server 2008?

Во-первых, нам нужен некий разделяемый дисковый ресурс, который будет использоваться в качестве кворума, а также для хранения данных. Это может быть любая система хранения данных (СХД), поддерживающая протоколы iSCSI, SAS или FibreChannel.

Разумеется, все узлы кластера должны иметь соответствующие адаптеры для подключения СХД. Все серверы, функционирующие в качестве узлов кластера, должны иметь если не полностью одинаковое аппаратное обеспечение (это идеальный вариант), то хотя бы процессоры одного производителя.

Также для функционирования кластера желательно, чтобы все узлы кластера связывались между собой более, чем по одному сетевому интерфейсу. Он будет использоваться как дополнительный канал обмена heartbeat, а в некоторых случаях и не только для этого (к примеру, при использовании Live Migration).

Если мы собираемся использовать виртуализацию, все узлы также должны удовлетворять системным требованиям Hyper-V (в особенности – выбранные нами процессоры).

Комплексное решение: виртуализация + отказоустойчивый кластер

Итак, как уже было упомянуто, решение на базе виртуализации может быть развернуто на платформе отказоустойчивого кластера. Что же нам это даст?

Мы сможем воспользоваться всеми достоинствами виртуализации, при этом избавившись от самого главного недостатка – единой точки отказа в виде аппаратного сервера. В случае отказа одного из серверов или каких-либо плано-

Рисунок 1. Процесс Failover – перенос сервиса со сбойного узла

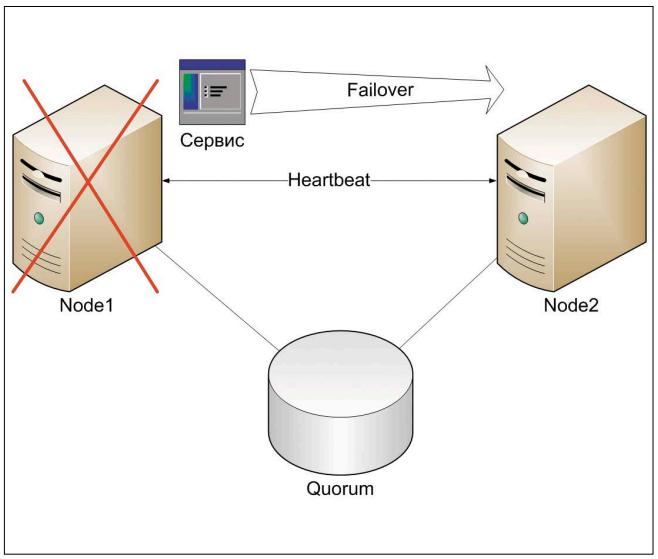
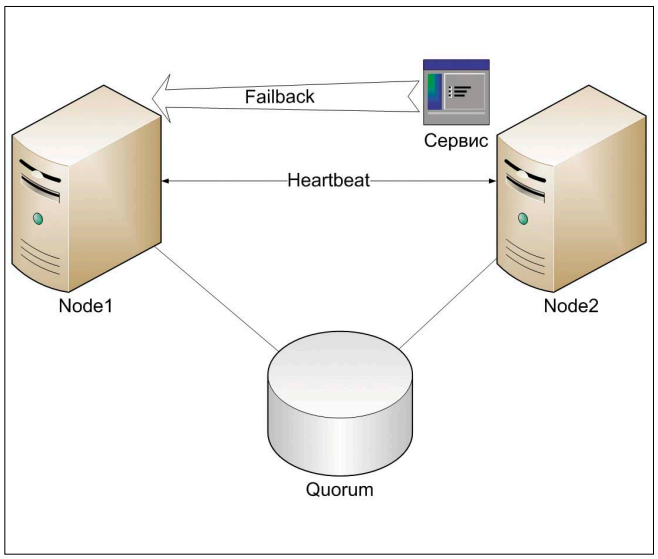


Рисунок 2. Процесс Failback – перенос сервиса после восстановления работоспособности узла



вых отключений – замены железа, установка обновлений ОС с перезагрузкой, и т.д. – виртуальные машины могут быть перемещены на работоспособный узел достаточно быстро или даже незаметно для пользователей. Таким образом, время восстановления системы после сбоя будет измеряться минутами, а плановых остановов серверов пользователи не заметят вообще. Недостаток тут один: удорожание системы.

Во-первых, скорее всего придется купить СХД, которая стоит определенных, а иногда и немалых денег.

Во-вторых – как минимум еще один сервер.

В-третьих, для работы в составе кластера понадобится более дорогая версия ОС – Enterprise или Datacenter Edition.

Это компенсируется бесплатным правом на запуск определенного количества гостевых ОС (до 4 на сервер – в Enterprise и без ограничений на сервер – в Datacenter) либо же можно использовать бесплатный продукт – Microsoft Hyper-V Server R2, в версии R2 он начал поддерживать кластеры.

Способы перемещения виртуальных машин между узлами кластера

Итак, допустим, у нас имеется кластер с запущенными виртуальными машинами. Недавно пользователи начали жаловаться, что им стало не хватать быстродействия системы. Анализ производительности показал, что приложениям не хватает оперативной памяти и иногда – мощности процессора. Было принято решение добавить в сервер несколько модулей ОЗУ и дополнительный процессор.

После того как процессор и модули памяти пришли от поставщика, встала задача, собственно, произвести замену. Как известно, для этого необходимо выключать сервер на некоторое время. Тем не менее пользователям надо работать, и простой даже 10 минут чреват убытками. К счастью, мы заранее подняли кластер, и поэтому оставаться после работы нам не нужно, а надо всего лишь переместить работающие виртуальные машины на другой сервер. Как это можно осуществить? Есть три способа:

Move – простое перемещение виртуальной машины с одного узла на другой. Виртуальная машина при этом переводится в состояние Offline (через завершение работы или сохранение состояния), а затем запускается на другом узле. Самый простой способ, но и наиболее долгий и «чувствительный» для пользователей. Перед перемещением необходимо оповестить всех пользователей, чтобы они могли сохранить свои данные и выйти из приложений.

Quick Migration – содержимое оперативной памяти сохраняется целиком на диск, а затем на целевом хосте происходит запуск виртуальной машины с восстановлением содержимого памяти с диска. Пользователей опять же нужно предупредить заранее – хотя процесс пройдет значительно быстрее.

Live Migration – одна из самых интересных новых технологий Windows Server 2008 R2. При Live Migration происходит прямое копирование содержимого памяти виртуальной машины по сети с одного хоста на другой, минуя диски. Процесс чем-то напоминает создание теневых копий открытых файлов (VSS). Вначале копируется все содержимое памяти. Затем, в случае если за время копирования произошли какие-либо изменения, копируется содержимое измененных страниц памяти. Процесс повторяется итеративно, до тех пор, пока содержимое областей памяти на обоих хостах не станет абсолютно идентичным. Как только это произошло – виртуальная машина тут же перезапускается на новом хосте. Файлы виртуальных дисков (VHD) хранятся на общем ресурсе и поэтому они просто «подцепляются» на новом хосте. Весь процесс перезапуска занимает доли секунды, меньше, чем тайм-аут TCP-соединения, и поэтому пользователи вообще ничего не замечают. Таким образом, все запланированные работы, требующие останова системы, можно проводить и в нормальное рабочее время, не отвлекая пользователей от работы.

Необходимо также отметить, что любой из перечисленных способов, и Live Migration в том числе, являются штатными возможностями Windows Server 2008 R2 и не требуют для использования покупки каких-либо дополнительных программных продуктов и лицензий. Разумеется, это не отменяет необходимости лицензирования гостевых ОС.

В случае если в качестве гостевых ОС используется Microsoft Windows, то предоставляется возможность бесплатного использования определенного количества гостевых ОС, в зависимости от версии хостовой ОС:

- Standard – одна гостевая ОС бесплатно;
- Enterprise – до четырех;
- Datacenter – без ограничений на один физический хост.

Если же используется бесплатный продукт Microsoft Hyper-V Server, то лицензирование гостевых ОС осуществляется по обычным схемам.

***

Виртуализация серверов позволяет использовать один сервер там, где раньше приходилось использовать десять. Разумеется, это позволит хорошо сэкономить практически на всем: и на «железе», и на лицензиях, и на накладных расходах. Тем не менее при использовании виртуализации значительно падает общая надежность системы.

В этой статье мы познакомились с тем, как повысить надежность такой системы за счет использования отказоустойчивых кластеров. Следующая статья будет целиком посвящена одной из «изюминок» Windows Server 2008 R2 – Live Migration. Будет рассказано о типовых сценариях использования Live Migration и дано практическое руководство по развертыванию. EOF

Визитка

СТАНИСЛАВ ШПАК, более 5 лет занимается сопровождением Active Directory и Windows-серверов. Имеет сертификаты MCSE по Windows Server 2000/2003

Терминальные службы

Установка, печать и балансировка нагрузки

Переходя на сервер терминалов под Windows Server 2008, вы получаете много новых возможностей. Я протестировал некоторые из них и готов поделиться с вами результатами

Технология терминального сервера и тонких клиентов была уже далеко не новой в год выхода Windows 2000, где она впервые появилась как часть операционной системы от Microsoft (лицензированной, кстати, у Citrix). В клиентской ОС встроенная возможность удаленного подключения на базе терминального сервиса появилась впервые в Windows XP (клиентские части для Windows 98 и Windows 2000 устанавливались отдельно). В Windows 2003 реализация терминальных служб не очень далеко ушла от предшественницы. Но в Windows Server 2008 решили это исправить. По крайней мере, в старый добрый терминальный сервис внесли столько новшеств, что на их полное рассмотрение потребуется книга, а не статья. Поэтому тут я расскажу только о том, что мне показалось наиболее важным и, главное, что уже успел проверить на практике. Те, кто работает с продуктами Citrix, могут справедливо сказать, что многие из «новшеств» уже были реализованы до Microsoft, но речь мы будем вести не о Citrix.

Терминальный сервер: апгрейд или чистая установка?

Разумеется, чистая установка всегда лучше, так как позволяет начать жизнь с нового листа. Однако иногда хочется, или даже необходимо, сохранить настройки и ПО сервера. В любом случае надо иметь в виду, что для терминального сервера с Windows Server 2008 потребуется сервер лицензий также под управлением Windows Server 2008. Существует обратная совместимость: сервер лицензий Windows Server 2008 может обслуживать серверы терминалов под Windows 2008, 2003, 2003 R2 и 2000, однако обратное неверно. Как вариант решения этой проблемы можно произвести апгрейд и сервера лицензирования, но затем потребуется его повторная активация [2].

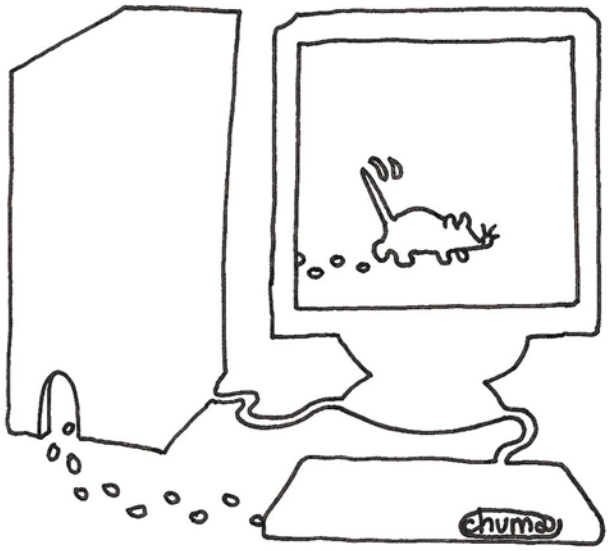
Требования к клиентскому компьютеру

Для подключения к терминальному серверу в Windows XP, 2003, Vista используется программа Remote Desktop Connection («Подключение к удаленному рабочему столу»), ярлык которой обычно располагается в «Пуск → Программы → Стандартные → Связь» (версия RDC 6.1 держит свой ярлык уже в «Пуск → Программы → Стандартные»). Можно подключаться к терминальному серверу с помощью любой версии RDC (если на самом сервере не настроены ограничения), однако чтобы воспользоваться всеми новшествами Windows Server 2008, потребуется версия RDC 6.0 или 6.1 (версии 6.0.6000 или 6.0.6001). Кроме того, если вы захотите использовать новые возможности печати, то придется установить .NET Framework 3.0 SP1 или выше. Все это можно свободно скачать с сайта Microsoft.

Установка терминального сервера

Если вы выполнили чистую установку Windows Server 2008 на новый компьютер, то по умолчанию на этом сервере не установлено ни одной роли, в том числе и терминального сервера. Чтобы сервер начал обслуживать клиентов в качестве терминального, нужно добавить соответствующую роль через оснастку Server management («Управление сервером») или вручную запустить мастер добавления роли сервера. Здесь и далее предполагается, что вы используете полную версию Windows Server 2008, а не Core Server. В качестве роли следует выбрать Terminal Services и далее выбрать необходимую службу роли (role service) из доступного списка (см. рис. 1). Для установки основного функционала терминального сервера достаточно выбрать первую опцию – Terminal Server («Сервер Терминалов»). При выборе каждого пункта справа появляется краткое описание, что особо удобно для тех, кто сначала предпочитает устанавливать что-то, а лишь потом читать инструкцию.

Далее следует предупреждение о том, что если на сервере установлены какие-то программы, то они могут не работать в многопользовательской среде, и рекомендуется в этом случае провести удаление и повторную установку таких программ. Далее нужно выбрать, хотим ли мы использовать Network Level Authentication («Проверка подлинности на уровне сети»). Выбор этой опции повышает безопасность, однако в качестве клиентов можно использовать только ОС Windows Vista или Server 2008. Вопрос на следующем экране о режиме лицензирования можно отложить на потом,



В старый терминальный сервис внесли столько новшеств, что на их рассмотрение потребуется книга

в течение 120 дней вы будете иметь полнофункциональный терминальный сервер, а за это время можно определиться и с режимом лицензирования, и с установкой сервера лицензий. На последнем шаге можно выбрать пользователей или группы пользователей, которым будет разрешен вход на сервер через удаленный рабочий стол. Группа «Администраторы» сюда уже включена. Так же, как и в Windows Server 2003, если вы планируете использовать удаленный рабочий стол только для администрирования, то совсем не обязательно устанавливать на него роль терминального сервера. Два удаленных подключения от членов группы локальных администраторов допустимы и без этого, достаточно разрешить подключение через удаленный рабочий стол в свойствах системы (там же, где и в Windows Server 2003).

Нажав Install и дождавшись установки компонентов, вам потребуется перезагрузить сервер.

После перезагрузки у вас будет полноценный терминальный сервер с базовым функционалом. В Windows Server 2008 теперь поддерживаются такие приятные мелочи, как пользовательское разрешение экрана (вплоть до размера 4096x2048), поддержка нескольких мониторов, сглаживание шрифтов, функция приоритезации данных (которая выделяет более высокий приоритет трафику от экрана, клавиатуры и мыши над трафиком печати и передачи файлов, что не дает терминальной сессии «замораживаться» при печати больших файлов) и многое другое [3]. Однако особо я бы отметил два новшества – Single Sign-On (единый вход) и печать.

Первое позволяет доменным пользователям, войдя в систему, производить вход на сервер терминалов, не вводя дополнительно учетных данных. Почему это может быть важно, я скажу чуть ниже, а вот новшества печати заслуживают того, чтобы поговорить о них отдельно.

Печать из терминальной сессии

Когда количество клиентов терминального сервера превышает несколько десятков и при этом используется парк из различных моделей принтеров, то это начинает создавать головную боль администратору. Ведь для того чтобы клиент мог печатать из своей терминальной сессии на свой локальный принтер, на сервере должен быть установлен драйвер этого принтера. Чем больше используемых моделей – тем больше драйверов. А это плохо сказывается не только на быстродействии, но и на стабильности сервера. Отчасти положение можно спасти так, как было описано в статье [4], отчасти – используя универсальный драйвер печати (для некоторых моделей принтеров HP), отчасти – осуществив подстановку драйверов на уровне сервера [5], однако эти методы кардинально не исправляют ситуацию. Похоже, в Microsoft тоже озадачились этой проблемой, и в результате в Windows Server 2008 Terminal Services появился универсальный драйвер печати Terminal Services Easy Print.

Теперь при подключении клиента к серверу терминалов система просматривает установленные на клиентском компьютере принтеры и пытается сначала установить для них в удаленном сеансе драйвер Terminal Services Easy Print. Только если это не удается, будет произведен поиск драйвера на сервере, и в случае его отсутствия принтер подключен не будет. Это поведение можно изменить через групповую политику Use Terminal Services Easy Print printer driver first («Использовать в первую очередь драйвер принтера Easy Print служб терминалов») в разделе Computer Configuration → Administrative Templates → Windows Components → Terminal Services → Terminal Server → Printer Redirection («Конфигурация компьютера → Административные шаблоны → Компоненты Windows → Службы терминалов → Сервер терминалов → Перенаправление принтеров»). Если перевести ее в состояние Disabled («Отключено»), сначала будет произведен поиск драйвера на сервере и только потом попытка установить для принтера драйвер TS Easy Print. В этом же разделе есть полезная политика Redirect only the default client printer («Перенаправлять только используемый по умолчанию принтер клиента»), в состояние Enabled («Включено») она позволяет подключать при входе на сервер только принтер по умолчанию клиента.

Что касается драйвера TS Easy Print, то, по заверениям Microsoft, он должен обеспечивать тот же набор возможностей, что и локальный драйвер, поскольку при вызове окна

настроек принтера в удаленном сеансе вызывается окно с локальными настройками (см. рис. 2). На рисунке видно, что окно свойств принтера располагается поверх окна удаленного подключения, то есть оно открывается локально у клиента.

Из неприятностей, которые могут вас ожидать с печатью из Windows Server 2008 Terminal Server, можно отметить то, что не происходит перенаправления LPT-порта при подключении к удаленному рабочему столу. Тем, кому нужна именно поддержка LPT, можно попробовать обратиться к странице [6], где рассматривается решение этой проблемы.

Попытка подключения принтера через драйвер TS Easy Print при входе на сервер включена по умолчанию, поэтому если этого не происходит, проверьте, установлен ли на клиентском компьютере .NET Framework (версия 3.0 SP1 или выше) и RDC 6.1 (именно 6.1, версия 6.0 TS Easy Print не поддерживает).

Однако, кроме новшеств основной функциональности терминального сервера, в Windows 2008 появились и некоторые дополнительные нововведения, реализованные через дополнительные службы ролей. Мы их видели при установке роли терминального сервера, к ним относятся:

TS Licensing – лицензирование служб терминалов;

TS Session Broker – посредник сеансов служб терминалов;

TS Gateway – шлюз служб терминалов;

TS Web Access – веб-доступ к службам терминалов.

Некоторые из этих служб уже так или иначе были реализованы ранее (например, лицензирование и посредник служб терминалов), остальные являются совершенно новыми. Обратимся сначала к старым знакомым службам и посмотрим, что нового предлагает нам в их реализации Server 2008.

TS Licensing

Эта служба присутствовала и в более ранних версиях Windows Server, поэтому не буду на ней останавливаться очень подробно. Из нового, о чем стоит упомянуть, появилось следующее:

- возможность отзыва выданных лицензий;
- возможность отслеживания выданных лицензий в режиме Per-User (если сервер лицензирования работает в доменном режиме, а не в режиме рабочей группы).

Ну и, конечно же, надо помнить о том, что для лицензирования сервера терминалов под управлением Windows Server 2008 нужен сервер лицензирования, также работающий под этой версией ОС.

TS Session Broker

Это логическое продолжение службы, известной в Windows Server 2003 как Terminal Services Session Directory, которая по умолчанию присутствовала в любой системе, но была отключена, так как становилась полезной только при работе кластера из нескольких терминальных серверов. Достаточно было включить эту службу и настроить каждый из терминальных серверов на работу с ней (через оснастку Terminal Server Configuration в разделе Server Settings). В результате при переподключении клиента к кластеру служба всегда перенаправляла его на тот сервер, где уже существовала запущенная от имени этого пользователя сессия, если таковая имелась. Это было удобно при разрывах связи между клиентом и сервером и исключало возможность запуска нескольких терминальных сессий на разных серверах кластера от имени одного и того же пользователя.

В реализации Windows Server 2008 пошли несколько дальше. Во-первых, служба была переименована в Terminal Server Session Broker (Посредник служб терминалов), во-вторых, она становится доступной только после установки соответствующей сервисной роли, а в-третьих, что самое важное, на мой взгляд, в функционал службы добавлена балансировка нагрузки кластера.

Итак, как это работает.

Допустим, у вас есть несколько терминальных серверов и вы хотите включить балансировку нагрузки так, чтобы сессии пользователей равномерно распределялись между серверами. Сделать это можно несколькими способами:

- Самый простой – это использовать DNS Round Robin (включено по умолчанию в каждом DNS-сервере от Microsoft). В DNS прописывается общее (кластерное) имя, с которым связываются IP-адреса всех тер-

Рисунок 1. Установка терминального сервера

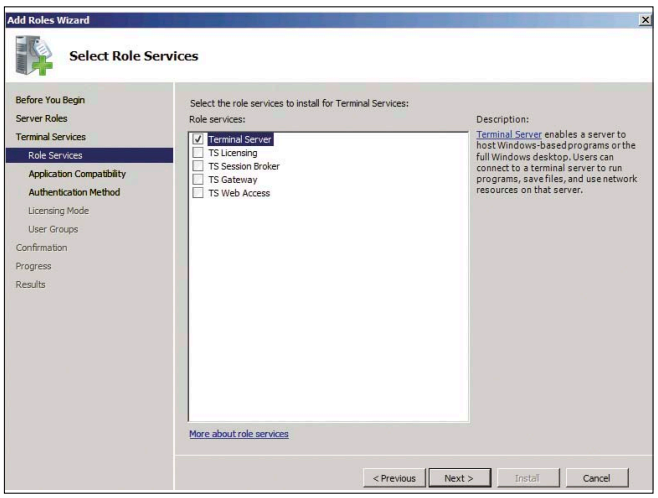
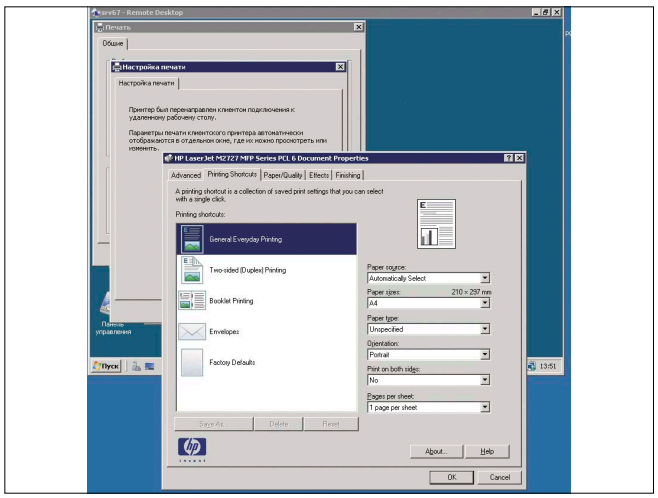


Рисунок 2. Из удаленного сеанса открывается окно настроек локального принтера



минальных серверов. После этого при обращении клиентов к удаленному рабочему столу по этому имени DNS-сервер будет последовательно выдавать IP-адреса всех терминальных серверов. Минусом этого способа является кэширование DNS-записей, что не позволит равномерно распределить сессии по терминальным серверам. Кроме того, при недоступности сервера получение следующего IP-адреса произойдет только по истечении 30-секундного тайм-аута.

- Другой способ – использовать балансировку нагрузки Windows (WNLB) или сторонних разработчиков. Это более гибкий способ, позволяющий точнее распределить сессии, однако тоже не лишен некоторых недостатков. Их рассмотрение выходит за рамки статьи, однако я готов пообщаться на эту тему на форуме журнала (www.samag.ru/forum).
- В Windows Server 2008 можно использовать TS Session Broker.

Чтобы воспользоваться третьим пунктом, нужно, чтобы на всех терминальных серверах, которые будут работать в качестве кластера, была установлена Windows Server 2008. Служба TS Session Broker не требует никакого конфигурирования, а вот на сервере потребуется задание некоторых параметров. Запустите оснастку Terminal Services Configuration («Конфигурация служб терминалов») и в разделе Edit Settings («Изменить настройки») дважды щелкните на пункт Member of farm of TS Session Broker («Член фермы в посреднике служб терминалов») (см. рис. 3).

В появившемся окне (см. рис. 4) потребуется:

- установить опцию Join a farm in TS Session Broker («Присоединиться к ферме в посреднике служб терминалов»);
- указать имя или адрес сервера с установленной службой роли TS Session Broker;
- указать общее имя фермы серверов. Это может быть имя, отличное от того общего имени, по которому клиенты будут обращаться к серверам, – оно служит только лишь для идентификации фермы на сервере-посреднике служб терминалов в том случае, если сервер обрабатывает несколько ферм. Однако во избежание путаницы рекомендуется задать имя фермы серверов такое же, как и общее имя для обращения клиентов к ферме (которое должно быть зарегистрировано в DNS вручную либо прописано в файлы host на клиентах);
- указать опцию Participate in Session Broker Load-balancing («Участвовать в балансировке нагрузки посредника»). Без этого балансировка нагрузки не включится, а будет происходить только перенаправление пользователей к их существующим сессиям, как при использовании Windows Server 2003;
- указать относительный весовой коэффициент данного сервера в ферме. Так как это относительная величина, то суммарный вес всех серверов может превышать 100. Например, на сервер с коэффициентом 50 будет приходиться в три раза меньше пользовательских сессий, чем на сервер с коэффициентом 150;
- опцию Use IP adress redirections («Использовать перенаправление IP-адресов») рекомендуется установить (по умолчанию). О ней я скажу позже;
- наконец нужно указать локальный IP-адрес для подключения (необходимо в том случае, если на сервере используется несколько сетевых карт или установлено несколько IP-адресов).

Сделав аналогичные настройки на всех серверах (серверы под управлением Windows 2003 не поддерживаются), мы получим ферму терминальных серверов с балансировкой нагрузки, которая будет работать следующим образом (далее для примера возьму имена серверов, указанных на рис. 4).

Клиент подключается к удаленному рабочему столу по общему имени FARM. Общее имя разрешается в какой-либо IP-адрес одного из серверов фермы, к которому затем и пытается подключиться клиент. Сервер, приняв запрос на подключение, запрашивает учетные данные пользователя, а затем обращается к серверу SRV67 с установленной службой TS Session Broker. Сервер SRV67 хранит у себя информацию о состоянии всех терминальных сессий серверов фермы, такую как ID сессии, связанные с ней имя пользователя и имя сервера фермы, на котором существует эта сессия. Если для указанного пользователя существует сессия на одном из серверов фермы, то пользователь перенаправляется на этот сервер и в результате подключается к своей существующей сессии. Если же такой сессии нет, то SRV67 возвращает IP-адрес сервера с наименьшим количеством клиентских сессий (с учетом такого фактора, как весовой коэффициент сервера). В результате клиент перенаправляется на этот сервер.

Если по каким-то причинам перенаправление клиента по IP-адресу невозможно, то можно сбросить опцию Use IP adress redirections. При этом перенаправление будет происходить посредством формирования специального токена (routing token). Однако Microsoft рекомендует использовать этот режим только в случае, если в качестве дополнительного решения для балансировки нагрузки используются средства, которые умеют работать с такими токенами. Замечу, что WNLB к таковым не относится.

Может возникнуть вопрос, для чего использовать дополнительные средства балансировки нагрузки, если TS Session Broker сам умеет проводить балансировку, но, несмотря на это, Microsoft рекомендует использовать его в сочетании с другими методами? Представьте себе, что в связи со сбоем электропитания или дружным началом рабочего дня несколько десятков клиентов пытаются обратиться к ферме терминальных серверов. Один сервер в состоянии обслужить только 16 одновременных подключений. Если не использовать дополнительную балансировку, то остальные клиенты получат ошибку подключения. Поэтому вы сами должны решить исходя из количества обслуживаемых клиентов, нужна ли дополнительная балансировка или нет.

Могут также возникнуть случаи, когда нужно выключить один из серверов фермы на обслуживание. При не форс-мажорных обстоятельствах для этого служит механизм drain-stop (приостановка). Перевести сервер в это состояние можно в разделе Edit Settings окна Terminal Server Configuration, дважды щелкнув пункт User Logon Mode («Режим входа пользователя в систему») (на рис. 3 этот пункт подчеркнут синим) и установив опцию Allow reconnections, but prevent new logons («Разрешить переподключения, но запретить новые попытки входа»). При этом пользователи смогут работать и подключаться к имеющимся на серве-

ре сессиям, однако новые пользователи на сервер входить уже не будут. Таким образом, через некоторое время сервер будет освобожден от пользовательских сессий.

А теперь поговорим о неприятных неожиданностях, которые обнаруживаются с использованием TS Session Broker.

TS Session Broker – подводные камни

Вот некоторые особенности, которые при внедрении TS Session Broker стоит иметь в виду и быть к ним готовым.

- Иногда, работая с фермой серверов, есть необходимость подключиться к какому-то конкретному серверу. При включении балансировки через TS Session Broker этого нельзя сделать. Для администраторов существует лазейка, а вот если вы простой пользователь – то, увы. С одной стороны, это можно рассматривать как защиту от слишком умных пользователей, которые иногда любят подключаться к терминальным серверам в обход балансировки, с другой – даже вы сами не сможете это сделать для пользователя, если возникнет такая необходимость. Будучи локальным администратором терминального сервера, вы можете подключиться удаленно к нужному вам серверу (несмотря на включенный TS Session Broker или режим запрета новых подключений), вызвав RDP-клиент с параметром /admin (mstsc.exe /admin).
- Если клиенты используют ОС ниже Windows Vista, то, возможно, им придется дважды вводить свои учетные данные при подключении к ферме терминальных серверов с включенным TS Session Broker. Вызвано это тем, что такие ОС не могут использовать функцию «единого входа» (single sign-on), о которой я писал выше. Соответственно первый раз учетные данные нужно будет ввести при подключении к серверу фермы, который затем обратится к серверу TS Session Broker и перенаправит клиента на нужный сервер, где клиенту придется повторно ввести свои учетные данные уже при подключении к конечному серверу.
- Когда я проводил тестирование новых возможностей балансировки, то обнаружил, что при выключении одного из серверов фермы информация об этом попадает на сервер TS Session Broker почему-то далеко не сразу. В течение приблизительно 5 минут после того как один из серверов фермы уже стал недоступен, TS Session Broker все равно продолжает перенаправлять клиентов на этот сервер. Соответственно работа пользователей будет прервана на это время.

***

Новая версия терминального сервера от Microsoft производит приятное впечатление. Появление универсального драйвера печати позволяет избавиться от головной боли с драйверами принтеров, а новый подход к балансировке нагрузки – от необходимости настройки и конфигурирования WNLB (которая не всегда так проста, как кажется на первый взгляд). Разумеется, есть и не совсем приятные особенности, но уже вам решать – стоит ли предпринимать переход на Windows Server 2008 или нет. BOF

1. Бирюков А. Терминальные службы в Windows Server 2008. //Системный администратор, №5, 2008 г. – С. 12-16.
2. Activate a Terminal Services license server – http://go.microsoft.com/fwlink/?linkid=101640.
3. Terminal Services Core Functionality – http://technet.microsoft.com/en-us/library/cc772366(WS.10).aspx.
4. Коробко И. Оптимизация сетевой печати. //Системный администратор, №11, 2008 г. – С. 14-20.
5. Статья KB239088 – http://support.microsoft.com/kb/239088/en-us.
6. Перенаправление LPT-порта в Server 2008 (терминальний режим) – http://social.technet.microsoft.com/Forums/ru-RU/ws2008ru/thread/fc593264-592c-4fd2-94f1-22a50031602e.

Рисунок 3. Конфигурация служб терминалов

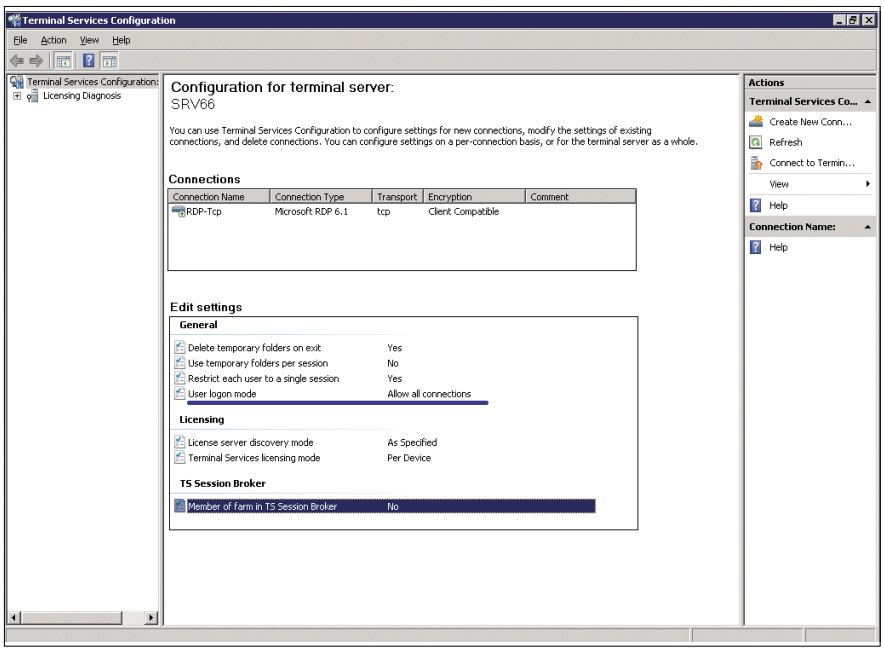
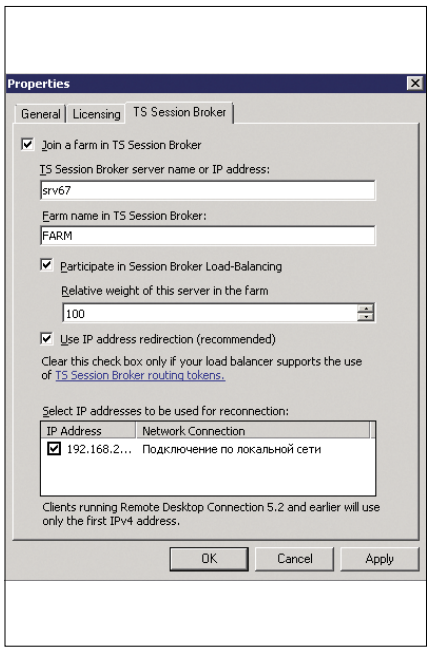


Рисунок 4. Конфигурирование терминального сервера как члена фермы серверов



Множественные уязвимости в Network Security Services

Программа: Network Security Services версии до 3.12.3.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки в обработчике регулярных выражений при сравнении общих имен в сертификатах. Удаленный пользователь может с помощью специально сформированного сертификата, подписанного CA, или сертификата, принятого пользователем, вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.

2. Уязвимость существует из-за ошибки при обработке определенных полей в сертификате. Злоумышленник может обманом заставить пользователя принять специально сформированный сертификат.
URL производителя: www.mozilla.org/projects/security/pki/nss.
Решение: Установите последнюю версию 3.12.3 или выше с сайта производителя.

Переполнение буфера в Keyview XLS в Lotus Notes

Программа: IBM Lotus Notes 5.x, IBM Lotus Notes 6.x, IBM Lotus Notes Client 6.x, IBM Lotus Notes 7.x, IBM Lotus Notes 8.0.x, IBM Lotus Notes 8.5.x.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: Целочисленное переполнение обнаружено в просмотрщике Keyview XLS (xlssr.dll). Удаленный пользователь может с помощью специально сформированного XLS-вложения, открытого приложением, вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www.lotus.com.
Решение: Установите исправление для IBM Lotus Notes версий 7.x, 8.0.x и 8.5.x. Уязвимость не устранена в версиях 5.x и 6.x.

Целочисленное переполнение в Autonomy KeyView SDK

Программа: Autonomy KeyView Export SDK 10.x, Autonomy KeyView Filter SDK 10.x, Autonomy KeyView Viewing SDK 10.x.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: Целочисленное переполнение обнаружено в Keyview XLS просмотрщике (xlssr.dll версии 8.0.0.7214, 8.5.0.8339 и 10.5.0.0) при обработке SST-записей (Shared String Table). Удаленный пользователь может с помощью специально сформированного XLS-файла вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www.autonomy.com.
Решение: В настоящее время способов устранения уязвимости не существует.

Множественные уязвимости в Microsoft Remote Desktop Connection

Программа: Microsoft Remote Desktop Connection Client for Mac 2.x, Microsoft Windows 2000, Microsoft Windows XP, Microsoft Windows 2003, Microsoft Windows Vista, Microsoft Windows 2008.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за ошибки в mstscax.dll RDP-библиотеке (Remote Desktop Connection) при обработке ответов от сервера. Злоумышленник может обманом заставить пользователя подключиться к вредоносному RDP-серверу, вызвать повреждение памяти и выполнить произвольный код на целевой системе.

2. Уязвимость существует из-за ошибки в Remote Desktop Web Connection ActiveX-компоненте. Удаленный пользователь может с помощью специально сформированного веб-сайта вызвать переполнение динамической памяти и выполнить произвольный код на целевой системе.
URL производителя: www.microsoft.com.
Решение: Установите исправление с сайта производителя.

Множественные уязвимости в Adobe Flex

Программа: Adobe Flex 3.3 SDK и более версии.
Опасность: Высокая.
Наличие эксплоита: Нет.
Описание: 1. Множественные уязвимости существуют из-за использования уязвимой версии Adobe Flash Player.

2. Уязвимость существует из-за недостаточной проверки URL в установочном шаблоне index.template.html. Удаленный пользователь может выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.
URL производителя: www.adobe.com/products/flex.
Решение: Установите последнюю версию 3.4 SDK с сайта производителя.

Переполнение буфера в ProFTP

Программа: ProFTP 2.9, возможно, другие версии.
Опасность: Средняя.
Наличие эксплоита: Да.
Описание: Уязвимость существует из-за ошибки проверки границ данных при обработке ответов от FTP-сервера. Удаленный пользователь, контролирующий FTP-сервер, может с помощью специально сформированного ответа вызвать переполнение стека и выполнить произвольный код на целевой системе.
URL производителя: www.labtam-inc.com/index.php?act=products&pid=1.
Решение: В настоящее время способов устранения уязвимости не существует.

Составил Александр Антипов

Визитка

ИВАН КОРОБКО, сертифицированный специалист MCP, автор более 50 статей и двух книг. Занимается созданием различных приложений для Active Directory

Домашний хостинг

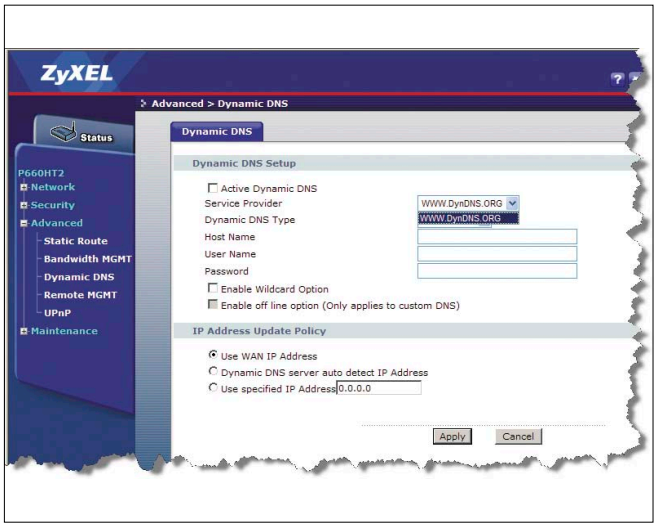
Используем сервисы динамического DNS

Как обеспечить постоянный доступ к сетевым ресурсам, у которых периодически меняется IP-адрес

В большинстве современных сетей TCP/IP используется служба DNS, главное назначение которой – сопоставлять простые для запоминания человеком имена типа company.com IP-адресам. Без этой службы использование Интернета было бы весьма затруднительным, т.к. запомнить множество IP-адресов гораздо сложнее, чем ассоциирующиеся с контентом имена сетевых ресурсов. Похожим образом работают сервисы Dynamic DNS (Dynamic DNS, DDNS), которая позволяет назначать доменные адреса ресурсам, часто меняющим IP-адрес. Для обновления изменившегося IP-адреса используется специальная программа, которая устанавливается на компьютере или включена в прошивку маршрутизатора.

Большинство xDSL-провайдеров предоставляет доступ в Интернет с помощью динамического IP-адреса, выделяемого из пула своих IP-адресов. При этом IP-адрес обычно меняется раз в 24 часа, по прошествии которых осуществляется разрыв существующей сессии с последующим ее восстановлением, при этом происходит обновление IP-адреса. В связи с этим ресурсы, находящиеся за маршрутизатором во внутренней сети (LAN), будут недоступны, т.к. теперь для обращения к ним IP-адрес неизвестен. Используя клиент службы DDNS, встроенный в большинство ADSL-модемов, добиваются сопоставления нового IP-адреса зарегистрированному доменному имени.

Рисунок 1. Служба DDNS в модеме ZYXEL



Сервисы Dynamic DNS в Интернете

В Интернете существует множество сайтов, бесплатно предоставляющих доменные имена с поддержкой DDNS. В настоящее время в мире существует несколько таких сервисов:

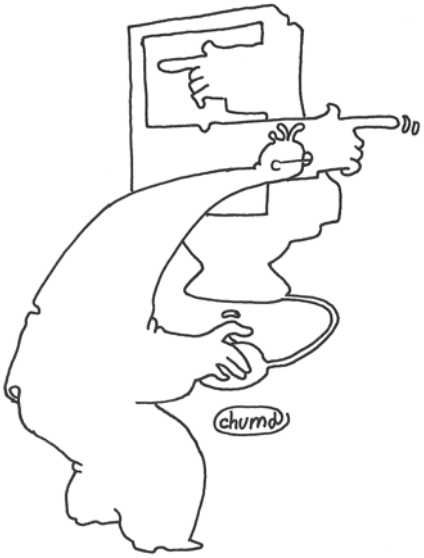
- no-ip.com
- tzo.com
- dyndns.com
- dyndns.dk
- FreeDNS.afraid.org
- ChangeIP.com

Список сервисов, из которых предстоит сделать выбор, ограничивается перечнем, находящимся на встроенном сайте ADSL-модема. Обычно он содержит один или два сайта. Этот список жестко записан в прошивку модема и может быть изменен только производителем (см. рис. 1).

Каждый из сервисов поддерживает несколько десятков доменных имен второго уровня, например dyndns.info, game-host.org, homeftp.net. Имя хоста пользователь определяет самостоятельно во время регистрации учетной записи на сервере.

Домены первого уровня

В настоящее время в Интернете серверы DDNS предоставляют домены второго уровня, при этом активно используются различные общесетевые домены верхнего уровня. Последние делятся на общесетевые и географические. Поскольку местонахождение регистрируемого домена пользователем с помощью сервера DDNS не является статическим, то сервер предоставляет только общедоступные имена доменов.



В Интернете много сайтов бесплатно предоставляют доменные имена с поддержкой DDNS

Регистрация на сервере Dynamic DNS

Перед регистрацией доменного имени необходимо создать на нем свою учетную запись. На сервере http://www.DynDNS.com процедура регистрации предельно проста. Необходимо задать несколько параметров:

Имя для входа на сайт (login), который впоследствии будет указан в ADSL-модеме (см. рис. 2).

Пароль и его подтверждение. Длина пароля должна быть не менее 5 символов.

Адрес электронной почты и его подтверждение. Адрес указанной почты не должен быть фиктивным, поскольку на него приходит письмо, в котором находится ссылка для активации созданной учетной записи.

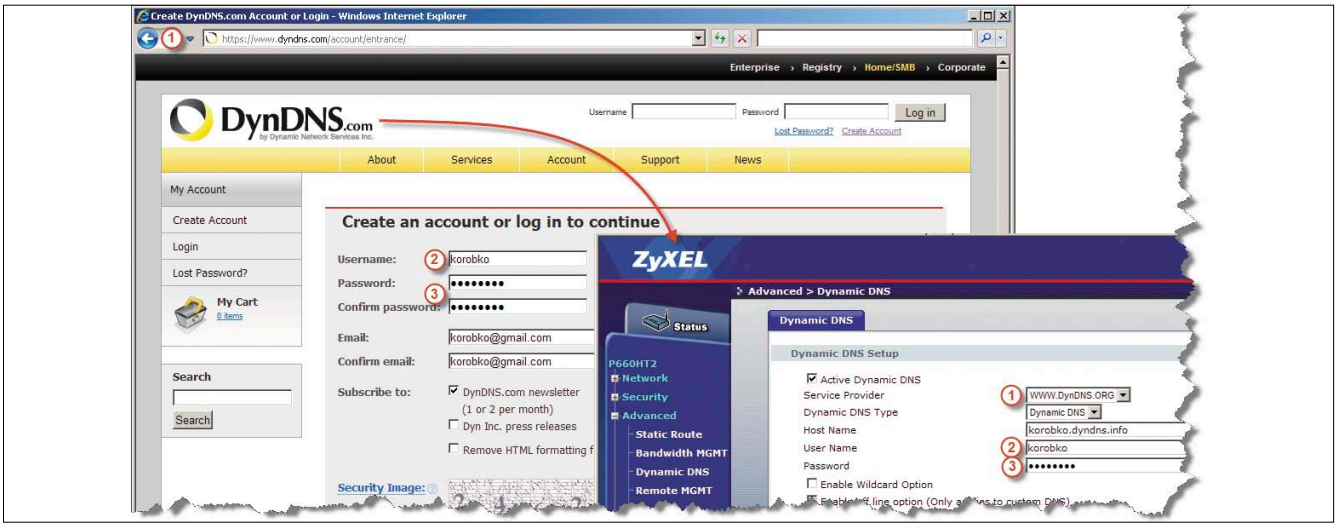
После завершения процесса создания учетной записи на указанный почтовый ящик будет отправлено письмо, в котором находится ссылка для активации созданной записи. После входа на сайт под зарегистрированным именем необходимо создать хост и выбрать доменное имя из предлагаемого списка (см. рис. 3), затем активировать сделанные изменения.

Настройка Dynamic DNS в ADSL-модеме

Любой современный ADSL-модем имеет встроенный веб-интерфейс. Однако не любой из них поддерживает службу Dynamic DNS. Прежде чем приобрести модем, необходимо ознакомиться с его спецификацией и убедиться в поддержке этой службы, выяснить список поддерживаемых DDNs-сайтов и убедиться, что в списке доменов второго уровня есть подходящий для вас. Несмотря на различия веб-интерфейсов различных модемов, идеология настройки службы DDNS везде одинакова.

Замечание: как правило, для входа на внутренний сайт модема используется стандартный адрес 192.168.1.1. Имя и пароль необходимо уточнить в документации или на сайте производителя модема. Обычно, именем для входа на сайт служит admin, паролем – admin или 1234.

Рисунок 2. Регистрация учетной записи на сервере DynDNS



Замечание: при попытке войти на сайт http://www.DynDNS.org осуществляется автоматическая переадресация на http://www.DynDNS.com.

На рис. 4 приведен скриншот настройки службы Dynamic DNS, которая условно разделена на две части.

В первой из них (Dynamic DNS Setup) осуществляется настройка соединения с сайтом – провайдером сервиса (www.DynDNS.org). Во второй части (IP Address Update Policy) осуществляется управление местоположением сайта, который можно создать на одном из компьютеров во внутренней сети.

Настройка Dynamic DNS Setup

В группе параметров Dynamic DNS Setup необходимо выбрать провайдера (на сайте которого зарегистрировано доменное имя), имя хоста и параметры учетной записи для входа на сайт провайдера. После того как все данные введены, необходимо активировать службу в ADSL-модеме. Для этого установите флажок напротив записи Active Dynamic DNS. На этом первичная настройка завершена. Удостоверьтесь, что соединение модема с сайтом установлено. Для этого необходимо выполнить несколько команд в оболочке командной строки:

ipconfig /flushdns – очищает кэш службы DNS на локальном компьютере;

ping korobko.DynDNS.info – позволяет определить, доступен ли сайт в данный момент времени.

Настройка IP Address Update Policy

В разделе IP Address Update Policy осуществляется настройка внутреннего IP-адреса, где будет располагаться сетевой ресурс, например веб-сайт. По умолчанию установлен IP-адрес маршрутизатора (Use WAN IP Address). Рекомендуется использовать либо автоматическое распознавание IP-адреса (Dynamic DNS server auto detect IP address), или указать IP-адрес (Use specified IP Address) компьютера внутренней сети, на котором расположен сайт или FTP-сервер.

Первый вариант настройки рекомендуется использовать при динамическом назначении IP-адресов в локальной сети, а второй – при статическом назначении.

Предположим, что все адреса локальной сети назначены статически и IP-адрес компьютера, на котором будет создан WEB- или FTP-сервер, 192.168.1.2. Исходя из этих предпосылок предпочтительнее всего выбрать третий вариант настройки (см. рис. 4, п. 2).

Установка и настойка IIS-сервера

IIS-сервер – стандартный компонент операционных систем семейства Windows, который не устанавливается по умолчанию.

Для его установки необходимо в «Панели управления» запустить оснастку «Установка и удаление программ». В появившемся диалоговом окне перейдите во вкладку «Установка компонентов Windows» и установите флажок напротив Internet Information Server (IIS).

После завершения процесса установки рекомендуется перезагрузить компьютер или воспользоваться утилитой iisreset.

Осуществляя настройку веб-сервера необходимо помнить, что пользователи, посещающие сайт, воспринимаются системой как anonimous, т.е. неопознанные. Поэтому в свойствах сервера необходимо отключить проверку учетных данных пользователей.

При использовании FTP-сервера рекомендуется деактивировать эту опцию. В этом случае для получения доступа к данным необходимо ввести имя и пароль локального пользователя, наделенного соответствующими правами.

Замечание: по умолчанию устанавливается только компонент WEB. Если необходимо установить FTP-сервер, то выберите соответствующий компонент в Internet Information Server (IIS).

***

Использование сервисов динамического DNS позволит обеспечить постоянный доступ к сетевым ресурсам, у которых периодически меняется IP-адрес. В частности, таким образом вы можете предоставлять доступ к файлам с использованием понятных человеку веб-адресов, тем самым заменяя файлообменники в Интернете. Кроме того, поддержка собственного сайта позволит сэкономить средства на хостинге, особенно, если это Windows Hosting (ASP/ASPX). EOF

Рисунок 3. Регистрация имени хоста

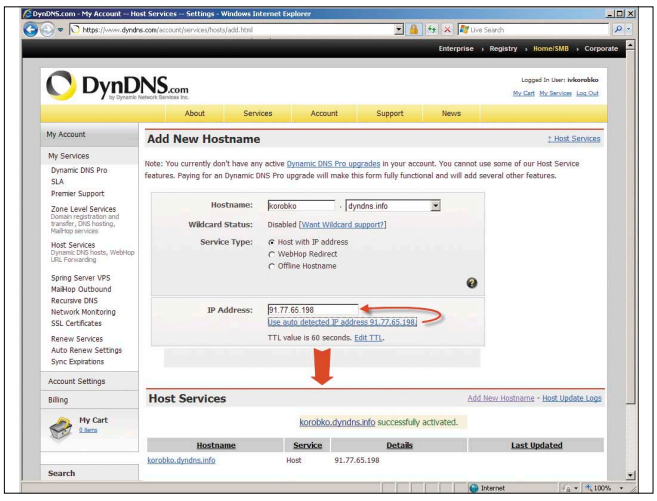
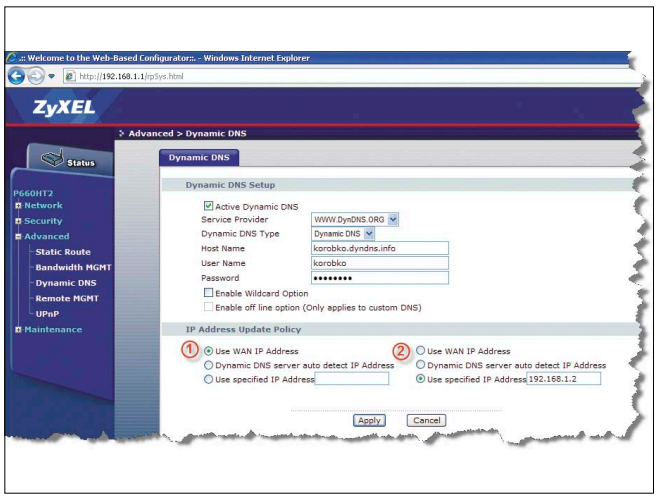


Рисунок 4. Настройка сервиса Dynamic DNS в ADSL-модеме



Визитка

СЕРГЕЙ СУПРУНОВ, инженер электросвязи широкого ИТ-профиля. В свободное время изучает FreeBSD и Python и пытается осмыслить свою нелюбовь к KDE

Конфигурируем DHCP-серверы
и настраиваем динамические обновления DNS *

Клиент, конечно, всегда прав. Но ровно настолько, насколько ему это позволено сервером

Установка и настройка DHCP-сервера ISC

Наиболее популярной реализацией DHCP-сервера в UNIX-подобных системах является dhcpd-разработка Internet Systems Consortium (ISC). По умолчанию в состав FreeBSD эта программа не входит. Она довольно легко устанавливается из коллекции «Портов»:

```
# cd /usr/ports/net/isc-dhcp30-server/
# make install
```

Но, к сожалению, на момент подготовки статьи единственная версия, которую можно было без проблем установить – 3.0.7, – была сильно устаревшей (в марте 2009 года официально прекращена её поддержка).

В итоге было принято решение ставить ISC DHCP версии 4.1.0 из исходников (команда «./configure --help» после распаковки позволит просмотреть доступные опции конфигуратора; возможно, некоторые из них вы захотите использовать):

```
# fetch http://ftp.isc.org/isc/dhcp/dhcp-4.1.0.tar.gz
```

```
dhcp-4.1.0.tar.gz       100% of 1061 kB  174 kBps
```

```
# tar xzvf dhcp-4.1.0.tar.gz
# cd dhcp-4.1.0
# ./configure
# make
# make install
```

После установки придётся вручную подготовить сценарий автозапуска. За основу можно взять какой-нибудь из имеющихся в /usr/local/etc/rc.d или /etc/rc.d. Я здесь немного схитрил и воспользовался сценарием из порта isc-dhcp-30-server:

```
# cd /usr/ports/net/isc-dhcp30-server
# mkdir work
# make apply-slist
# cp work/isc-dhcpd /usr/local/etc/rc.d/
```

Поскольку имя демона и большинство ключей запуска совпадают, такая подмена не должна вызвать проблем.

Есть один важный момент – для работы DHCP-сервера ядро системы должно быть собрано с поддержкой псевдоустройства bpf (Berkeley packet filter; используется для получения «сырых» данных с интерфейса, в т.ч. широковещательных пакетов). В ядро GENERIC эта поддержка всегда включается, так что если вы не исключали её явно, то пересборка ядра потребоваться не должна.

Проверить, включено ли это устройство в ваше ядро, можно так:

```
$ grep bpf /usr/src/sys/`uname -p`/conf/`uname -i`
```

```
# The `bpf' device enables the Berkeley Packet Filter.
# Note that 'bpf' is required for DHCP.
device          bpf             # Berkeley packet filter
```

Теперь добавим в /etc/rc.conf пару строк (правда, это будет работать лишь при условии, что обработка переменных предусмотрена сценарием автозапуска; в isc-dhcpd, который мы «выдрали» из портов, она предусмотрена):

```
dhcpd_enable="YES"
dhcpd_ifaces="nfe0"
```

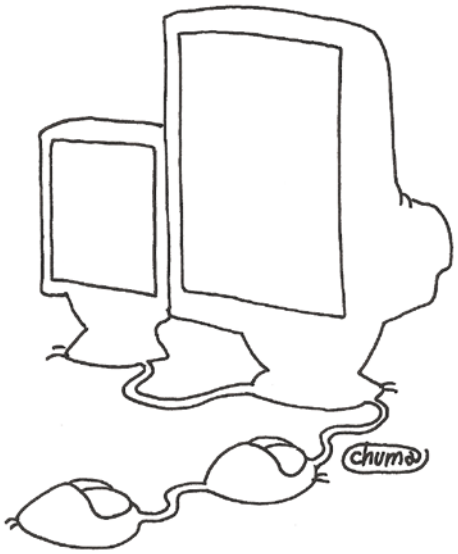
Основные параметры задаются в файле /usr/local/etc/dhcpd.conf (файл-пример будет установлен во время инсталляции).

Рассмотрим пример не сложной, но вполне работоспособной конфигурации:

```
# Доменное имя. Имена хостов клиентов будут дополняться
# до FQHN
option domain-name "example.org";

# DNS-серверы, которые будут предлагаться клиентам.
# Можно использовать и IP-адреса оных
option domain-name-servers ns1.example.org, ↵
        ns2.example.org;
```

* В статье «Сисадмин должен быть ленив. DHCP и динамический DNS» (см. №8 за 2009 г.) были рассмотрены общие принципы работы протокола DHCP, а также использование DHCP-клиентов.



DHCP позволяет автоматизировать весьма значительную часть процесса настройки сети

```
# «Умолчальное» и максимальное времена аренды адреса
# в секундах
default-lease-time 3600;
max-lease-time 86400;

# Является ли сервер авторитативным
authoritative;

# Способ динамического обновления DNS.
# Подробнее поговорим позже, сейчас отключим
ddns-update-style none;

# Источник сообщений для записи логов через syslogd
log-facility local7;

# Объявление подсети
subnet 192.168.1.0 netmask 255.255.255.0 {
   range 192.168.1.200 192.168.1.249;
   option routers 192.168.1.1;
}
```

В самом начале файла размещаются глобальные параметры, которые при необходимости могут быть переопределены далее, в отдельных «объявлениях» подсетей и диапазонов. Обычно здесь задаются имя домена, список DNS-серверов (если используются одни и те же для большинства обслуживаемых сетей), значения для времени аренды (по умолчанию максимальное, при необходимости можно задать минимальное).

Параметр authoritative позволяет объявить сервер авторитативным (ответственным) в обслуживаемой сети. Отличие авторитативного сервера от «обычного» заключается в том, что последний игнорирует любые запросы адресов, которые не описаны в его конфигурации, в то время как авторитативный сервер в ответ на такие запросы отсылает DHCPNAK. Благодаря такому поведению клиент, перемещённый из другой подсети, сможет быстрее получить новый адрес (в ответ на DHCPREQUEST с адресом из прежней подсети он сразу получит DHCPNAK и приступит к получению нового адреса; в противном случае DHCPDISCOVER будет отправлен лишь по истечении тайм-аута на ожидание ответа). В то же время «случайные» DHCP-серверы (например, ошибочно запущенные с настройками из файла-примера) с меньшей вероятностью смогут помешать работе сети, поскольку, будучи неавторитативными, будут просто игнорировать «чужие» запросы DHCPREQUEST.

Для ведения логов (а они никогда лишними не будут), помимо объявления источника (facility) в конфигурации dhcpd, вам нужно будет сделать ещё две вещи: создать файл протокола (например, командой «touch /var/log/dhcpd.log») и добавить строку в /etc/syslog.conf:

```
local7.*            /var/log/dhcpd.log
```

После чего syslogd нужно перезапустить:

```
/etc/rc.d/syslogd restart
```

Ну и не забудьте настроить ротацию данного лог-файла в /etc/newsyslog.conf.

Вернёмся к конфигурации dhcpd. Всё самое интересное содержится в описании подсети subnet. В нашем примере всё элементарно – строкой range мы задаём диапазон адресов, которые dhcpd сможет «сдавать в аренду» клиентам, опция routers задаёт список маршрутов по умолчанию. Ещё одна обязательная для работы в сети настройка – адреса DNS-серверов – будет получена из глобальной опции domain-name-servers.

Обратите внимание на то, что именно по описанию subnet авторитативный сервер будет различать «допустимые» и «недопустимые» адреса. Так, сервер, запущенный с приведённой выше конфигурацией, в ответ на запрос (DHCPREQUEST) адреса 192.168.0.22 будет возвращать DHCPNAK (поскольку запрошенный адрес в известную ему подсеть не попадает), но «промолчит» при запросе адреса 192.168.1.22 (т.к. этот адрес, хотя и не включён ни в один из диапазонов range, является правильным для данной подсети и вполне может обслуживаться вторым DHCP-сервером; об этом чуть подробнее поговорим через раздел).

Помимо диапазона адресов и шлюза в описании подсети можно задать огромное множество дополнительных опций. Полный список поддерживаемых опций можно найти в справке: man dhcp-options(5).

Если несколько подсетей доступны через один интер-

WIDE DHCP

Нужно сказать, что ISC DHCP – не единственный вариант. В коллекции портов можно найти ещё один DHCP-сервер – WIDE DHCP. Правда, этот проект трудно назвать «действующим» – текущая версия в «Портах» (1.4.0.6_2) практически не обновлялась с 2003 года, страничка проекта (http://www.sfc.wide.ad.jp/~tomy/dhcp/index-e.html) заброшена. Тем не менее с основной задачей он вполне справляется.

Установка из коллекции портов потребует дополнительной работы. Начало традиционно:

```
# cd /usr/ports/net-mgmt/wide-dhcp
# make install
```

В итоге в /usr/local/sbin появится файл dhcps, будет создан сценарий автозапуска, а также добавятся соответствующие man-страницы.

После этого нужно подправить сценарий автозапуска /usr/local/etc/rc.d/wide-dhcps.sh.sample (и переименовать его в wide-dhcps.sh). Мало того, что он в «старом» формате (т.е. не подготовлен для утилиты rcorder), так ещё и недоработан. Пришлось вручную ввести переменную PREFIX и добавить в строку запуска dhcps опции, определяющие местоположение конфигурационных файлов и баз данных. В этой же строке нужно указать интерфейсы, на которых dhcps должен работать.

Далее файлы dhcpdb.pool и dhcpdb.relay нужно будет создать вручную (по умолчанию в /etc, я изменил каталог на /usr/local/etc, как это принято во FreeBSD). Примеры можно найти в каталоге db_sample дистрибутива. Второй из них служит для указания маршрутизаторов, через которые доступны другие подсети, обслуживаемые этим DHCP-сервером, и в случае «линейной» структуры сети его можно оставить пустым (но сам файл должен существовать). Первый же – это основной конфигурационный файл. Приведу несложный пример:

```
# Создаём подсеть (сюда выносим общие параметры:
# маску, шлюз и широковещательный адрес)
subnet:snmk=255.255.255.0:rout=10.0.0.1:brda=10.0.0.255:
# Далее идут описания каждого адреса пула
# (первое поле - просто идентификатор записи,
# далее задаются IP-адрес, время аренды (по умолч. и макс.),
# а также подключается определённая выше конфигурация subnet)
ip198: :ipad=10.0.0.198:dfll=3600:maxl=7200:tblc=subnet:
ip199: :ipad=10.0.0.199:dfll=3600:maxl=7200:tblc=subnet:
```

Как видите, для каждого адреса пула нужно прописывать свою строку, но зато и каждому адресу можно выдать свои параметры. Используется тот же формат файла, что и для системных баз termcap, printcap и т.п.; список доступных параметров достаточно обширен (найти его можно на странице справки man dhcpdb.pool(5)).

После всех этих мучений сервер можно запускать, и он должен корректно функционировать. Правда, учитывая запутанный синтаксис конфигурации, я для себя не нашёл ни одной причины, почему эту реализацию можно было бы использовать вместо ISC. Но знать о её существовании в любом случае полезно.

фейс, то их объявления subnet должны быть вложены в объявление shared-network:

```
shared-network rl0-net {
    subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.100 192.168.1.199;
        option routers 192.168.1.1;
    }
    subnet 192.168.2.0 netmask 255.255.255.0 {
        range 192.168.2.100 192.168.2.199;
        option routers 192.168.2.1;
    }
}
```

Общие для всех подсетей опции можно вынести непосредственно в объявление shared-network – в этом случае они будут влиять на все объявления subnet.

Пулы и классы

Иногда возникает необходимость разделять клиенты по тому или иному признаку и выдавать им разные опции. Например, мы хотим клиентам, имя хоста которых начинается на «a» (например, «acer»), раздавать адреса из диапазона 10.0.0.10 – 10.0.0.19, а всем остальным – из 10.0.0.20 – 10.0.0.99. Для этого можно использовать объявление класса и два так называемых пула:

```
class "a-clients" {
  match if substring (option host-name, 0, 1) = "a";
}

subnet 10.0.0.0 netmask 255.255.255.0 {
   pool {
      allow members of "a-clients";
      range 10.0.0.10 10.0.0.19;
   }
   pool {
      deny members of "a-clients";
      range 10.0.0.20 10.0.0.99;
   }
}
```

То есть мы отнесли к классу a-clients клиентские машины согласно приведённому выражению, а затем создали два пула, в одном из которых разрешили обслуживание членов соответствующего класса, в другом, наоборот, запретили. Аналогично можно строить выражения по MAC-адресу (переменная hardware) и другим опциям. Подробнее о синтаксисе выражений, допустимых в конфигурации dhcpd, можно почитать на странице man dhcp-eval(5).

Помимо признака членства в определённом классе команды allow/deny поддерживают выражения unknown-clients (клиенты, не передавшие свои имена хостов), known-clients (соответственно передавшие) и all clients (любые клиенты). Подробности ищите в документации.

Фиксированные адреса

Иногда возникает необходимость более чётко контролировать получение адресов некоторыми хостами. Например, выход в Интернет требуется лишь некоторым компьютерам локальной сети, а на прокси-сервере доступ регулируется по IP-адресу. Можно «избранным» хостам назначать вручную адреса, не попадающие в интервал, обслуживаемый сервером. А можно воспользоваться механизмом назначения фиксированных адресов, который предоставляет dhcpd:

```
host acer {
   hardware ethernet 00:1b:38:22:8c:17;
   fixed-address 10.161.193.177;
}
```

Если добавить этот фрагмент в конфигурационный файл (и не забыть перезапустить dhcpd), то данный хост будет всегда получать указанный IP-адрес. Идентификация хоста будет осуществляться по MAC-адресу. IP-адреса, закреплённые таким образом за определёнными хостами, не должны попадать ни в один из диапазонов range.

Если нужно описать несколько хостов, имеющих много одинаковых опций, их можно объединить в группу:

```
group {
   ...общие опции...
   host acer { ...специфичные для хоста опции ...}
```

```
    host fuji { ...специфичные для хоста опции ...}
}
```

В этом случае общие опции выносятся в объявление группы, а индивидуальные остаются в объявлениях host.

Особенности использования нескольких DHCP-серверов

В сравнительно больших сетях по соображениям надёжности и балансировки нагрузки обычно используют несколько DHCP-серверов. Очевидно, что при этом необходимо избегать «перекрытия» адресного пространства, когда один и тот же IP-адрес может быть выдан различными серверами. В противном случае возможен конфликт – ведь если сервер «А» выдаст клиенту некоторый адрес, то сервер «Б» по-прежнему будет считать его свободным и может выдать его другому клиенту (клиент, перед тем как принять IP-адрес, должен с помощью ARP-запроса убедиться, что тот свободен; это несколько снижает вероятность конфликта, но всё же не исключает его полностью).

Классическая рекомендация, известная как «правило 80/20», звучит так: один сервер (основной) должен обслуживать 80% адресов пула, второй сервер (вспомогательный) – оставшиеся 20%. Это позволит сети «продержаться» некоторое время на одном вспомогательном сервере в случае проблем с основным, при условии, что не все клиенты начнут запрашивать адреса одновременно. Правда, применяя это правило в своей сети, желательно убедиться, что именно основной сервер является более быстрым – иначе вспомогательный сразу раздаст свой пул клиентам, и при возникновении нештатной ситуации ему просто нечего будет им предложить. (В настройках сервера ISC можно задать параметр min-secs, задающий задержку в секундах перед выдачей ответа клиенту. Её использование на вспомогательном сервере повысит шансы на то, что первым будет отвечать основной.)

Возникает вопрос – а не будут ли мешать друг другу два авторитативных DHCP-сервера? Если у них будет одинаковое объявление подсети (но разные, непересекающиеся интервалы адресов, описанные в range), то запрос адреса из такой подсети, даже не попадающий в обслуживаемый конкретным сервером диапазон, не будет рассматриваться как «чужой» – сервер просто ничего не будет отвечать, позволяя тем самым обработать этот запрос другому серверу. Если этот «другой сервер» недоступен, то клиент, не дождавшись ответа на DHCPREQUEST, отправит запрос DHCPDISCOVER и будет благополучно обслужен оставшимся в строю сервером.

Некоторые серверы (в частности, ISC), поддерживают так называемый механизм DHC-FAILOVER (подготовка соответствующего стандарта остановилась на документе

Вопросы безопасности

К сожалению, безопасность протокола DHCP пока оставляет желать лучшего. Рабочими группами IETF предлагаются различные методы её повышения (например, аутентификация клиентов, см. RFC 3118), но они в большинстве своём ещё нигде не реализованы.

Пока же наиболее действенной является рекомендация закрыть UDP-порты 67 и 68 по периметру сети и по возможности более строго контролировать оборудование, работающее внутри.

DHCP Relay

Как упоминалось в первой части статьи, в протоколе DHCP активно используются широковещательные запросы, которые практически всегда отбрасываются маршрутизаторами. То есть их распространение обычно ограничено одним сегментом локальной сети.

Однако зачастую оказывается нецелесообразно устанавливать отдельный DHCP-сервер в каждом сегменте. В этом случае на помощь приходит способность большинства маршрутизаторов работать в режиме DHCP Relay, «перебрасывая» запросы и ответы между отдельными подсетями. Принцип работы такого «прокси-сервера» заключается в следующем: получив на одном из обслуживаемых интерфейсов широковещательный DHCP-запрос, он перенаправляет его (уже обычным, одноадресным пакетом) определённым в его конфигурации DHCP-серверам. Полученный ответ транслируется широковещательным (при необходимости) пакетом в ту подсеть, откуда пришёл исходный запрос.

Большинство аппаратных маршрутизаторов функцию DHCP Relay поддерживают. Если же роль маршрутизатора между вашими подсетями выполняет FreeBSD или Linux, то можно установить либо программу dhcrelay, входящую в состав пакета ISC DHCP, либо отдельный сервер из коллекции портов – /usr/ports/net/dhcprelay. Настройка в обоих случаях сводится к запуску этой программы с опциями, определяющими список прослушиваемых интерфейсов и список DHCP-серверов, которым запрос нужно ретранслировать. Во FreeBSD в случае программы dhcprelay для её автозапуска достаточно включить в /etc/rc.conf три строки:

```
dhcprelay_enable='YES'
dhcprelay_server='10.0.0.220'
dhcprelay_ifaces='ed0
```

Кстати, функция DHCP Relay в некотором роде повышает управляемость сети, поскольку позволяет явно задать список DHCP-серверов, с которыми следует работать.

http://tools.ietf.org/html/draft-ietf-dhc-failover-12), предоставляющий двум серверам возможность разделять общий пул IP-адресов, синхронизируя информацию о выданных адресах и позволяя динамически подменять друг друга при необходимости.

Динамический DNS

Итак, задачу автоматического получения настроек компьютерами сети мы решили. Но остался ещё один вопрос – интеграция с DNS-сервером. Конечно, в большинстве сетей можно обойтись и без этого – доступ по имени обычно бывает нужен только на серверы, которые в свою очередь практически всегда настраиваются вручную, и потому статического DNS вполне достаточно. Но иногда всё же бывает удобно, когда каждый клиентский компьютер доступен в сети под своим именем (особенно если на постоянство IP-адреса положиться нельзя), поэтому рассмотрим, как настроить динамическое обновление DNS-записей (предполагая, что в качестве DNS-сервера используется ISC BIND).

Прежде всего в настройках DNS-сервера нужно разрешить автоматические обновления:

```
# Объявляем ключ доступа (можно задавать и в каждой зоне,
# но так удобнее)
key DHCP_KEY {
   algorithm hmac-md5;
   secret "c20f9433f5f5ecf1f245a6112d7dd651";
};

# «Прямая» зона
zone "test.inr" {
   type master;
   allow-update { key DHCP_KEY; };
   file "master/test.inr";
};

# «Обратная» зона
zone "0.0.10.in-addr.arpa" {
   type master;
   allow-update { key DHCP_KEY; };
   file "master/0.0.10.in-addr.arpa";
};
```

То есть мы объявляем ключ DHCP_KEY (имя может быть любым) и затем указываем его как условие, при котором будет разрешено обновление зоны (в описании всех зон, которые должны автоматически обновляться). Для генерации ключа (в строке secret) можно воспользоваться утилитой mmencode:

```
$ echo 'Super secret key' | mmencode

U3VwZXIgc2VjcmV0IGtleQo=
```

Неплохо справляется с задачей утилита md5:

```
$ echo 'Super secret key' | md5

25ecc0ad8ba5c6b56d85c8ae9811e881
```

Хотя более «правильным» способом формирования ключа считается использование утилиты dhssec-keygen:

```
$ dnssec-keygen -a HMAC-MD5 -b 128 -n HOST test.inr

Ktest.inr.+157+41531

$ ls -l Ktest.inr.+157+41531.*

-rw-------  1 amsand  amsand  52 11 авг 19:08
Ktest.inr.+157+41531.key
-rw-------  1 amsand  amsand  92 11 авг 19:08
Ktest.inr.+157+41531.private
```

Здесь мы создаём «хостовый» ключ длиной 128 бит, используя алгоритм HMAC-MD5, с именем test.inr. В результате формируется два файла – ключ можно извлечь из любого.

Ну и для повышения безопасности (поскольку файл named.conf обычно доступен на чтение всем пользователям) оператор key выносят в отдельный файл, доступный на чтение только пользователю root, а в named.conf подключают его с помощью оператора include:

```
include "key.conf";
```

Вместо allow-update можно использовать более «мощную» секцию update-policy (дополнительно ограничим тип записей и поддомен):

```
zone "test.inr" {
   type master;
   update-policy {
      grant DHCP_KEY subdomain test.inr A TXT;
   };
   file "master/test.inr";
};
```

Теперь осталось внести некоторые изменения в конфигурацию DHCP:

```
# Указываем метод обновления (существует ещё ad-hoc,
# но он не рекомендуется)
ddns-update-style interim;
```

```
# Описываем тот же ключ (можно просто скопировать
# из named.conf - синтаксис тот же)
# Если оператор key вынесен в отдельный файл, можно,
# как и в named.conf, использовать оператор include:
### include "key.conf";
key DHCP_KEY {
   algorithm hmac-md5;
   secret "c20f9433f5f5ecf1f245a6112d7dd651";
}

# «Прямая» зона, которую нужно обновлять
zone test.inr {
   primary 10.0.0.220;
   key DHCP_KEY;
}

# «Обратная» зона, которую нужно обновлять
zone 0.0.10.in-addr.arpa {
   primary 10.0.0.220;
   key DHCP_KEY;
}
```

В параметре primary при описании зон указывается адрес первичного DNS-сервера, обслуживающего зону (т.е. того, на который следует отправлять обновления).

Теперь осталось убедиться, что пользователь, от имени которого выполняется процесс named (на FreeBSD это обычно bind), имеет право создавать файлы в каталоге /var/named/etc/namedb/master.

Если всё сделано правильно, то после того как клиент в следующий раз запросит адрес, в каталоге master появятся файлы, соответствующие файлам зон, с расширением jnl. Это – журналы обновлений, используемые сервером для восстановления соответствующей информации после перезагрузки. В них-то и будут храниться соответствующие ресурсные записи (в двоичном формате, поэтому их чтение ничего полезного вам не даст).

В случае проблем обращайтесь к лог-файлам. Ниже показаны два сообщения из /var/log/messages, с которыми приходится сталкиваться наиболее часто:

```
May  4 17:23:14 freetest dhcpd: if acer.example.org
IN A rrset doesn't exist add acer.example.org 300
IN A 10.0.0.180: timed out.
May  4 17:27:23 freetest named[95949]: master/test.inr.jnl:
create: permission denied
```

Первая запись в данном случае вызвана нестыковкой доменных имён – в конфигурации DHCP был задан «умолчальный» домен example.org, который нашим DNS-сервером не обслуживается. К сожалению, это не единственная причина возникновения тайм-аута – следует рассматривать также права доступа к соответствующим файлам, правила пакетных фильтров (если DHCP и DNS работают на разных машинах), правильность указания адреса DNS-сервера.

Вторая указывает на то, что процесс named не может создать jnl-файл в каталоге master. Очевидно, что проблема кроется в правах доступа – позаботьтесь, чтобы процесс named мог создавать файлы в каталоге master, и проблема исчезнет.

***

Как видите, DHCP – довольно мощный протокол, позволяющий автоматизировать весьма значительную часть процесса настройки сети. К сожалению, из-за некоторой «сырости» в плане стандартизации и определённых проблем безопасности о 100-процентной автоматизации пока говорить не приходится. Но это не мешает его использовать уже прямо сейчас. EOF

Утилита nsupdate

В дистрибутиве ISC BIND есть утилита, позволяющая отправлять динамические обновления DNS. Она может пригодиться как для поиска проблем (например, если при выдаче клиенту адреса сервером DHCP обновление зоны не происходит, но через nsupdate зоны обновляются нормально, становится очевидным, что следует разбираться с конфигурацией dhcpd), так и для обновления зон «вручную».

Рассмотрим типичный пример работы:

```
$ nsupdate -v
> server 10.0.0.220
> key DHCP_KEY c20f9433f5f5ecf1f245a6112d7dd651
> update add new.test.inr 300 A 10.1.1.15
> show

Outgoing update query:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
;; flags: ; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
;; UPDATE SECTION:
new.test.inr.          300     IN      A       10.1.1.15

> send
> quit
```

То есть мы объявляем DNS-сервер и секретный ключ, после чего командой update add помещаем в «очередь» команду на добавление соответствующей записи (300 в данном примере – время жизни записи). Командой show можно просмотреть текущее состояние «очереди», send отправляет данный пакет серверу. Если всё прошло нормально (а поскольку никаких сообщений об ошибках не выведено, то так и должно быть), то, запросив у DNS-сервера адрес для new.test.inr, вы получите 10.1.1.15. (Да, этот адрес не является допустимым для нашей подсети, но в эти «тонкости» DNS-сервер не вдаётся – он просто делает свою работу.)

Помимо интерактивного режима работы nsupdate позволяет выполнять команды из файла, что может оказаться полезным для автоматического выполнения обновлений. Подробности ищите в справке man nsupdate(8).

1. Страница официального сайта ISC DHCP – https://www.isc.org/software/dhcp.
2. Колисниченко Д. Конфигурирование DHCP. //Системный администратор, №5, 2003 г. – С. 12-14 (http://www.algoint.ru/?MenuItem=tech_dhcp2).
3. Иванов П. DHCP: искусство управления IP-адресами. – http://www.citforum.ru/internet/tifamily/dhcp.shtml.
4. Bog BOS: Протокол BOOTP/DHCP – http://www.bog.pp.ru/work/bootp.html.
5. RFC 2131. Dynamic Host Configuration Protocol – http://tools.ietf.org/html/rfc2131.
6. RFC 2132. DHCP Options and BOOTP Vendor Extensions – http://tools.ietf.org/html/rfc2132.
7. DHCP Failover Protocol – http://tools.ietf.org/html/draft-ietf-dhc-failover-12.
8. DHCP Failover – http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ciscoasu/nr/nr3.0/concepts/cg03.htm.
9. Authentication for DHCP Messages – http://www.ietf.org/rfc/rfc3118.txt.
10. Настройка статических маршрутов через DHCP в Windows и Linux – http://www.linux.by/wiki/index.php/FAQ_DHCP_routes.
11. RFC 2136. Dynamic Updates in the Domain Name System (DNS UPDATE) – http://www.ietf.org/rfc/rfc2136.txt.

Визитка

СЕРГЕЙ ЯРЕМЧУК, инженер автоматизации. Автор более 800 статей и 4 книг. В «Системном администраторе» публикуется с первого номера. Интересы: сетевые технологии, защита информации, свободные ОС

Альтернативы MS Project

Пробуем популярные Open Source-решения

В любой компании не обойтись без инструмента, позволяющего оптимально распределить время, задачи, доступные средства и правильно настроить бизнес-процессы

Можно, конечно, все планы держать в голове или отмечать процесс на бумаге, самостоятельно отслеживая прогресс, но такой способ может подойти только для сравнительно небольших организаций и проектов.

Наиболее разрекламированным и потому известным продуктом для управления проектами является Microsoft Project [1], поставляемый в двух комплектациях – Standard и Professional. Вариант Standard – однопользовательская версия, предназначенная для небольших проектов, в тех случаях, когда руководителю не требуются средства совместной работы и координации деятельности. Версия Professional – корпоративный вариант, поддерживающий совместное управление проектами и ресурсами с возможностью подключения к Microsoft Office Project Server 2007 и Microsoft Office Project Web Access 2007. Стоят обе версии MS Project необоснованно дорого (600 и 1000$ соответственно), поэтому попробуем найти им замену.

Следует заметить, что средства для планирования имеются в большинстве решений групповой работы (GroupWare). Например, в eGroupWare [2, 3] присутствуют модуль ProjectManager (управление проектами) и календарь. Созданный силами ProjectManager проект имеет все необходимые атрибуты: категория, приоритет, дата начала/окончания, доступ, список участников, бюджет, диаграммы Ганта и прочие данные. Аналогичный модуль есть и у Group-Office [4]. Это решение имеет бесплатную версию, распространяемую в исходных текстах под лицензией AGPL. А также в Open-Xchange [5] и некоторых других. Есть модуль управления проектами даже в некоторых CRM, например Project Management в SugarCRM. Дальше сосредоточимся на специализированных решениях.

OpenProj

Serena Software предлагает два варианта продукта: платный POD (Projects On Demand), предоставляемый в виде услуги (SaaS, Software as a Service), имеющего функции совместной работы над проектом и бесплатного OpenProj [6], предназначенного для персонального использования и являющегося аналогом однопользовательского MS Project Standard. Оба используют одну и ту же кодовую базу, поэтому основные возможности и принципы работы их в основном совпадают, также возможен обмен сохраненными проектами. Стоимость платной подписки POD составляет 20$ в месяц, что немного, учитывая, что для реализации сервиса не потребуется дополнительная закупка оборудования и ОС, которые тоже обойдутся в некоторую сумму. Бесплатная версия OpenProj включена в состав Star Office, продаваемого в Европе. По статистике проекта, OpenProj скачан более 1,250 миллиона раз, что является отличным показателем.

Написан OpenProj на Java и может быть запущен на любой платформе, для которой есть JRE (Java Runtime Environment). Разработчики предлагают установочные пакеты для Windows, Linux (есть и rpm, и deb) и Mac. Системные требования разработчиками не заявлены, но опыт показывает, что для запуска достаточно минимальных требований, предъявляемых к ОС. Интерфейс OpenProj локализован и внешне напоминает MS Project, хотя дизайн выглядит несколько устаревшим. Зато возможности достаточно большие. В OpenProj доступны диаграммы Ганта, гексаграммы и таблицы, отображающие ресурсы, задействованные в проекте, свободные ресурсы, сетевые графики, диаграммы освоенных объемов работ и использованных ресурсов, задач-предшественников и задач-последователей, также фактических затрат. Доступен ряд отчетов – по проекту, ресурсам, задачам и исполнителям.

В комплекте из документации поставляются только подсказки (Tips), при необходимости можно получить онлайн-помощь. На сайте проекта есть раздел с незаконченной документацией на русском языке [7]. Хотя программа построена логично, поэтому пользователь, имеющий опыт работы с подобными решениями или понимающий процесс, без труда разберется с созданием проекта.

Немаловажно, что OpenProj поддерживает импорт файлов Microsoft Project вплоть до 2007 и других продуктов, в которых возможно сохранение проектов в .mpp, .xml и .mpx, в том числе и Gnome Planner. Экспорт возможен

В различных ситуациях побеждают разные свойства, придется подбирать решение по себе

в XML, совместимый с MS Project 2003, и собственный формат (.pod). Экспорт в формат PDF реализован только в коммерческом POD.

OpenWorkbench

Начало этого проекта датировано 1984 годом, тогда он назывался «Project Workbench». Через некоторое время после покупки в 2005 году корпорацией CA компании Niko, которой принадлежали права на этот продукт, стал доступен код, и проект получил новое имя OpenWorkbench. Первое время OpenWorkbench выпускался как betaware, сейчас распространяется под freeware-лицензией. Позиционируется как открытая альтернатива Microsoft Project для использования на персональном компьютере. За дополнительную плату доступен дополнительный модуль Clarity Schedule Connect, который позволяет хранить проекты на централизованной базе с возможностью совместной работы. Написан на Java, но распространяется только для Windows 2000, XP, 2003 и Vista. Интерфейс переведен на английский, французский и немецкий языки. Хотя есть возможность самостоятельно локализовать таблицы в проекте.

В OW за основу проекта берутся возможности ресурса, помноженные на количество часов, необходимое для решения задачи. В этом его отличие от MS Project и некоторых других решений, в которых расчет ведется от требуемого времени выполнения, а не потенциальных возможностей организации. Поэтому области применения этих продуктов могут отличаться. Если, к примеру, необходимо рассчитать, за сколько времени 2 человека выполнят определенную работу, здесь удобнее OW. А если стоит задача к определенному дню настроить сеть и рассчитать, сколько потребуется человек, то MP лучше оптимизирован для этого. Хотя в настройках проекта OW можно указать фиксированный срок выполнения. Учитывая, что в большинстве случаев ограничены именно ресурсы, OW подходит для большинства организаций малого и среднего бизнеса.

Возможны создание зависимостей между задачами проекта и между проектами, субпроекты, автоматическое создание расписания, WBS, диаграммы Ганта, PERT и CPM, несколько методов отслеживания хода процесса, расчет стоимости. Вывод данных можно изменить в зависимости от характера проекта.

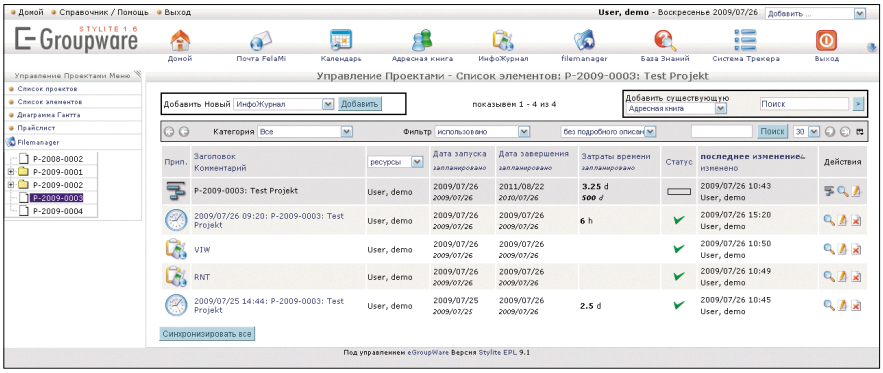
Возможен импорт и экспорт как в файл собственного формата (.rmp), так и в XML-файл, который можно открыть в других программах, поддерживающих такой тип данных.

С ходу научиться работать с OpenWorkbench, скорее всего, не получится. На сайте проекта доступно 301-страничное руководство пользователя (на английском), которое поможет освоиться.

GanttProject

GanttProject [9] – бесплатная кроссплатформенная программа для управления проектами в диаграмме Ганта. Исходные тексты доступны под лицензией GNU GPL, поддерживается работа на Windows, Linux и MacOSX. Проект представляет дерево задач,

Рисунок 1. Модуль для работы с проектами в eGroupWare



Термины

Диаграмма Ганта – распространенный тип горизонтальных диаграмм, который используется для иллюстрации плана, графика работ или структуры задач какого-либо проекта. С левой стороны обычно показана задача, поле вверху соответствует дате. Вертикальная линия, проведенная по текущей дате, будет соответствовать ходу выполнения задачи. Помимо временной последовательности задач диаграмма отображает и связи между задачами. Разработана Генри Л. Гантом (Henry L. Gantt) в 1910 году.

Сетевой график (PERT, Program Evaluation and Review Technique) – диаграмма, отображающая зависимости между задачами проекта.

Critical Path Method (CPM) – последовательности задач от начала проекта до его окончания с учетом их взаимосвязи. Критические задачи обладают наименьшей гибкостью для планирования и напрямую влияют на сроки реализации проекта.

WBS (Work breakdown structure) – структурная декомпозиция работ, определяет перечень элементарных работ проекта.

RBS (Risk Breakdown Structure) – иерархически организованное представление возможных рисков проекта.

Ресурс – в контексте человек или группа людей, материалы или оборудование.

для которых выделяется определенное количество времени и закрепляется исполнитель. Между задачами устанавливаются связи. Все это выводится при помощи диаграмм Ганта и PERT, заявленные работы над CPM (Critical Path Method) пока не доведены до конца. Для удобства на диаграмму можно вывести дополнительные данные (координатор, дата, название задачи, назначенные ресурсы). Реализована подсветка занятости человека (обычный человек, завален работой, недорабатывает и отпускные дни) и хода выполнения задач (не изменилось, опережение и запаздывание). Есть возможность задания пользовательских полей в задачах. Интерфейс программы локализован и, можно сказать, стандартен, поэтому проблем с его освоением быть не должно. Принцип прост: вначале создается пустая задача, которая отображается на графике, затем вызывается и редактируется окно свойств. В качестве ресурсов подразумевается человек. При создании учетной записи нового человека задаются имя, роль, телефон, электронный адрес и отпускные дни. По умолчанию могут быть установлены две роли – «координатор» и «неопределенно», но в настройках программы можно задать любое количество ролей.

Для совместимости со сторонними проектами используется библиотека MPXJ (http://mpxj.sourceforge.net). Но из пяти поддерживаемых этой библиотекой форматов в GanttProject реализован только один – MS Project Exchange (MPX). Кроме этого, есть возможность экспорта и импорта файлов, проектов в формате txt и XML. Файлы XML, экспортированные из OpenProj и OpenWorkbench, в GanttProject открыть не удалось.

Отчеты можно сохранить в HTML, PDF, рисунок (PNG, JPEG) и CSV (с возможностью выбора своего варианта разделителя). Последнее позволяет в дальнейшем использовать программы для работы с электронными таблицами.

Еще одна важная особенность GanttProject – возможность загрузки и сохранения файла проекта на FTP, что позволяет открывать документ сразу несколькими пользователями. Правда, коллизии при редактировании придется разрешать вручную. Также GanttProject может использоваться для оффлайн-редактирования проектов для]project-open[(о нем чуть позже).

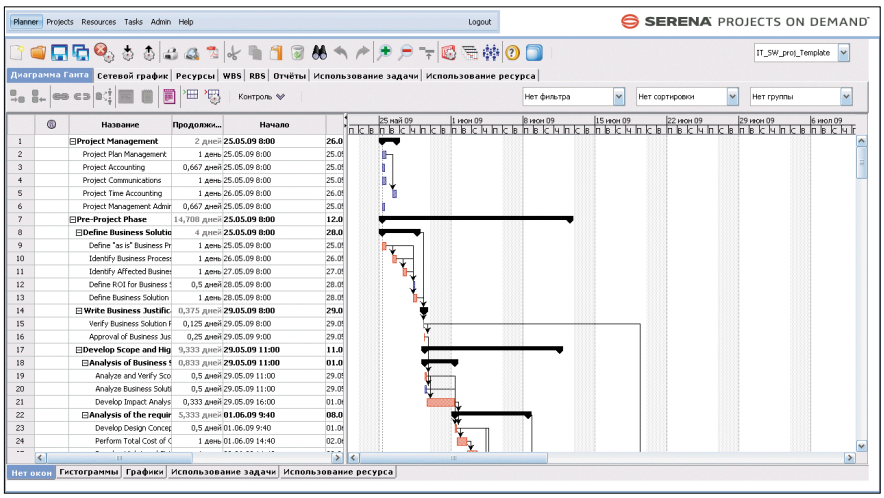
Предыдущие программы были рассчитаны на индивидуальное использование и являются аналогами MS Project Standard, далее идут более серьезные решения.

DotProject

DotProject (ранее dotmarketing.org) – очень мощное решение, предназначенное для управления проектами, написанное с применением веб-технологий [10]. Возможности достаточно велики и позволяют руководителю управлять проектами, задачами и ресурсами в нескольких компаниях. В список программы можно ввести данные о клиентах, производителях, поставщиках, консультантах и прочих участниках и ресурсах проекта. Примечательно, что адрес любого можно указать вплоть до точки на Google Maps. Меню позволяет быстро отобрать задачи и проекты, удовлетворяющие определенным условиям. Доступны диаграммы Ганта. Реализована многоуровневая схема доступа, каждый пользователь в своем рабочем пространстве также получает информацию обо всех задачах и проектах, в которых он участвует, в виде списка To Do и календаря (событий и задач). Поддерживаются форумы, обмен файлами через веб-интерфейс (программа требует, чтобы в php.ini была разрешена загрузка файлов до 32 Мб). Система заявок (тикетов) позволяет участнику отправить сообщение администрации.

Модульная архитектура дает возможность расширить функциональность, часть модулей идут в стандартной поставке, и их достаточно активировать. Остальные модули доступны по адресу [11], там же находятся пакеты локализации интер-

Рисунок 2. Projects On Demand от Serena Software



фейса, среди которых есть и русский. Альтернативный пакет, содержащий файлы русского языка для версии 2.1.x, доступен по адресу [12]. Самостоятельно перевести пункты меню можно использовав пункт Translation Management. Интерфейс легко перестроить под свои задачи, в поставке имеются три темы, которые можно использовать за основу.

Написан dotProject на PHP, информация записывается в базу данных. Рекомендованной платформой для работы является Linux или другая UNIX-подобная операционная система с Apache 1.3.27, PHP 4.2+ и MySQL. Особо обозначено, что MySQL поддерживается в полном объеме. Хотя в качестве СУБД могут быть использованы PostgreSQL, iBase, Informix, Microsoft SQL Server, Oracle, SQLite, Sybase и некоторые другие. Во время установки скрипт выводит рекомендацию использовать именно Linux:

```
dotProject is usually tested on Linux
first and will always have better
support for Linux than other operating
systems.
```

Установка DotProject проста и выполняется методом, обычным для решений, написанных на PHP. Настройки его работы и управление проектами также нельзя назвать очень сложными.

Project.net

Весьма оснащенный функционально, относительно простой в управлении, но сложный в установке продукт уровня предприятия, распространяемый под двойной лицензией [13]. Организация рабочего пространства основана на Project Workspace, которое является отображением реального проекта в виртуальной среде. Реализовано много функций – проекты, задачи, общий календарь (с поддержкой iCalendar), обмен документами с отслеживанием версий, дискуссионные группы и форум для обмена сообщениями, повторяемые процессы, материалы, контроль расходов и прочие составляющие. Проекты группируются по бизнес-единицам, составляя портфель. Каждый проект имеет связанный WiKi, фиксирующий информацию о задачах, и блог, выводящий статус. Информация в пределах Project Workspace по умолчанию доступна только ее участникам, но можно разрешить доступ и другим пользователям.

Руководитель может задать, просматривать и отслеживать задачи

Рисунок 3. OpenProj обладает хорошими возможностями

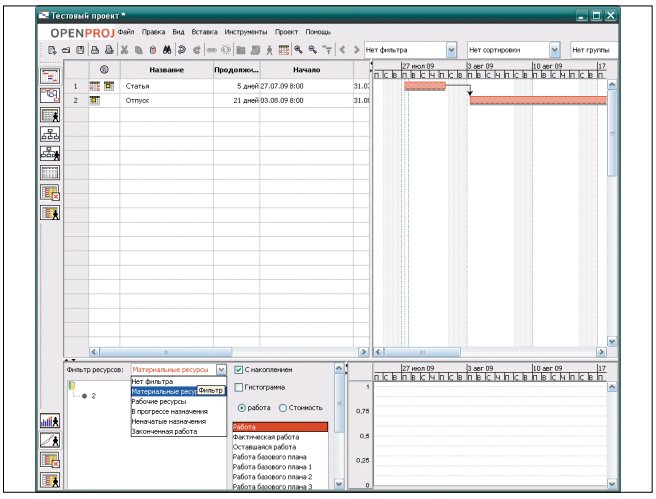


Рисунок 4. К OpenWorkbench придется привыкать

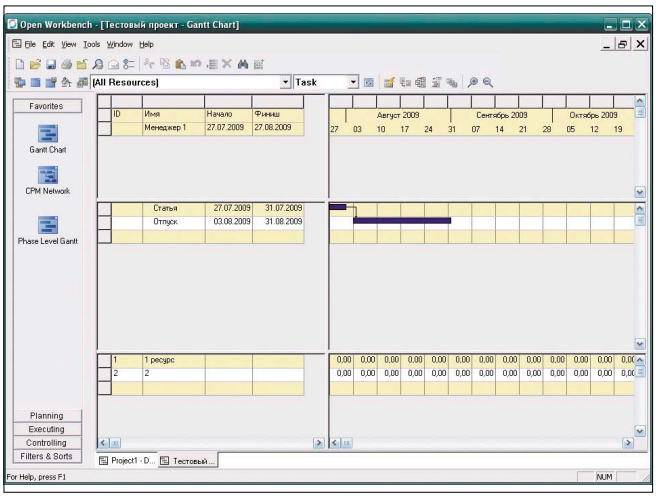


Рисунок 5. GanttProject умеет работать с XML-файлом проекта, расположенным на FTP

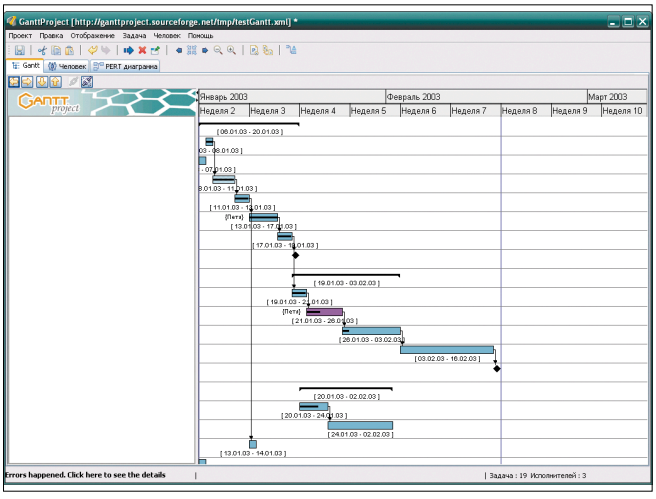
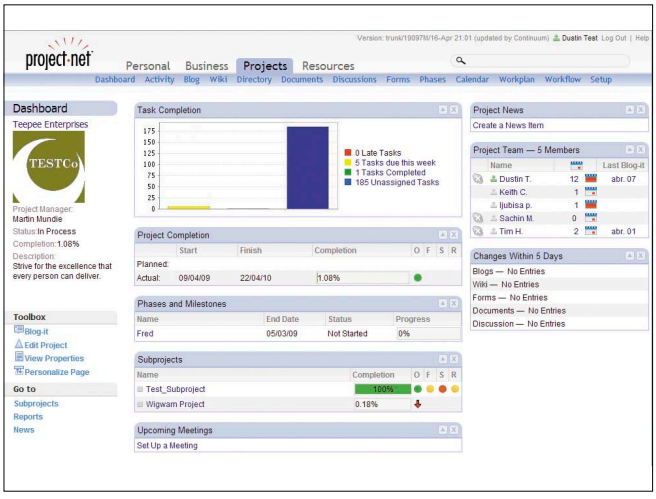


Рисунок 6. Освоить Project.net довольно просто, но вот с установкой придется повозиться



для всей команды, с контролем выполнения в реальном времени. Информация по текущим задачам выводится отдельным пользователям. Реализовано несколько видов отчетов, также нужную информацию можно получить в виде графиков задач и диаграмм Ганта. Создание проектов упрощается за счет использования шаблонов, в которых описаны все основные настройки. Успешные проекты потом можно использовать в качестве шаблона. Реализованы различного рода оповещения о статусе проекта, появлении новой задачи, в том числе по электронной почте. Все настройки производятся через веб-интерфейс. В настоящее время доступна только английская версия, но некоторые локальные данные вроде даты, валют и календаря выводятся в зависимости от установок браузера.

Возможны импорт и экспорт проектов в XML-файл, совместимый с MS Project.

В качестве сервера приложений используется Apache Tomcat или WebLogic, веб-сервер Apache или IIS, СУБД только Oracle Database. Реализован многоуровневый доступ на основе ролей, поддерживается аутентификация через LDAP/Active Directory. Хотелось бы также отметить неплохую документацию проекта.

Redmine

Redmine [14] – одно из лучших приложений для управления проектами и контроля задач, построенное с использованием веб-технологий. Поддерживается одновременно несколько проектов, каждый из которых имеет свои настройки. При создании проекта выбираются доступные модули – «Задачи», «Файлы», «Учет времени», «Документы», Wiki, «Форум» и так далее. На сайте проекта доступны еще два десятка дополнительных модулей, позволяющих рассчитать бюджет, блог, диаграммы, графики, чат и многое другое. В Standard реализована гибкая система отслеживания задач с диаграммами Ганта и календарем. Диаграммы можно экспортировать в PDF или PNG, но русские символы отображаются некорректно (необходима правка rfpdf). Все поля в таблицах задач, проектов, пользователей настраиваются, при необходимости можно убрать или добавить дополнительные поля. Внешний вид легко изменяется при помощи тем или редактированием CSS. Реализованы ленты новостей и оповещения по почте. Реализовано управление задачами через почтовые сообщения. В настройках можно указать несколько SMTP-серверов для разных групп. Пользователи могут регистрироваться самостоятельно, с автоматической активацией, активацией по почте или вручную администратором. Реализована поддержка LDAP. В разных проектах пользователь может иметь разные уровни доступа. Проекты имеют публичный статус, то есть могут быть видимы для всех или быть закрытыми. Несколько простых отчетов – по пользователям, типам задач, видам деятельности и так далее. Поддерживаются системы контроля версий – SVN, CVS, Git, Mercurial, Bazaar и Darcs. К проектам подключаются отдельные хранилища.

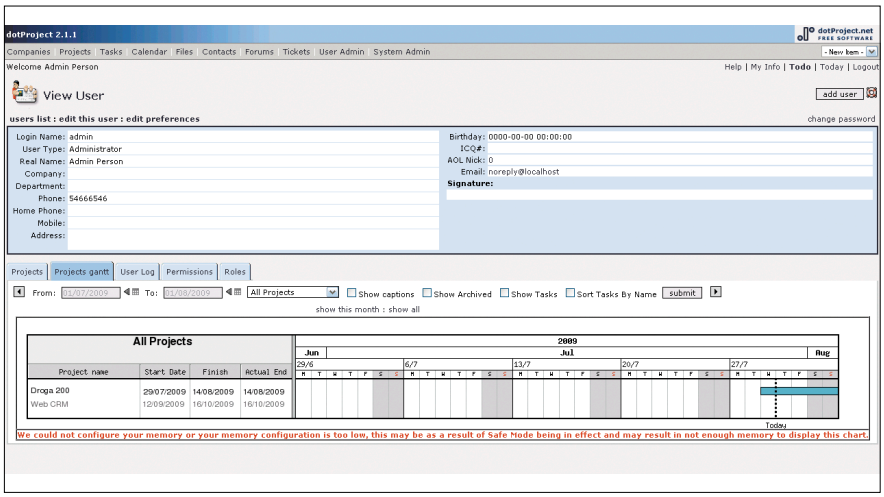
Интерфейс переведен на несколько языков, в списке есть русский. Написан Redmine на Ruby и может быть запущен на любой системе (UNIX/Linux, MacOSX, Windows), для которых доступны – Apache, Subversion, Ruby, Rails и некоторые другие опциональные компоненты. В качестве СУБД может быть использован MySQL, PostgreSQL и SQLite. Проект предлагает исходные тексты только через svn, инструкция по установке достаточно короткая, выполнив ее рекомендации в пошаговом режиме, получим готовую систему. Готовые пакеты для большинства систем (Linux, Windows, Mac и Solaris) можно найти на сайте BitNami [15], здесь же доступен образ для VMware.

Achievo

Еще один менеджер управления проектов, построенный с применением веб-технологий.

Сильные стороны Achievo – регистрация времени, разнообразные отчеты и статистики, показывающие, кто, когда и чем занимается. Всего этого в Achievo более чем достаточно. Поддерживается любое количество проектов, заданий, событий, организаций и участников. Привязка заданий к проектам, контроль над полученными заданиями, удобный календарь и планировщик. Предусмотрено создание повторяющихся событий. По проектам доступны статистики за любой период, диаграмма Ганта, напоминание по срочным делам. К потраченному времени можно добавить пояснение. Создание проектов и задач упрощают шаблоны, проекты подразделяются также и по категориям. Есть возможность обмена документами, контроля платежек, создания компаний. Информацию о пользователях можно хранить в формате vCard. Реализовано несколько уровней доступа к Achievo. Пользователь сопоставляется с несколькими категориями – Functionlevel (менеджер, директор, программист и т.п.), Department (отдел), он может быть руководителем, иметь профиль администратора или офицера безопасности. В настройках есть полезная функция – отслеживание дней рождения всех участников. Реализован экспорт отчетов в CSV-файл.

Рисунок 7. DotProject прост в установке и использовании



Интерфейс локализован, освоить работу с Achievo достаточно просто.

Написан Achievo на PHP, для его установки понадобится традиционная связка – веб-сервер с поддержкой PHP и MySQL. Возможна работа и с PostgreSQL и Oracle, но разработчики рекомендуют именно MySQL. Процесс установки стандартен для такого вида приложений.

Другие решения

Как видите, вопреки устоявшемуся мнению, бесплатные решения, способные заменить Microsoft Project, на самом деле есть, и выбор очень велик. Из тех проектов, что удалось уже попробовать, в обзор не попали:

]project-open[(http://www.project-open.org) – довольно запутанное приложение, построенное на OpenACS (Open Architecture Community System).

Streber (http://www.streber-pm.org) – простое и понятное решение, находящееся пока в стадии начальной разработки (текущая версия 0.0902), пока реализована базовая функциональность. Так, Streber не имеет на данный момент календаря, отчетов, диаграмм Ганта.

ClockingIt (http://www.clockingit.com) – интересное решение с приятным интерфейсом, поддерживающим Ajax, Drag`n`drop, поэтому управление напоминает работу с настольным приложением. К сожалению, интерфейс пока не переведен на русский язык. Особенностью ClockingIt является то, что все действия организованы вокруг задачи, проект их только группирует, выступая своего рода ярлыком. То есть все задачи, файлы и так далее находятся на одном уровне, принадлежность к проекту просто одно из свойств. Распространяется под свободной MIT/X Consortium License. Причем свой сервер устанавливать необязательно, проект предлагает использовать его площадку.

Gnome Planner (http://live.gnome.org/Planner) – один из компонентов Gnome Office, предназначенный для управления проектами, задачами, ресурсами, отслеживания ресурсов, контроля при помощи диаграмм Ганта. Довольно прост и удобен. Поддерживается импорт проектов MS Project XML, нормально открывает XML, созданный в OpenWorkbench. Результат можно экспортировать в HTML. Gnome Planner доступен в репозиториях большинства *nix-дистрибутивов. На сайте проекта кроме исходных текстов есть сборка для Windows.

***

Выбрать победителя довольно сложно. В различных ситуациях побеждают разные свойства, поэтому в любом случае вам придется подбирать решение по себе.

Мне нравятся dotProject и Redmine, которые имеют нужную функциональность и удобны в установке и использовании. Среди решений, предназначенных для персонального использования, особо хочу выделить OpenProj и GanttProject. BOF

1. Сайт Microsoft Project – http://office.microsoft.com/ru-ru/project.
2. Сайт проекта eGroupware – http://www.egroupware.org.
3. Яремчук С. Устанавливаем eGroupware. //Системный администратор, №3, 2007 г. – С. 36-41.
4. Сайт проекта Group-Office – http://www.group-office.com.
5. Сайт проекта Open-Xchange – http://open-xchange.com.
6. Сайт проекта OpenProj – http://openproj.org, http://sourceforge.net/projects/openproj.
7. Wiki проекта OpenProj – http://openproj.org/wiki.
8. Сайт проекта Open Workbench – http://www.openworkbench.org.
9. Сайты проекта GanttProject – http://ganttproject.biz, http://ganttproject.blogspot.com.
10. Сайт DotProject – http://www.dotproject.net.
11. Модули к DotProject – http://sourceforge.net/projects/dotmods.
12. Частичный перевод интерфейса DotProject 2.x – http://blog.vityasev.ru/2007/08/30/dotproject-2-russian-translation.
13. Сайт Project.net – http://www.project.net/open_source.htm.
14. Сайт проекта Redmine – http://www.redmine.org.
15. Установочные файлы Redmine – http://bitnami.org/stack/redmine.
16. Сайт проекта Achievo – http://achievo.org.

Рисунок 8. Создание проекта в Redmine

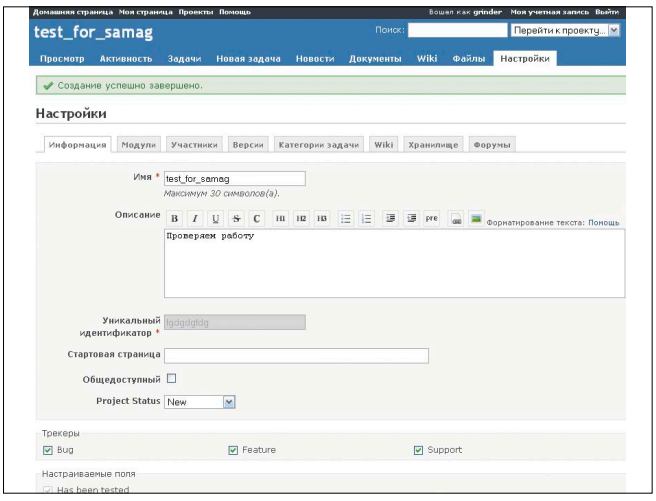


Рисунок 9. Achievo хорош для контроля времени

Визитка

РАШИД АЧИЛОВ, поклонник FreeBSD с 14-ти летним опытом использования ее в совмещенных с Windows сетях и сторонник Open Source. Администратор сетей и средств защиты крупной торговой сети

Заменяем сервер MS Exchange

Установка Horde Groupware

Вместо сервера Microsoft Exchange устанавливаем набор бесплатных программ без потери какой-либо функциональности, при этом Groupware-компонент заменяем на Horde Groupware

Программа, не имеющая аналогов

Именно так с давних пор именуют Microsoft Exchange как его сторонники, так и противники на различных интернет-форумах. Запрос к любой поисковой системе «замена exchange» (точная фраза), как правило, вызывает отображение не менее 500 ссылок. К сожалению, за всем этим энтузиазмом просматривается одна картина – заменять предлагают либо на Communigate Pro [1] (вещь хорошая, но тоже далеко не бесплатная), либо на Zimbra [2] (который настолько привязан к Linux, что инструкция по его сборке на FreeBSD читается как какое-то страшное колдовство [3]). Предлагаются также другие программы разной степени доработанности, собственные решения... Среди них – eGroupware [4], moreGroupware [5] и Horde Groupware Webmail Edition [6]. Правда, зачастую ко всем этим решениям следует приписка: «Извините, в этом случае у вас не будет работать то-то, а в этом то-то...».

Вкратце я напомню, что входит в состав Exchange-сервера и почему его считают не имеющим аналогов:

- SMTP-сервер, обеспечивающий почтовый обмен с внешней средой;
- POP3/IMAP/MAPI-сервер, обеспечивающий почтовый обмен с внутренними клиентами;
- Groupware-сервер, обеспечивающий совместную работу с календарем, задачами и заметками, а также их синхронизацию с Microsoft Outlook;
- веб-сервер, предоставляющий доступ к почтовому ящику Exchange из веб-браузера (Outlook Web Access, OWA);
- интеграция в Active Directory, позволяющая брать информацию о почтовых адресах и контактную информацию непосредственно из AD.

Неудивительно, что во времена контрафактного программного обеспечения Microsoft Exchange был столь популярен – одна программа действительно решала все (ну или почти все) коммуникационные проблемы, в то время как для реализации подобного функционала на свободном программном обеспечении необходимо не менее 5 компонентов, поэтому рассматривать замену Exchange мы будем постепенно. На первом этапе функции OWA и groupware-сервера возьмет на себя какой-либо Open Source-проект. На втором этапе функции почтового сервера, а также предоставление доступа к общим папкам возьмут на себя соответственно sendmail и dovecot.

В качестве замены OWA и Groupware-сервера рассматривались уже упомянутые eGroupware, moreGroupware и Horde Applications Framework. Выбор остановился на Horde. Несмотря на то что eGroupware кажется перспективной разработкой и процесс её установки более прост для пользователя, не обладающего достаточной квалификацией, Horde построен более логично и реализован более понятно, что позволяет квалифицированному пользователю в короткий срок доработать его под имеющиеся требования. Ну и не в последнюю очередь потому что настраивается он исключительно в стиле UNIX-way – все параметры находятся в текстовых конфигурационных файлах и достаточно прозрачны, хоть и мало документированы.

Horde Groupware Webmail Edition (groupware-разработка Horde Applications Framework) содержит в себе три компонента Exchange – groupware-сервер (календарь, управление задачами и заметками), обеспечение доступа к почте через браузер и адресная книга с интеграцией в Active Directory.

Но решающим фактором как обычно становится цена. А иногда и не только цена...

С цветным дисплеем Nokia

Скажу честно – изначальная задача не несла в себе никаких элементов замены Exchange. Изначальная задача формировалась просто и скромно – предоставить доступ к корпоративной почте с мобильного компьютера (как их горделиво называет Nokia) модели Nokia N97. В экспериментах принимали участие также Nokia N95 8G и дешевенький коммуникатор Mitac MIO серии DigiWalker с Windows Mobile 5.x.

Вариант «мобильный VPN + Mail for Exchange» был забракован из-за отсутствия поддержки PPTP, из-за сложности и неочевидности настроек, из-за необходимости много времени затратить на эксперименты. Да и покупать аппаратный шлюз как-то не хотелось.

Но решающим фактором как обычно становится цена. А иногда и не только цена

Единственный оставшийся вариант – вывести наружу OWA, запущенный на внутреннем сервере Exchange через нестандартный порт, усилив для безопасности протоколом SSL. Тогда еще никто не предполагал, что эта скромная задача окажется первой, что стала не по зубам Exchange.

Подробности настройки NAT я опущу. Скажу только, что OWA на порту, допустим 11222, заработал как надо, и на экране коммуникатора после обязательного запроса имени и пароля появилось содержимое почтового ящика.

Счастливые и довольные, мы набрали тот же самый адрес на Nokia N97. И долго с непонимающим видом смотрели на надпись «Internet: невозможно подключиться к защищенному соединению». При чем тут защищенное соединение, было совершенно непонятно, потому что не то что https не указывался – SSL еще и поднят не был!

После того как и Nokia N95 выдала точно такое же сообщение, была установлена пробная версия браузера Opera Mini. Opera оказалась более общительной, и тут нас осенило – диалог авторизации! Браузер не может выдать диалог авторизации, потому что авторизация в OWA выполняется не через форму, а используется системный диалог!

Ну и последним ударом по решению о доступе через OWA стал звонок в техническую поддержку Nokia, в которой разъяснили, что у нас ничего и не могло получиться из попытки зайти на OWA и для этого как раз и предназначен Mail for Exchange. Круг замкнулся. Я ничего не хочу сказать про отношения Nokia и Microsoft, но почему-то он замкнулся таким образом, что OWA остался с одной стороны круга, а мы со своей задачей – с другой.

Добро пожаловать в Портал

Что ж, Nokia показала себя далеко не с лучшей стороны, и немного повздыхав, мы модифицировали задачу до «каким-нибудь образом предоставить доступ извне к корпоративному почтовому ящику». Ну а где корпоративный почтовый ящик, там неизбежно возникают календарь, планировщик задач и встреч, заметки. Обратиться к Open Source groupware-проектам мне показалось вполне естественно.

Не буду подробно расписывать процесс тестирования проектов, которыми я попробовал воспользоваться. Скажу только, что у eGroupware есть определенный потенциал, если, конечно, разработчики поймут, что адресная книга в Groupware, особенно когда авторизация делается через Actve Directory, как правило, в ней же и хранится. Пока что я там не обнаружил никаких средств, позволяющих указать, где брать адреса и как их представлять – только собственная адресная книга. moreGroupware имел точно такой же недостаток и к тому же несколько более примитивный интерфейс, что и привело меня в конце концов к Horde Applications Framework Project.

Вообще говоря Horde содержит значительно больше, чем нам нужно. Здесь есть и менеджер фотоальбомов, и багтрекер, и файловый менеджер, и менеджер закладок... Поэтому я поискал Horde в портах, и, конечно же, нашел. Несколько неприятным сюрпризом было то, что порт назывался не horde, как того следовало бы ожидать по аналогии с многими другими (X11, KDE), а почему-то horde-meta:

```
# cd /usr/ports/www/horde-meta
# make
```

Как и положено хорошему мета-порту, порт horde содержит экран конфигурации, на котором можно выбрать модули, которые следует установить (см. рис. 1).

Нам нужны далеко не все модули. Некоторые, конечно же, могут оказаться полезными, но понадобятся нам:

IMP – модуль почтового клиента, предоставляющего доступ к почтовым ящикам IMAP/POP3.

MIMP – мобильная версия IMP, оптимизированная для работы на телефонах и коммуникаторах. Предоставляет доступ только к почтовому ящику (то есть, если браузер распознается как мобильный – не будет ни глобальной адресной книги, ни календаря-планировщика. Впрочем, как правило в мобильных устройствах есть свой собственный).

Ingo – модуль фильтров для обработки почты и отсеивания нежелательной корреспонденции.

Kronolith – модуль календаря и планировщика дня.

Turba – модуль адресной книги.
Nag – модуль списка задач.
Mnemo – модуль управления заметками.

Отмечу, что в составе портала имеется сервер синхронизации мобильных устройств (смартфоны, коммуникаторы) и программы Microsoft Outlook с помощью протокола SyncML версии 1.1 или 1.2.

Сразу же несколько огорчает тот факт, что связной документации на horde практически нет, ни на английском, ни тем более на русском. Имеется только Wiki по адресу [7], где сделана не слишком удачная попытка сгруппировать все имеющиеся знания. Правда, файлы conf.php имеют собственные секции комментариев, некоторые вещи очевидны, про некоторые другие можно прочитать в списках рассылки. Ситуация обычная – для установки портала желательно быть программистом, имеющим опыт работы на PHP. Что ж, начали.

Глобальных запросов у horde немного – веб-сервер,база данных (мы будем использовать MySQL, хотя возможно использование PostgreSQL и пяти других СУБД, в том числе и MS SQL. Для более подробной информации смотрите scripts/README) и PHP с установленными необходимыми модулями. Вот в этом «необходимыми модулями» и кроется подвох – здесь будут и OpenLDAP, и mbstring, и еще многоемногое другое. Более подробный список можно прочитать в каталоге порта, в файле docs/INSTALL.

Перед началом установки horde порт устанавливает количество всевозможных PEAR-модулей, которое иным словом, кроме как «немыслимо огромное», не назовешь. Он их ставит и ставит, ставит и ставит, и все равно нет никакой гарантии, что при тестировании установки не потребуется что-то еще.

Решаем, что разместим портал на порту 18511 (Почему? 18511 -> 0x484F -> «HO» в текстовом виде), адрес портала во время тестирования был http://horde.shelton.net.

Установка, как правило, проходит без ошибок – им пока браться неоткуда. Устанавливается все в каталог /usr/local/www/horde. Настоятельно рекомендую до тех пор, пока не решили, что добились от установленного портала всего, что хотели, этот каталог не трогать, а копировать его по мере необходимости в другие каталоги. Например, я организовал виртуальный хост:

```
Listen 18511
<VirtualHost 192.168.1.1:18511>
    ServerName horde.shelton.net
    ServerAdmin webmaster@shelton.net
    DocumentRoot "/usr/local/www/vhosts/horde/"
    ErrorLog "/usr/local/www/log/horde/httpd"
    CustomLog "/usr/local/www/log/horde/access" common
    <IfModule php5_module>
        AddType application/x-httpd-php .php
        AddType application/x-httpd-php-source .phps
    </IfModule>
    Include etc/apache22/extra/httpd-languages.conf
    <IfModule mime_module>
        AddType application/x-tar .tgz
        AddEncoding x-compress .Z
        AddEncoding x-gzip .gz .tgz
        AddHandler cgi-script .cgi
    </IfModule>
    <Directory "/usr/local/www/vhosts/horde">
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Все пути к файлам будут указываться от приведенного выше значения параметра DocumentRoot.

Поскольку весь портал целиком и любой его модуль в частности настраиваются простым редактированием файлов conf.php, то никакой защиты от неверных настроек (когда ты в одном месте задал нужную настройку, а в другом позабыл) нет. Мне в процессе настройки приходилось не менее десятка раз восстанавливать промежуточные копии conf.php.

Начинаем выполнять то, что сказано в файле docs/INSTALL. Первым шагом нам рекомендуют создать БД, которую будет использовать портал:

```
cd ../scripts/sql
mysql -u root -p < create.mysql.sql
```

Предварительно просматриваем скрипт и увидев, что он создаст пользователя MySQL horde с паролем horde и БД horde, а также даст этому пользователю все права на БД, запускаем этот скрипт.

Скрипт отрабатывает без ошибок. Идем просматривать структуру созданной БД и...что такое? Откуда тут взялся Latin-1, когда предполагается вовсю использовать русский язык?

Рисунок 1. Экран конфигурации порта horde-meta

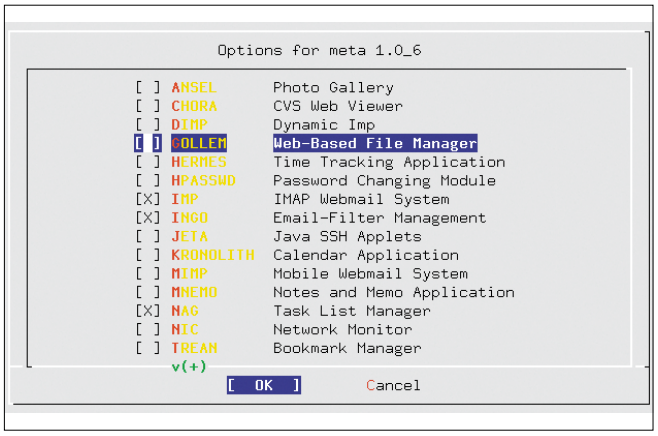
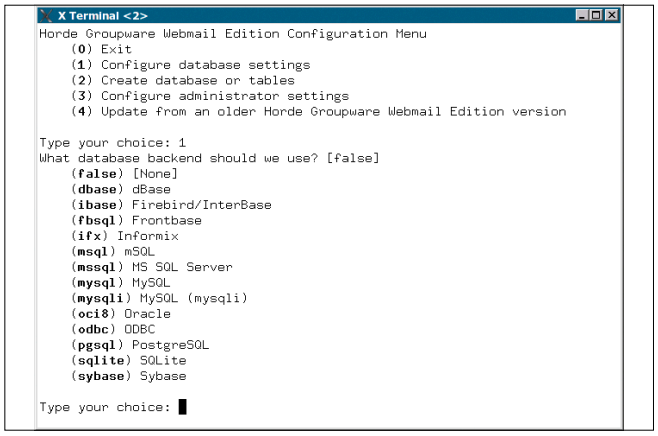


Рисунок 2. Экран настройки Horde Groupware Webmail Edition



```
mysql> show create database 'horde';
```

```
horde   CREATE DATABASE `horde` /*!40100 DEFAULT CHARACTER
SET latin-1*/
```

В поисках ответа на этот вопрос идем на [6] и обнаруживаем, что наш порт, из которого мы ставили портал, устарел – уже существует более новая версия базового пакета портала horde-3.3.4. Новая версия вышла 1 мая 2009 года, но порт до сих пор почему-то ссылается на старую. Что ж, скачиваем вручную и распаковываем поверх в корневой каталог.

Можно еще скачать с главной страницы [6] архив Horde Groupware Webmail Edition и полностью заменить им старый каталог. Не забываем удалить созданную БД. А вот пользователя лучше оставить.

Отличия заметны сразу – теперь начальная установка управляется собственным скриптом (см. рис. 2)!

Все пункты меню подписаны вполне самодостаточно. Предупреждаю сразу, не используйте драйвер СУБД mysql (mysqli) – по неясным причинам он вызывает зависание портала сразу после включения – портал загружает окно входа в систему и никак не может его загрузить, используйте обычный mysql-драйвер.

Задаем настройки БД (вот тут нам и пригодится пользователь, которого мы не удалили, почему-то скрипт этого не делает), создаем необходимые таблицы. Пояснений требует только пункт 3 – Configure administrator settings. Дело в том, что непонятно зачем разработчики портала сменили начальный метод авторизации – в версии 3.3.3 сразу после установки происходила автоматическая регистрация в системе под учетной записью Administrator, в версии же 3.3.4 зачем-то придумали авторизацию по почтовому серверу – docs/INSTALL рекомендует исправить файл imp/config/servers.php, указав в нем адрес сервера, если он установлен не на том же компьютере, что и портал. А в данном пункте требуется указать учетные записи, при входе под которыми будут предоставляться права администратора.

Мне это показалось до крайности неудобным – ведь еще ничего не настроено! – поэтому я предпочел вернуть обратно автоматическую авторизацию, по крайней мере на время. Для этого нужно в файле config/conf.php (основном конфигурационном файле horde) исправить следующие строки:

```
$conf['auth']['admins'] = array('Administrator');
$conf['auth']['driver'] = 'auto';
$conf['auth']['params'] = array('username' => 'Administrator');
```

Всё, действий в консоли больше предпринимать не придется, разве только какой-нибудь отсутствующий модуль доустановить.

Перед тем как начать настраивать портал, необходимо проверить как раз полноту установки всех требуемых модулей. Для этого заходим на страницу http://horde.shelton.net:18511/test.php и внимательно читаем все, что там написано. Сообщения, написанные желтым цветом, – это замечания, их нужно прочитать внимательно. Те, что не относятся к нашей системе (например, про отсутствие поддержки PostgreSQL), можно игнорировать, те же, что относятся, лучше исправить. Сообщения же, написанные красным цветом, – это ошибки, без исправления которых портал будет неработоспособен полностью или частично. Как правило, это сообщения об отсутствии какого-либо существенного компонента или о неверных настройках PHP.

После исправления всех ошибок и тех замечаний, которые необходимо было исправить, заходим на http://horde.shelton.net:18511 (см. рис. 3).

Автоматически нас регистрируют в системе как Administrator. Почтовый ящик сейчас закрыт – для входа в него требуется, чтобы данный пользователь существовал на почтовом сервере, которым пока еще является Exchange. Впрочем, почта нас и не интересует, нам нужны только настройки системы.

Раскрываем пункт Administration и выбираем Setup. Это основное место для настройки параметров модулей horde. Сейчас, когда настроены только некоторые основные параметры портала, напротив всех остальных пунктов стоит missing configuration, что означает, что файлы conf.php для данных модулей еще не созданы. Но сначала мы займемся настройками самого портала.

Заходим в настройки модуля horde. Нас встречает экран с таким количеством настроек, что только диву даешься – и это все нужно? Нет, конечно же, не все. Хотя не менее двух третей закладок придется посетить (см. рис. 4).

Закладки настроек расположены примерно в том порядке, в котором их необходимо настраивать. Только закладку Authentication советую настраивать самой последней – как только метод авторизации меняется на что-либо другое, кроме auto, так тут же будет показан экран регистрации в системе. А процесс настроек – вещь длинная. После настройки каждой закладки рекомендую сохранять конфигурацию. Ниже я опишу некоторые настройки, которые изменял. Если какая-то закладка здесь отсутствует – значит в ней менять ничего не надо.

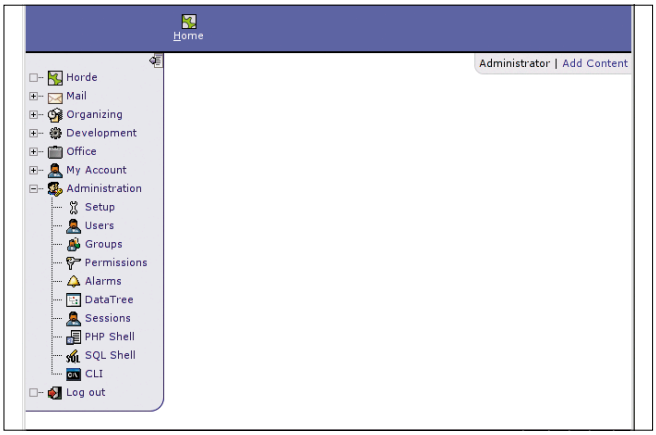
Закладка General

Здесь настраиваются разнообразные общие параметры:

[tmpdir] – можно настроить каталог для временных файлов, если /tmp чем-то не устраивает;

[server][port] – в этой настройке обязательно задавать номер порта, если horde работает не на стандартном порту веб-сервера;

Рисунок 3. Главный экран портала непосредственно после установки



[cookie][path] – эта настройка обязательно должна совпадать с путем, где установлен портал. Если портал является корнем виртуального хоста, она должна быть равна «/».

Закладка Database

Параметры на данной закладке зависят от выбранного типа СУБД. Для MySQL они, как правило, стандартны – имя БД, имя и пароль пользователя для подключения, адрес и порт для подключения. Еще раз напоминаю – выбирайте драйвер mysql, а не mysql(mysqli)!

Закладка Logging

Незаменима при отладке:

[log][enabled] – включить/выключить запись журнала;

[log][priority] – задает уровень детальности сообщений от критических ошибок до сообщений отладки;

[log][type] – задает, куда будут направлены сообщения. От вариантов выбора глаза разбегаются – тут присутствует все что угодно, от БД до окна на экране. Мне, правда, наиболее оптимальным показалось банально записывать в файл.

Остальные настройки закладки вполне очевидны, тем более, что каждая настройка снабжена комментарием, может, не слишком подробным, но исчерпывающим.

Закладка Preference System

Здесь расположены параметры настройки системы записи индивидуальных настроек пользователей. Если оставить все как есть, настройки сохраняться не будут. Впрочем, единственный параметр, который нужно изменить, – это [prefs][driver]. По умолчанию он установлен в PHP Session, его нужно обязательно изменить на SQL Database (ну или другое значение, представленное в списке).

Закладка Alarm System

[alarms][driver] – указать, как будут храниться аварийные оповещения. Вариантов, правда, всего два – не хранить совсем или хранить в БД.

Закладка DataTree System

DataTree – это некая древовидная структура данных, используемая порталом для хранения другой информации вместо, например, БД. Мне ее применение кажется сомнительным, но раз она есть – значит зачем-то нужна.

[datatree][driver] – задает, где будет храниться ее информация, точнее говоря, будет она храниться или нет, потому что вариантов, как и на предыдущей закладке, всего два.

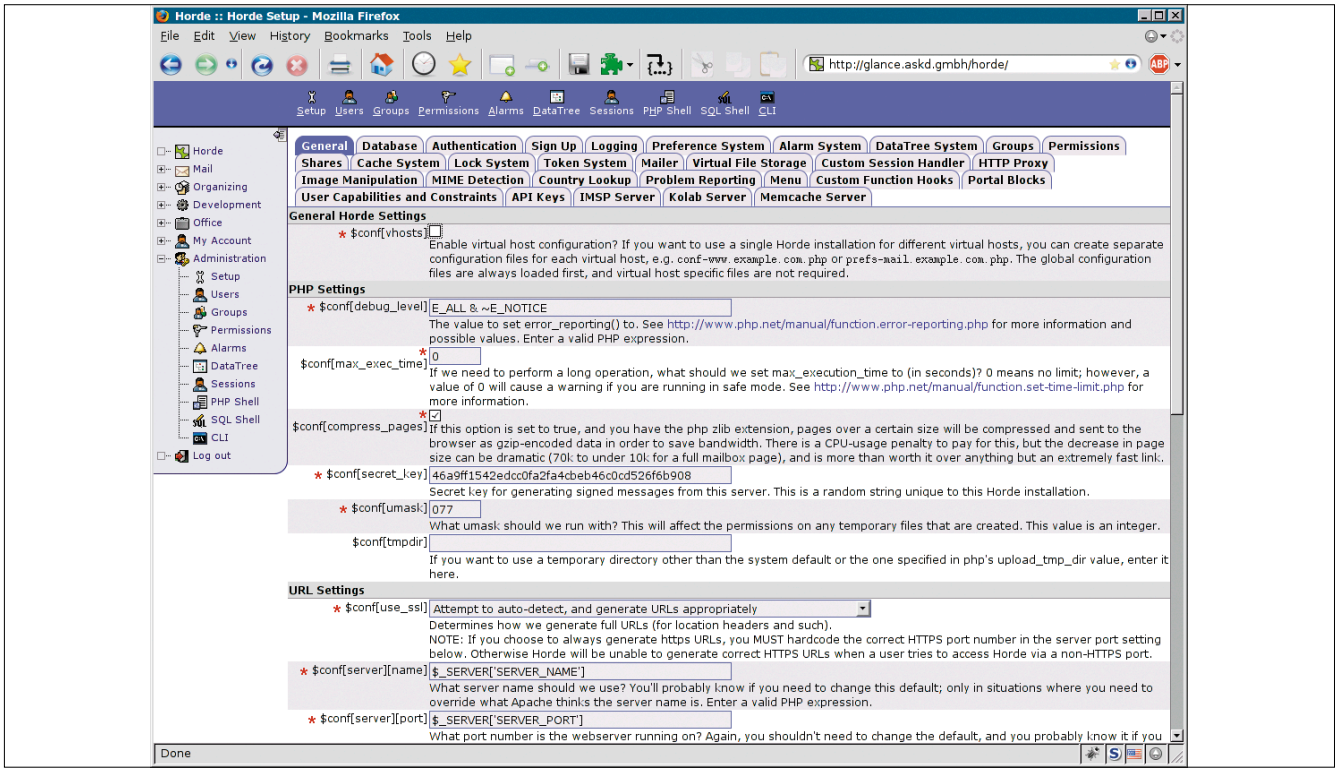
Закладка Groups

[group][driver] – задает, где будут храниться группы, созданные в портале. Это именно внутренние группы портала, создаваемые через подпункт «Группы» в меню «Управление».

Закладка Permissions

[perms][driver] – задает, где будут храниться созданные в портале права доступа, настраиваемые через подпункт «Права» в меню «Управление». Права, которые необходимо минимально назначить для того, чтобы портал работал, будут описаны в следующей части статьи.

Рисунок 4. Настройки портала



Закладка Shares

Здесь задаются настройки для обьектов, доступ к которым может регулироваться пользователями, например для календарей:

[share][any_group] – если отмечено, то пользователи могут предоставлять доступ к своим данным любому другому пользователю, в противном случае – только тем, кто входит с ними в одну группу;

[share][driver] – задает, где будут храниться данные о разделяемых ресурсах.

Закладка Lock System

[lock][driver] – задает использование системы блокировки обьектов.

Закладка Mailer

Задает параметры для отправки почты. Для отправки через сторонний сервер нужно задать [mailer][params][host]; [mailer][params][localhost] – имя домена для формирования адреса для отправки через Sendmail – [mailer][params][sendmail_path].

Закладка Virtual File Storage

Настраивается расположение виртуального хранилища файлов. Используется только если будет в дальнейшем устанавливаться Gollem – файловый менеджер.

[vfs][type] – задает драйвер VFS. В простейшем случае – хранить в файловой системе;

[vsf][params][vfsroot] – задает путь к корню VFS, в случае хранения в файловой системе.

Закладка HTTP Proxy

Задает набор достаточно очевидных настроек для выхода в Интернет в случае использования прокси-сервера.

Закладка MIME Detection

Единственная настройка задает путь к базе данных MIME magic. Для FreeBSD обычно это /usr/share/misc/magic.

Сохраняем конфигурацию портала. Да, мы еще не были на закладке Authentication. Не торопитесь туда. Если включить авторизацию, не создав конфигураций модулей, есть риск сразу же после перезагрузки увидеть вместо портала два окна с сообщениями об ошибках PHP, поскольку horde совершенно не проверяет создаваемую конфигурацию.

Заходим в каждый из модулей по очереди и нажимаем кнопку «Сохранить». Менять там ничего не надо, да и настроек немного – у каждого модуля по одной закладке, за исключением Imp, у которого на первой закладке можно при желании настроить пути к программам aspell и gpg.

Сохраняем последние изменения в настройке, делаем резервную копию файла config/conf.php и настраиваем авторизацию. Авторизация в horde может быть настроена огромным количеством путей, но нас интересует только один – авторизация посредством сервера Active Directory, поэтому будет описываться именно этот набор параметров.

Закладка Authentication

[auth][admins] – задает список учетных записей через запятую, у которых будут права администратора. Эти учетные записи должны существовать в Active Directory;

[auth][driver] – задает механизм авторизации. Для использования Active Directory следует выбрать LDAP authentication;

[auth][params][hostspec] – задает адрес сервера авторизации;

[auth][params][basedn] – задает корневое DN сервера, которое обычно равно его имени домена. Например, для домена shelton.net оно будет равно dc=shelton, dc=net;

[auth][params][binddn] – задает имя пользователя, от которого будет выполняться подключение к серверу LDAP. Поскольку Windows 2003 не поддерживает анонимного подключения, необходимо наличие некоторого пользователя, который не может пользоваться никакими ресурсами и подключаться от его имени. В качестве значения параметра нужно указать полное CN-имя. Например, для пользователя ldapread, находящегося в OU Other Users, значение параметра будет равно CN=ldapread,OU=Other Users,DC=shelton,DC=net;

[auth][params][password] – задает пароль пользователя для подключения к LDAP;

[auth][params][version] – проверьте, что стоит LDAP v3, иначе не будет работать;

[auth][params][scope] – задает область поиска, должно быть задано Subtree search;

[auth][params][ad] – задает, что сервер, который выполняет авторизацию, является сервером Active Directory;

[auth][params][uid] – задает имя атрибута, содержащего идентификатор пользователя. В этом поле необходимо задать «samaccountname»;

[auth][params][encryption] – задает тип шифрования пароля при его проверке. Указать plain или msad;

[auth][params][filter_type] – задает тип фильтра, используемый для отбора учетных записей при проверке авторизации. Выбрать A complete LDAP filter expresson;

[auth][params][filter] – задает фильтр для отбора учетных записей при проверке авторизации. Фильтр нужен для ускорения авторизации, чтобы не использовались учетные записи групп, системных объектов и т.д. Простейший фильтр выглядит так – «(&(sAMAccountName=*)(mail=*))», что означает «отобрать все объекты, у которых заданы поля sAMAccountName и mail.

***

Итак, портал установлен и уже способен проверить учетную запись и пароль по серверу Active Directory. Но проделана еще только малая часть работы, главное – доступ к почте и глобальной адресной книге, синхронизация с Outlook нас ждет впереди. EOF

1. Сайт программы Communigate Pro – http://www.communigate.com.
2. Сайт программы Zimbra Collaboration Suite – http://www.zimbra.com.
3. Статья о том, как собрать Zimbra на FreeBSD – http://pcbsd.org/~dwhite/zimbra.
4. Сайт программы eGroupware – http://www.egroupware.ru.
5. Сайт программы moreGroupware – http://www.moregroupware.de.
6. Сайт Horde Applications Framework – http://www.horde.org.
7. Wiki по Horde Applications Framework – http://wiki.horde.org.

Визитка

АНТОН БОРИСОВ, специализируется на экспертизе аппаратных и программных решений

Sun Secure Global Desktop

Все ваши приложения в окне браузера

Как предоставить сотруднику, часто бывающему в командировках, унифицированный доступ к приложениям ИТ-инфраструктуры?

Вся ИТ-инфраструктура предприятия постепенно перемещается в виртуализированные контейнеры и окружение. С одной стороны, таким образом снижаются расходы на поддержание работоспособности рабочего места, поэтому сотрудник может работать вне зависимости, где конкретно он сейчас находится – дома, в офисе, в командировке. С другой стороны, необходимо предоставить такой унифицированный доступ к приложениям, чтобы можно было приступить к работе из любой операционной системы. Что это за инструмент? Безусловно, веб-браузер.

История развития

Попытки создавать интеграционное ПО для работы в гетерогенных сетях уходят корнями в далекие 80-е. Когда становится понятным, что мир меняется в сторону персональных ЭВМ и появляются операционные системы с разнонаправленной идеологией – Windows, UNIX. Понятно, что технологически легче визуально предоставить доступ к приложениям работающим, например, на UNIX-мейнфреймах, нежели переписывать эти приложения с нуля, для одной лишь цели – запустить их в другой среде. Как пример – Windows. Впрочем, возможны были ситуации и обратного характера – клиентам из UNIX-мира требовалось иногда работать и с Windows-приложениями, как впрочем и с другим ПО под другой UNIX-мейнфрейм. Визуальный доступ посредством X11-протокола позволил бы без всяких ухищрений обойти технологические трудности. Чем, в частности, и занимались компании IXI Limited и Visionware, пока их не купила Santa Cruz Operation в первой половине 90-х.

Покупка в конечном итоге вылилась в появление в 1996 году проекта Tarantella, главный девиз которого был «Любое приложение от любого клиента с любого места». Такой унифицированный подход позволял запустить любое приложение с любого устройства, на котором установлен веб-браузер с Java-плагином. В некоторой степени проект Tarantella являлся конкурирующим продуктом в пику решений от Citrix – Citrix Metaframe/Presentation Server.

Компания Santa Cruz Operation понимала важность этого проекта и поэтому, по-видимому, позиционировала Tarantella как отдельный бренд, который никак не ассоциируется с темной славой SCO.

В силу определенных причин в 2005 году этот бизнес перешел под крыло Sun Microsystems, поэтому теперь он звучит как Sun Secure Global Desktop Software.

Аппаратные требования

Что из себя представляет Secure Global Desktop? В первую очередь это веб-сервер Apache, контейнер сервлет-приложений Tomcat, а также сервер приложений, связывающий логику работы первых двух с управляющим внутренним функционалом Tarantella. Фактически SGD преобразует видеопредставление, идущее от браузера, в представление, понятное для конкретного сервера – X11, SSH, RDP, ICA. Серверная часть может работать на следующих операционных системах и платформах:

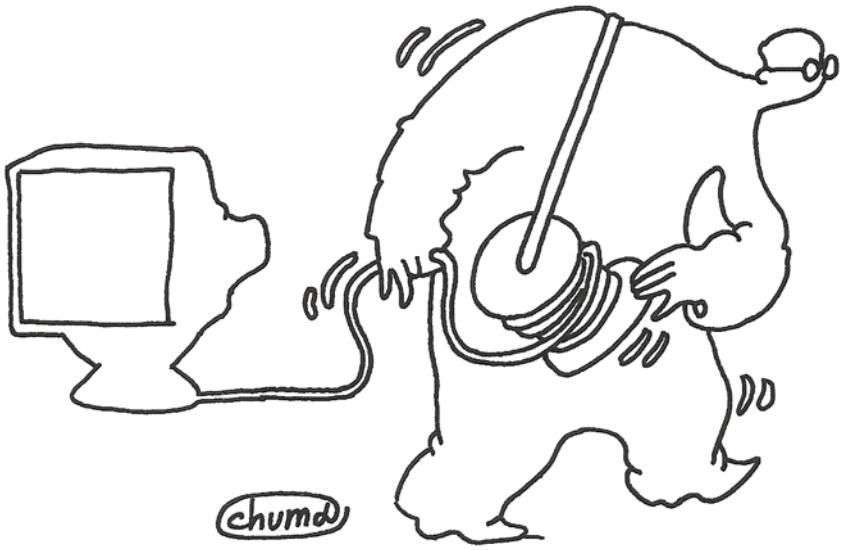
- Sun Solaris 8, 9, 10 (SPARC);
- Sun Solaris 10 (x86);
- SUSE LINUX Enterprise Server (Intel x86) 9/10;
- Red Hat Enterprise Linux (Intel x86) 4.0/5.0;
- Fedora Linux Core 8.

Минимальные требования, предъявляемые к серверу: 256 Мб ОЗУ, свободное дисковое пространство не менее 500 Мб, тактовая частота не ниже 1 ГГц. Для работы клиентов следует исходить из правила: каждое новое подключение требует 20 Мб памяти и в среднем 15 МГц тактовой частоты процессора.

Сервером SGD поддерживаются следующие протоколы передачи данных: Microsoft RDP, X11, http/https, ssh, Telnet VT, ANSI tn3270/tn5250. Авторизоваться пользователи могут по протоколам: LDAP v3, Microsoft Active Directory, RSA SecurID, NIS, Microsoft Windows Domains и некоторым другим.

Установка

В качестве серверной операционной системы я выбрал CentOS 5.3 (бесплатный аналог RedHat Enterprise Linux 5). И теперь переходим к загрузке пробной, 30-дневной версии SGD. Загружаем со страницы [1] RPM-пакет для RHEL5 (x86). Начинаем установку.




```
# rpm -ih tta-4.50-907.i386.rpm
```

Получаем ошибку – необходимые пользователи ttaserv и ttasys в системе не зарегистрированы:

```
ERROR: Required users (ttaserv and ttasys) are not
correctly defined.
Setup will now exit.

You must create two user accounts before you can install
Secure Global Desktop.

  - The user names must be "ttaserv" and "ttasys".
  - Both must have their primary group set to "ttaserv".
  - You can use any UIDs and GID you want.
  - Both users must have a valid shell, for example /bin/sh.
  - Both users must have writeable home directories.
  - We recommend that you lock the user accounts (passwd -l).
```

Можно создать их вручную либо запустить скрипт, идущий в этой поставке SGD:

```
# /tmp/SGDCreateUsers.sh
```

```
Locking password for user ttasys.
passwd: Success
Locking password for user ttaserv.
passwd: Success
```

В прошлой версии администраторы были лишены такой поблажки.

Еще раз запустим установку:

```
# rpm -ih tta-4.50-907.i386.rpm
```

```
########################################### [100%]
########################################### [100%]

To complete the installation, please run /opt/tarantella/
bin/tarantella start
```

Предварительный этап пройден. Теперь запускаем сервер и приготовимся сконфигурировать Apache/Tomcat (веб-сервер и контейнер сервлетов).

```
# /opt/tarantella/bin/tarantella start
```

Рисунок 1. Задача SGD-сервера заключается в транслировании протокола клиента в протокол сервера приложений

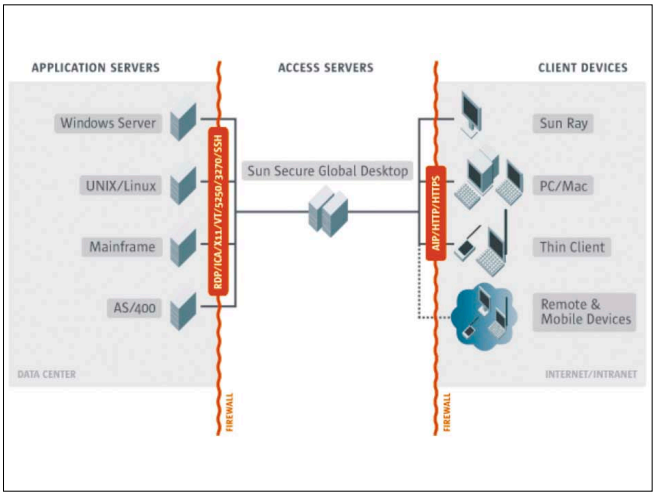
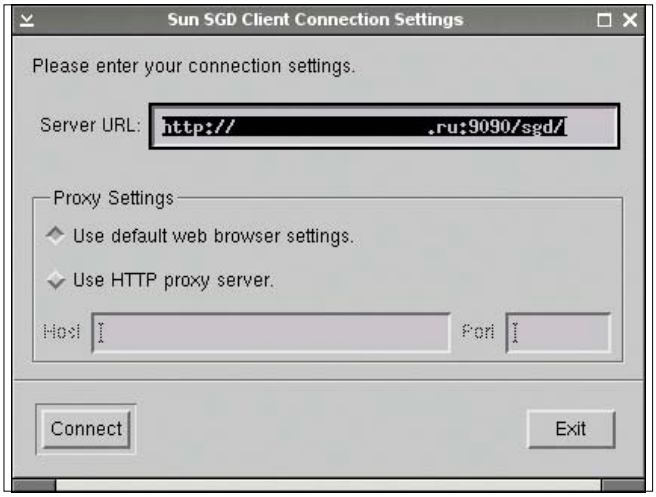


Рисунок 2. Вместо JRE можно поставить клиент SGD с тем же функционалом



В ходе процесса конфигурирования необходимо будет выбрать порт, на котором будет работать веб-сервер, выбрать DNS-имя этого сервера и время ротирования логов. В общем и целом достаточно тривиально. В любой последующий момент вы можете сами поменять настройки, благо все компоненты – как сервер Apache, так и контейнер сервлетов Tomcat – широко распространены и хорошо задокументированы.

```
Secure Global Desktop is now installed and ready to use.

To get started:

 - In a web browser, go to:
   http://sgd.intra.net:9090/
 - When prompted, log in with username "Administrator" and
   root's password.
 - On your webtop, click Administration Guide to learn more
   about the product.

To add license keys, type:
/opt/tarantella/bin/tarantella license add <key>...
Note: Setup has created some application objects.
Setup detected that telnet is not enabled on your system
but SSH is, and SSH appears to be configured to work with
Secure Global Desktop.
Consequently the application objects have been configured
to use SSH.

For more information on Secure Global Desktop and SSH,
see the Administration Guide.
-------------------------------------------------------------
Successfully installed Sun Secure Global Desktop Software
-------------------------------------------------------------
```

Настройка приложений

Теперь необходимо настроить приложения, которые пользователи будут запускать со своих клиентских мест.

Я решил не возиться со сложной авторизацией и просто создал новые учетные записи пользователей, которые будут обслуживаться в CentOS.

Чтобы добавить новые приложения, например, OpenOffice, нам потребуется зайти под администраторской учетной записью и произвести необходимые изменения. Для полноценной работы с SGD необходимо, чтобы на пользовательских местах была установлена JRE (Java Runtime Environment). Однако при невозможности использования JRE есть альтернативный вариант – установить клиент от SGD и с помощью его функционала авторизовываться в Secure Global Desktop (см. рис. 2). В случае если в системе не будет обнаружена Java, то браузер будет автоматически перенаправлен на локальную страницу загрузки клиента.

Фактически в этом случае для браузера не надо будет устанавливать Java-плагин, а вместо этого клиент от SGD сам запустит браузер по умолчанию и проделает работу по интеграции ввода/вывода браузера и серверной части. Для старта необходимо набрать полный адрес вашего установленного SGD-сервера: http://sgd.intra.net:9090/sgd/index.jsp.

Запускаем SGD-клиент:

```
$ ~/bin/ttatcc
```

Вводим полный адрес (см. выше) и перед нами появляется окно авторизации SGD (см. рис. 3).

Входим под пользователем Administrator (он же пользователь root в CentOS) – и перед нами главное окно управления сервером. Отсюда мы можем поменять серверные настройки, создать новые приложения, назначить права на запуск приложений и т.п. (см. рис. 4).

Нас же в первую очередь интересует консоль управления – Administration Console. Зайдем туда и попробуем создать приложение, которое перенаправляло бы пользователя на сторонний сервер – Windows Server 2003 – вкладка Applications.

Далее создаем объект – нажимаем на кнопку New и выбираем тип объекта – X Application. Заполняем название объекта, например RDP2Windows2003, и подтверждаем создание объекта клавишей Save (см. рис. 5).

Объект создан, теперь предстоит выбрать, как его запускать, и некоторые характиристики для отображения окна на клиентской машине.

Для этого нажимаем на название объекта и в открывшемся меню переходим на вкладку Launch.

В поле Application Command вбиваем путь к исполняемому файлу, который находится на CentOS: /home/ab/go_rdp.sh.

Рисунок 3. Окно авторизации в Secure Global Desktop

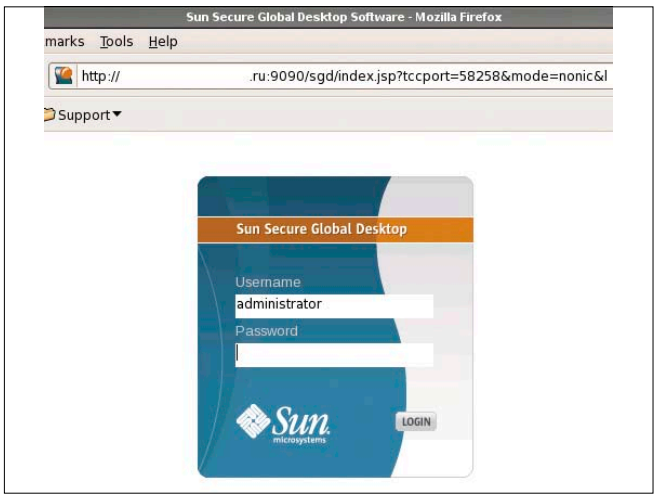
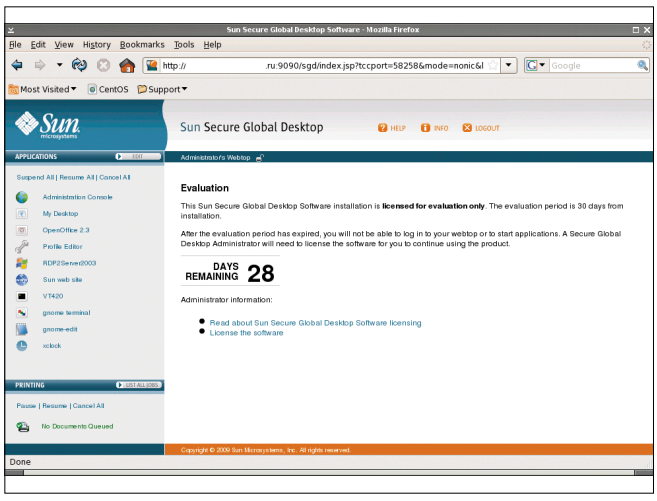


Рисунок 4. Под учетной записью администратора можно поменять серверные настройки



```
#!/bin/sh

rdesktop-vrdp win2003server.intra.net ↵
    -u RemoteAccessUser -g 1024x768
```

В поле Connection Method выбираем ssh-доступ и не забываем поставить флаг -X – перенаправление вывода X11-протокола (см. man ssh) (см. рис. 6).

Для удобства отображения приложений, в частности цветовая гамма, размер окна и т.д., предусмотрена вкладка Presentation. За выбор оконного отображения отвечает Window Type. Также можно зафиксировать размер окна – поле Window Size. Либо сделать его масштабируемым. И выберите цветовую гамму – Color Depth, – иначе вся радуга цветов будет ограничена 8 битами.

На этом можно было бы сказать, что настройка завершена и пользователи могут запускать свои браузеры. Однако пока рано рапортовать о готовности. Чтобы у пользователя появилась возможность запуска приложения RDP2Windows2003Server, этого пользователя (либо группу пользователей) надо назначить приложению – вкладка Assigned User Profiles. В таблицу Editable Assignments добавляем всех возможных пользователей.

Не забываем нажать на кнопку Save. Всё – теперь можно зазывать пользователей опробовать новый сервис – запуская из браузера RDP-сессию на сервер Windows 2003, вы можете забыть про сложности взаимодействия разных платформ и программных решений (см. рис. 7).

Ценность подобного решения – унификация. Вы даете пользователям свободу перемещения. Единственная их забота, чтобы под руками у них был веб-браузер с Java-плагином, да и чтобы не забывали свой пароль. А выделить приложения для запуска и синхронизировать внутренний документооборот с внутренними требованиями, я думаю, не так уж сложно (см. рис. 8).

Здесь был приведен пример, когда настройка приложений и всей системы, ведется через веб-интерфейс. Однако в еще тогдашнем продукте от Tarantella хорошо был продуман режим администрирования через командную строку. И фактически режим веб-администрирования – это только верхушка айсберга, полностью же функционал реализовывается через такие, например, команды, как:

```
# /opt/tarantella/bin/tarantella ↵
        object new_person
# /opt/tarantella/bin/tarantella ↵
        license info
```

Безопасность

Secure Global Desktop – достаточно интересная игрушка. Как с точки зрения технологии, так и возможностей применения. Следует помнить, что даже сильно «интеллектуальные» системы можно вывести из строя, если не произвести дополнительных манипуляций в настройке.

Первое, и наиболее важное, условие – ограничение доступа на уровне файловой системы. Хотя SGD и работает с правами пользователя ttaserv/

Рисунок 5. Создаем новое приложение для доступа к RDP-серверу

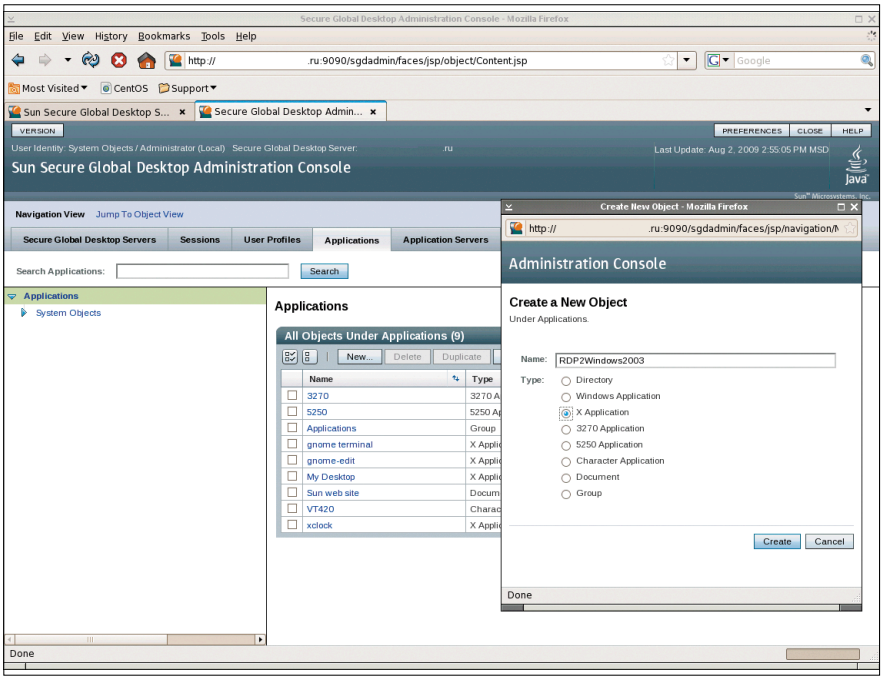
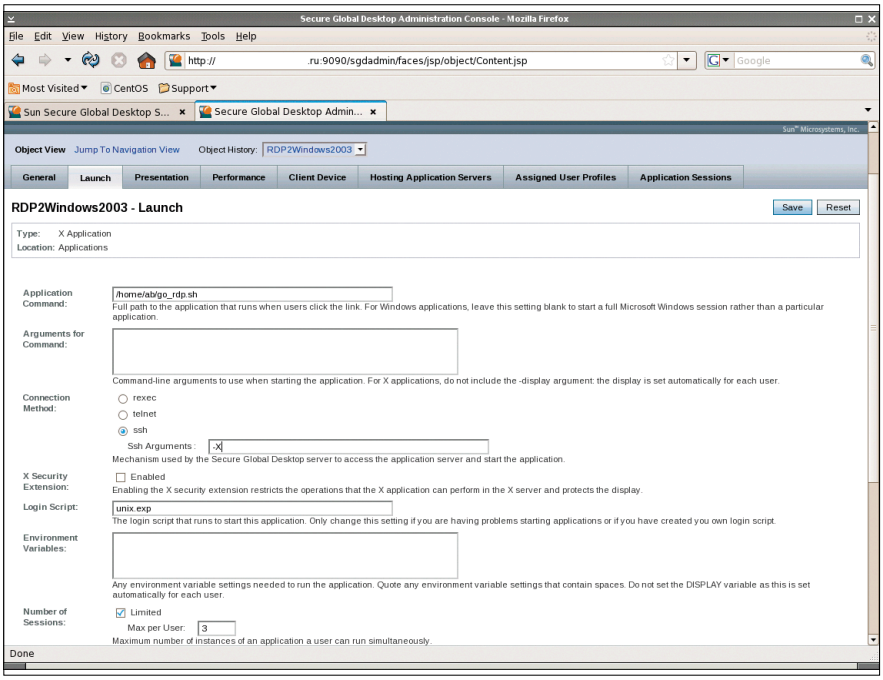


Рисунок 6. Выбор исполняемого файла и механизма доступа SGD-сервера к серверу приложений



ttasys, при первоначальной настройке apache/tomcat лучше задействовать порты выше 1024. Второй момент – закрывайте файерволом порты, которые не потребуются при работе серверных частей (см. Sun Secure Global Desktop 4.5 – Administration Guide, пункт Firewalls [2]). В-третьих, по умолчанию, работа с клиентскими местами ведется по открытому, незашифрованному протоколу HTTP. Поэтому предполагается, что специалисты, ответственные за функционирование SGD, самостоятельно заведут SSL-сертификаты (например, самоподписанные – с помощью OpenSSL, либо для работы во внешней сети – от компании VeriSign) и перейдут на использование HTTPS (такой функционал в комплекте предоставляется, но закомментирован как в Apache, так и в Tomcat). И четвертое условие, которое касается взаимодействия SGD и серверов приложений внутри предприятия – следует использовать инкапсуляцию передачи данных посредством SSH и внимательно отнестись к документации на протоколы данных. Поскольку RDP представляет возможность шифрации трафика, а вот ICA (от Citrix) не всегда – представляет только безопасную передачу пароля.

Рисунок 7. Запущено SGD-приложение RDP2Windows2003, обеспечивающее доступ на RDP-сервер

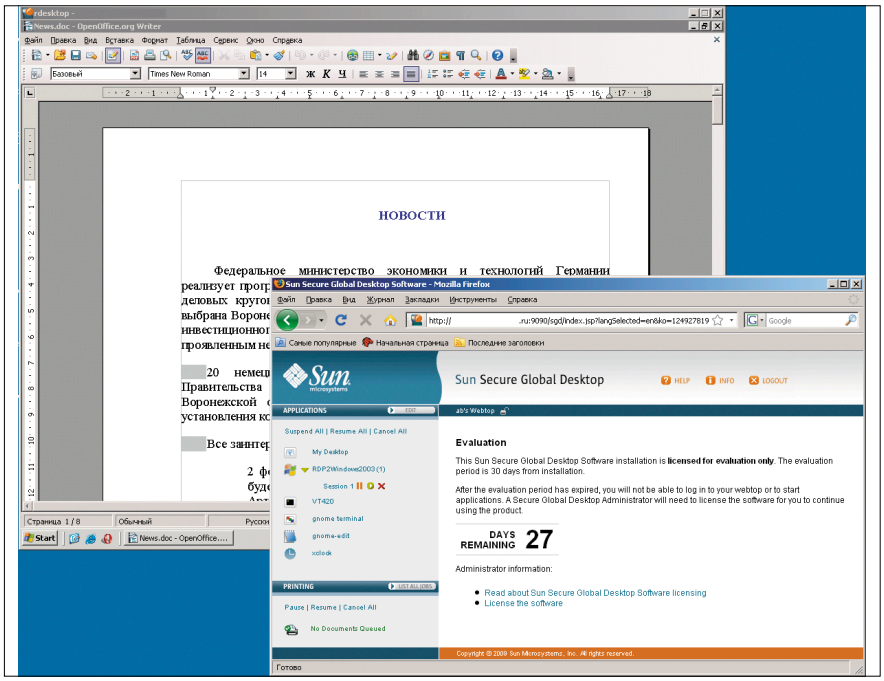
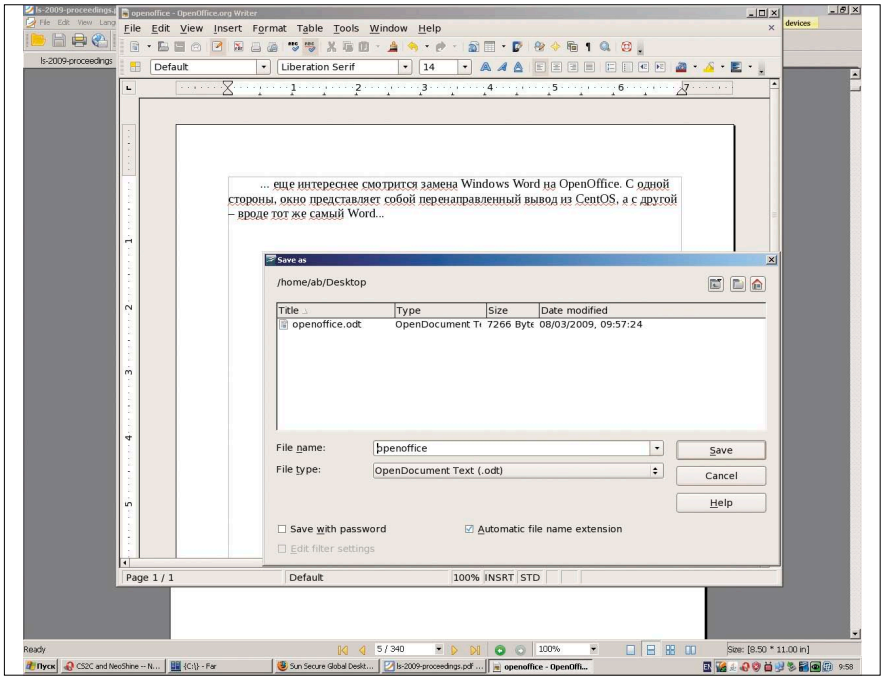


Рисунок 8. Перевести документооборот с Microsoft Office на OpenOffice, развернутом в CentOS, достаточно тривиально



По большому счету четвертое условие наиболее актуально для интеграции SGD с портальными решениями. Когда используются портлеты, являющиеся на самом деле «обертками» для SGD-приложений. Которые в свою очередь запускают то или иное приложение с сервера приложений [3].

Лицензирование

Как и любой коммерческий продукт, Secure Global Desktop является платным. Сия особенность проявляется в годовой подписке, т.е. за каждого пользователя платится лицензионное отчисление за год. Фактически вы платите за годовое обслуживание серверной части [2] и выбираете число пользователей, которые будут зарегистрированы на SGD-сервере. Так, например, годовое обслуживание сервера с поддержкой UNIX-протоколов и 25 пользователей выльется в сумму 219 + 25*103 = 2794 USD. В дальнейшем стоимость лицензии для клиентских мест может составлять в разы меньше, нежели в первый год обслуживания [4]. Насколько она будет круглой для вашей организации и перевесят ли плюсы минусы – судить вам. Думаю, что за 30 дней вполне можно будет выразить свое непредвзятое мнение. EOF

1. http://www.sun.com/software/products/sgd/get.jsp.
2. http://dlc.sun.com/pdf/820-6690/820-6690.pdf.
3. http://rabbs.com/uuasc/ssgd.pdf.
4. http://catalogs.sun.com/is-bin/INTERSHOP.enfinity/WFS/Sun_Catalogue-Sun_Catalogue_RU-Site/en_US/-/USD/ViewCatalog-Browse?CatalogCategoryID=0cVIBe.ddWAAAAEUxV05G_c2&PriceIt=true.
5. http://www.sun.com/software/products/sgd/whitepapers/wp_tei_sgd.pdf.

Решите проблемы **лицензирования ПО** с помощью профессионалов!

Операционная система GNU/Linux и свободное программное обеспечение помогут вам с минимальными затратами решить проблему лицензирования программного обеспечения, повысить безопасность и надежность вашей компьютерной сети.

Компания ГНУ/Линуксцентр предлагает вам внедрение ОС GNU/Linux и свободного программного обеспечения, реализацию и техническую поддержку сложных технических решений на базе свободного ПО, обучение ваших сотрудников — как пользователей, так и технических специалистов.

С НАШЕЙ ПОМОЩЬЮ ВЫ СМОЖЕТЕ:

- оптимизировать затраты на лицензирование ПО за счет максимально возможного использования свободного ПО;
- существенно сократить время системных администраторов, затрачиваемое на устранение последствий деятельности вирусов и сбоев в программном обеспечении.

ТИПОВЫЕ ПРОЕКТЫ:

- миграция рабочих станций и серверов с Microsoft Windows на GNU/Linux;
- установка 1С на серверах и рабочих станциях под управлением GNU/Linux;
- миграция с Microsoft Windows Active Directory на Mandriva Directory Server;
- миграция с Microsoft Exchange на Zimbra;
- внедрение интернет-телефонии на базе Asterisk;
- внедрение свободной CRM-системы SugarCRM;
- создание кластеров высокой доступности;
- реализация терминальных решений;
- создание порталов любой сложности на базе свободных CMS-систем — Joomla, Drupal, Plone;
- внедрение защищенных систем на основе сертифицированного ФСТЭК ПО.

СРЕДИ НАШИХ КЛИЕНТОВ:

- Правительство Московской области;
- Правительство Нижегородской области;
- администрация Черниговского района Приморского края;
- Министерство финансов республики Саха (Якутия);
- Владивостокский государственный университет экономики и сервиса;
- группа компаний «ИМАГ»;
- компания «Азбука мебели»;
- компания «Бестли — выставочные материалы»

и другие организации различного профиля.

Наш опыт внедрения свободного программного обеспечения в компаниях различного профиля поможет выбрать оптимальное сочетание свободного и коммерческого программного обеспечения, подходящее именно для вашей организации, а также поможет избежать техничеких и организационных проблем при внедрении свободного ПО.

Linux center www.linuxcenter.ru

Департамент внедрений компании ГНУ/Линуксцентр
Телефон в Москве: (499) 271-49-54,
в Санкт-Петербурге: (812) 309-06-86

ЗВОНИТЕ СЕЙЧАС!

Реклама

Визитка

ВАДИМ АНДРОСОВ, ассистент ВНУ, специалист МСР. Занимается анализом архитектур организаций с защищаемыми бизнес-процессами

Управляем доступом
к ресурсам домена на основе Windows Server

Это решение позволит распределять права доступа с учетом того, с какой рабочей станции выполнен вход

Общее описание концепции

Сегодня я хотел бы показать реализацию не вполне обычного подхода к управлению доступом к файловым ресурсам домена на основе Windows Server. Обычно администратор разрешает или запрещает операции с файлами и папками, используя объекты пользователей и групп. Это стандартный подход, который прекрасно себя зарекомендовал. Однако иногда права удобно предоставить рабочей станции.

Рассмотрим пример. На защищаемой территории предприятия пользователи работают на компьютерах, на которых нельзя воспользоваться съемными носителями информации. Кроме того, доступ в закрытые отделы ограничен. Даже владея паролем сотрудника, злоумышленник не сможет физически добраться до необходимой рабочей станции.

В то же время в организации существует некоторый отдел, где политика безопасности не настолько сурова и к машинам имеется свободный доступ. В результате достаточно заполучить пароль сотрудника с необходимыми правами, чтобы получить доступ к данным, не сталкиваясь с усиленной пропускной системой и техническими ограничениями.

Проблема может быть решена запретом доступа к некоторым ресурсам для пользователей, работающих на определенных рабочих станциях, даже если у них имеются необходимые права. То есть, грубо говоря, появляется возможность сказать: «С этой машины в эту папку заходить нельзя, кто бы за ней ни работал».

Интересно, что у администратора есть возможность с помощью штатных средств запретить доступ для объекта компьютера к определенному ресурсу (см. рис. 1).

Однако такая настройка не запрещает пользователям, работающим на этом компьютере, доступа к папке. Он может свободно создавать, смотреть и модифицировать файлы, если, конечно, имеет на это разрешение.

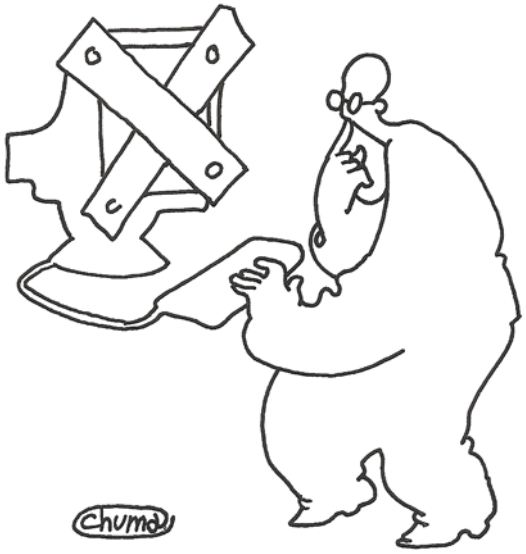
Здесь будет рассмотрена простейшая работоспособная реализация механизма распределения прав доступа к ресурсам домена, когда полномочия предоставляются не пользователю или группе, а рабочей станции. Для расширения функциональности решения придется модифицировать схему AD, добавлять новую функциональность в оснастки оснасток, как это делалось в [1-3]. Однако мне хотелось бы показать саму суть подхода, не загромождая его посторонними деталями, которые к тому же были подробно рассмотрены ранее. Предлагаемое решение вполне применимо в реальных ситуациях. Не стоит забывать, что чем проще, тем надежнее.

Основная идея состоит в привязке к каждой рабочей станции группы безопасности. При входе (login) на компьютер пользователь будет автоматически заноситься в эту группу. Я решил, что будет достаточно обрабатывать событие входа. При этом пользователь будет добавляться в группу компьютера и удаляться из других аналогичных групп, если он к какой-то из них принадлежал. Таким образом, отдельно события выхода (logoff) обрабатывать не нужно. Тем более что надеяться на то, что оно произойдет, нельзя – работа с компьютером может быть завершена аварийно.

Сделать прикрепление группы к рабочей станции можно разными способами. Наиболее универсальный – с помощью создания специального свойства класса (этот подход использовался, например, в [2]). Однако сегодня я остановлюсь на связывании на основе имени объекта. Прикрепленная группа будет называться <<имя_компьютера>>_cg (сокращение от Computer Group). Храниться эти группы будут в том же контейнере Active Directory, что и компьютеры. Как будет видно в дальнейшем, реализация позволяет хранить привязанные группы в дочернем контейнере по отношению к тому, в котором находится компьютер. Правила именования соблюдать все равно нужно, поскольку это единственный механизм связывания объектов в предлагаемом решении. Итак, все, что нужно сделать, это обработать событие входа пользователя на компьютер, удалить его из прошлой группы, если он в какую-либо входил, и добавить в новую.

Реализация

Приступим. В [1] я описывал обработку события входа пользователя на рабочую станцию, отслеживая процесс появления экземпляров класса Win32_NTLogEvent. Что они собой представляют, вы можете увидеть, открыв оснастку Event Viewer (см. рис. 2).



Основная идея состоит в привязке к каждой рабочей станции группы безопасности

События происходит огромное количество, нас же интересует лишь одно с кодом 672 и названием Authentication Ticket Granted. Это и есть аутентификация нового пользователя. Установим обработчик события. Этот процесс описан в [2, 3], поэтому ограничусь лишь кратким описанием.

Нужно создать экземпляры трех классов. Сначала подключаемся к пространству имен, где расположен класс Win32_NTLogEvent. Это root\CIMv2.

```
Set objSWbemServices = GetObject("winmgmts:" & ↵
       "{impersonationLevel=impersonate}!\\.\root\CIMv2")
```

Затем создаем первый класс обработчика. Это фильтр событий. С его помощью система получает информацию, что мы хотим обрабатывать. Интересующее нас событие описывается с помощью запроса на языке WQL (WMI Query Language, является упрощенной версией SQL). Здесь используется запрос вида:

```
select * from __instanceCreationevent where ↵
        targetinstance isa 'Win32_NTLogEvent' ↵
        and targetinstance.EventIdentifier = 672
```

Так мы сообщаем системе, что собираемся обрабатывать события создания экземпляра (__instance Creationevent) класса Win32_NTLogEvent. Причем интересны не все объекты (которых создается довольно большое количество), а лишь те, идентификатор (EventIdentifier) которых 672. В листинге 1 создается объект класса фильтр событий (__EventFilter), инициализируются его поля (имя объекта, язык запроса, текст запроса и пространство имен). Затем с помощью метода Put_ класс помещается в постоянное хранилище.

Листинг 1. Создание фильтра событий

```
Set eventFilterClass = objSWbemServices.Get("__EventFilter")
set userFilter = eventFilterClass.SpawnInstance_()
userFilter.Name = "SecureCompFilter"
userFilter.QueryLanguage = "WQL"
userFilter.Query = ↵
```

Рисунок 1. Запрет всех операций для рабочей станции

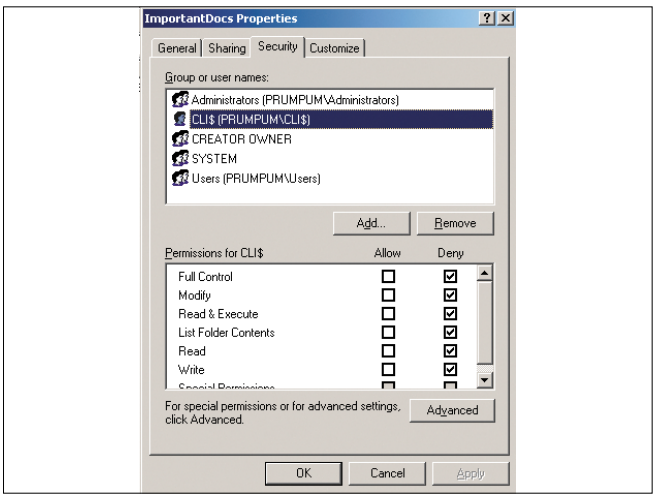
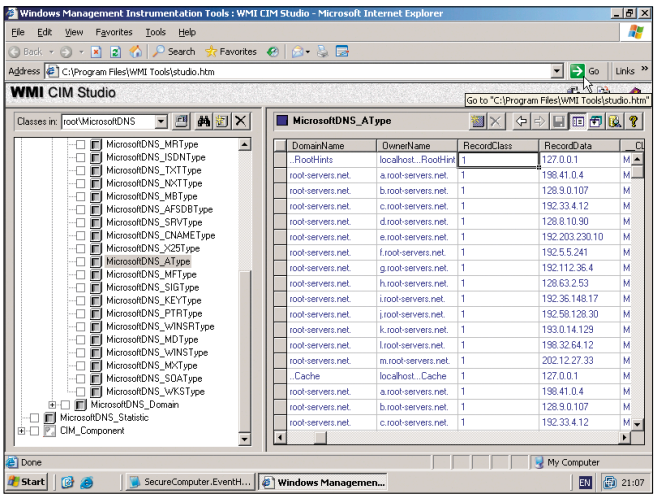


Рисунок 2. Оснастка Event Viewer



```
        "select * from __instancecreationevent " & ↵
        "where targetinstance isa 'Win32_NTLogEvent' " & ↵
        "and targetinstance.EventIdentifier = 672"
userFilter.EventNamespace = "root\CIMv2"
userFilter.Put_()
```

Затем аналогичным образом нужно создать и поместить в хранилище класс, который отвечает за действия, выполняемые в ответ на событие. Нам требуется выполнить определенный сценарий. Для этого в Windows существует специальный класс, который, однако, нужно сначала откомпилировать. Для этого выполните следующую команду:

```
mofcomp %SYSTEMROOT%\system32\wbem\scrcons.mof
```

После этого переходим к созданию объектов класса:

Листинг 2. Создание объекта реакции на событие

```
Set consumerClass = ↵
        objSWbemServices.Get("ActiveScriptEventConsumer")
set userConsumer = consumerClass.SpawnInstance_()
userConsumer.Name = "RunSecureCompScript"
userConsumer.ScriptText = "createObject("& chr(34) & ↵
        "SecureComputer.EventHandler" & chr(34) & ↵
        ").handle(targetEvent.targetInstance)"
userConsumer.ScriptingEngine = "VBScript"
userConsumer.Put_()
```

В качестве исполняемого сценария передается строчка:

```
createObject(SecureComputer.EventHandler).handle( ↵
        targetEvent.targetInstance)
```

Здесь создается объект класса SecureComputer.EventHandler, и сразу же вызывается его метод handle, в качестве параметра которому передается объект типа Win32_NTLogEvent, вызвавший событие.

Чтобы активировать обработку событий требуется связать два созданных выше объекта. Это делается посредством создания экземпляра класса __FilterToConsumerBinding, в котором посредством имен указывается, что с чем связывать. Имена классов должны содержать двойные кавычки, которые напрямую нельзя использовать в строках VBScript. Поэтому пришлось воспользоваться добавлением кавычек по коду символа с помощью функции chr(34).

Листинг 3. Объект для связи фильтра события с его обработчиком

```
Set bindingClass = ↵
        objSWbemServices.Get("__FilterToConsumerBinding")
set userBinder = bindingClass.SpawnInstance_()
userBinder.Filter = "__EventFilter.Name=" & ↵
        chr(34) & "SecureCompFilter" & chr(34)
  userBinder.Consumer = ↵
        "ActiveScriptEventConsumer.Name=" & chr(34) & ↵
        "RunSecureCompScript" & chr(34)
userBinder.Put_()
```

Как только последний объект помещается в хранилище (см. листинг 3), события начинают обрабатываться. Причем проделывается эта операция один раз. При перезагрузках системы ничего не отключается. Если же все-таки нужно отменить обработку, для этого нужно выполнить сценарий из листинга 4. В нем находятся объекты, созданные в листингах 1-3, и удаляются из хранилища.

Листинг 4. Отключение прослушивания событий

```
Set objWIMService = GetObject("winmgmts:\\.\root\CIMv2")
Set objList = objWIMService.ExecQuery("references of ↵
        {__EventFilter.Name='SecureCompFilter'}")
For each objInst in objList
  objInst.Delete_
Next
Set obj = GetObject("winmgmts:\\.\root\CIMv2:" & ↵
        "ActiveScriptEventConsumer='RunSecureCompScript'")
obj.Delete_
Set obj = GetObject("winmgmts:\\.\root\CIMv2:" & ↵
        "__EventFilter='SecureCompFilter'")
obj.Delete_
```

Перейдем к написанию основного класса. Код можно было оформить и в виде простого сценария, но использование классов позволяет абстрагироваться от физического расположения файла с текстом программы, что удобно, особенно в случае работы с классом ActiveScriptEvent Consumer, отредактировать поля которого проблематично.

Используя WSC (Windows Script Component), мы можем свободно перемещать файл, заново регистрируя его в системе. На способе его использования это не отразится. Создание классов подробно рассматривалось в [3], поэтому здесь ограничимся поверхностным описанием.

Рисунок 3. Объекты для тестирования

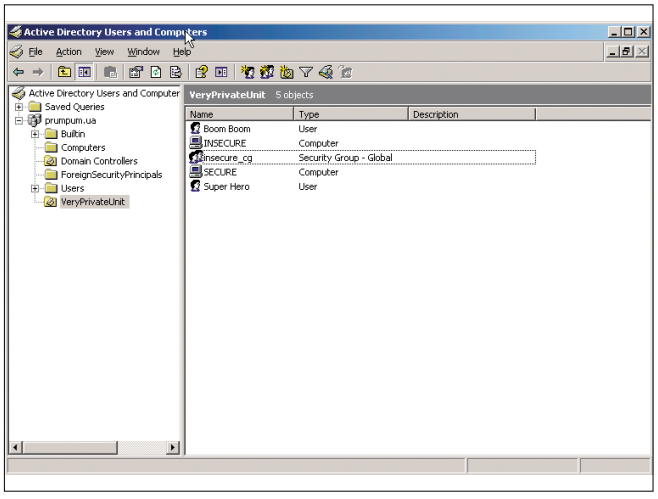
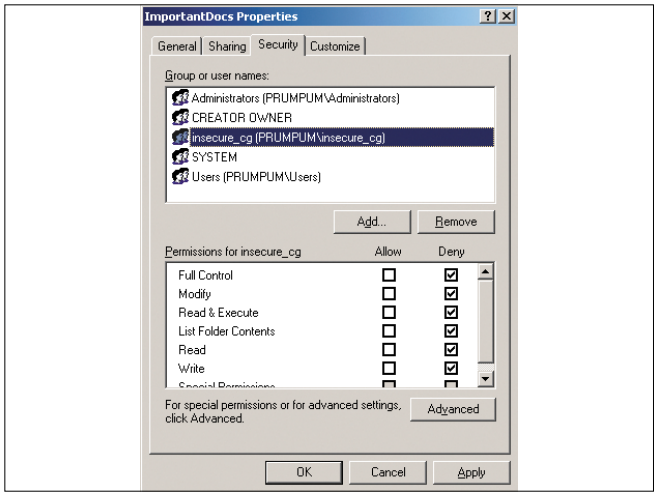


Рисунок 4. Запрет работы с ресурсом для группы insecure_cg



Это обычный текстовый файл с расширением wsc, в котором в нашем случае будет содержаться следующий код:

Листинг 5. Заготовка класса SecureComputer.EventHandler

```
<?xml version="1.0"?>
<component>
<registration
        description="SecureComputer.EventHandler"
        progid="SecureComputer.EventHandler"
        version="1.00"
        classid=
      "{5599b415-12e5-4d68-bee5-0746988c4140}"
>
</registration>
<public>
        <method name="handle">
                <PARAMETER name="eventInstance"/>
        </method>
</public>
<script language="VBScript">
<![CDATA[
Const ADS_SCOPE_SUBTREE = 2
function handle(eventInstance)
        bindUserComp _
          eventInstance.InsertionStrings(9), _
          eventInstance.InsertionStrings(0)
end function
Function bindUserComp(ip, userName)
'…
End Function
]]>
</script>
</component>
```

Это XML-документ. В разделе registration содержится информация, необходимая для регистрации компонента (класса) в системе: название (progid), описание, версия, идентификатор. Лучше создавать заготовку класса с помощью специальных утилит (например, Windows Script Component Wizard), которые сами генерируют корректный уникальный идентификатор (classid).

Раздел public содержит открытые методы класса, т.е. те, которыми могут пользоваться внешние программы. В нашем случае это единственный метод handle, который и вызывается в обработчике события. В качестве параметра ему передается объект события.

Дальше идет реализация метода. Метод handle достаточно прост – он сводится к вызову подпрограммы bindUserComp, которая и делает всю необходимую работу. У объекта события есть свойство InsertionStrings, которое представляет собой массив строк отчета. В строке с индексом 0 содержится имя пользователя (samAccountName), с индексом 9 – IP-адрес компьютера, с которого осуществляется вход. Остальная информация для этой задачи несущественна.

Рассмотрим подробно функцию прикрепления пользователя к компьютеру. Фактически она заносит объект user в группу безопасности, соответствующую определенной рабочей станции, и удаляет его из предыдущей, если он в таковой находится. Если прикрепленной группы не существует – ничего не происходит. Таким образом, нет необходимости создавать отдельные группы для каждого компьютера – только для тех, для которых это действительно нужно.

```
Function bindUserComp(ip, userName)
```

В самом начале проверяется IP-адрес, если это контроллер домена, на котором запущен предлагаемый механизм, то ничего не произойдет. Обычно на такие машины доступ есть только у администраторов домена, от которых предлагаемое решение все равно защитить не в состоянии.

```
if ip = "127.0.0.1" then exit function
```

Далее создаются необходимые объекты.

ADSystemInfo. Информация о системе. С его помощью получим название нашего домена.

ADODB.Connection. Имени пользователя и IP-адреса компьютера недостаточно, требуется привязаться к соответствующим объектам Active Directory. Для поиска удобно использовать Active Directory Provider, который позволяет делать запросы к службе каталогов, используя SQL-подобный язык. Объект соединения инициализируется для работы именно с этим средством.

ADODB.Command. Представляет собой запрос. Инициализируется текстом запроса и областью поиска. Здесь поиск производится в поддереве, начиная с заданного корневого элемента, для этого используется константа **ADS_SCOPE_SUBTREE.** Ее нужно определить выше или просто использовать цифру 2.

RootDSE. Заранее неизвестно, какой организационной единице принадлежит пользователь, поэтому искать нужно, начиная с общего корневого элемента. В Active Directory он называется так:

```
set info = createObject("ADSystemInfo")
Set objConnection = CreateObject("ADODB.Connection")
Set objCommand =   CreateObject("ADODB.Command")
objConnection.Provider = "ADsDSOObject"
objConnection.Open "Active Directory Provider"
```

```
Set objCOmmand.ActiveConnection = objConnection
objCommand.Properties("Searchscope") = ADS_SCOPE_SUBTREE
Set rootDSE = GetObject("LDAP://RootDSE")
```

Вспомогательные объекты созданы, переходим к непосредственной работе. Сначала нужно отыскать объект пользователя по известному имени. Для этого используется запрос поиска объектов класса user, со свойством samAccountName, равным искомому. Если по каким-то причинам пользователь не найден (свойство EOF команды истинно), функция завершает работу. В рамках домена свойство samAccountName уникально (в отличие от cn (Common Name), которое у разных учетных записей может совпадать), поэтому в результате будет получена только одна запись, к которой программа и переходит посредством вызова метода MoveFirst результата запроса.

Выбирается свойство ADSPath (полный путь к объекту), на основе которого сразу же выполняется непосредственное подключение с помощью метода getObject. После этого объект user для манипуляций готов. После ключевого слова select указывается контейнер, с которого нужно начать поиск. В нашем случае это корневой элемент Active Directory.

```
objCommand.CommandText = "Select ADsPath from 'LDAP://" & ↵
        rootDSE.Get("defaultNamingContext") & "' " ↵
        & "Where objectClass='user' and " & ↵
        "samAccountName ='" & userName & "'"
Set objRecordSet = objCommand.Execute
if objRecordSet.EOF then exit function
objRecordSet.MoveFirst
Set user = getObject(objRecordSet.Fields(0).value)
```

Дальше нужно найти компьютер. Здесь используется другой подход, потому что IP-адрес не является свойством объекта computer. Адрес нужно сначала разрешить (resolve), обратившись к DNS-серверу. Для этого существует специальный объект WMI MicrosoftDNS_AType, который находится в пространстве имен root\MicrosoftDNS. Здесь также поддерживается SQL-подобный язык запросов (тот самый WQL, который использовался для подписки на события).

В таблице IP-адрес может встречаться несколько раз, поэтому уточним, что искать нужно только для текущего домена (info.domainDNSName). В поле OwnerName результата будет содержаться доменное имя компьютера в виде computer.domain. Оно уникально для каждой рабочей станции, поэтому, зная его, можно искать объект в AD.

Обратите внимание на дополнительную проверку при анализе результатов выборки. Дело в том, что существует DNS-запись для самого домена, которая соответствует адресу одного из контроллеров, ее нужно проигнорировать.

```
Set objWMIService = ↵
        GetObject("winmgmts:\\.\root\MicrosoftDNS")
Set colItems = objWMIService.ExecQuery( ↵
        "SELECT * FROM MicrosoftDNS_AType " & ↵
        "WHERE IPAddress = '" & ip & ↵
        "' and domainName = '" & info.domainDNSName & "'")
For Each objItem in colItems
  if objItem.OwnerName <> info.domainDNSName then ↵
        DNSHostName = objItem.OwnerName
next
```

Теперь, зная имя компьютера, ищем его в каталоге так же, как искали пользователя. Класс объекта теперь выставляем computer, а поле, по которому осуществляется поиск, – DNSHostName.

```
objCommand.CommandText = "Select ADsPath from 'LDAP://" & ↵
        rootDSE.Get("defaultNamingContext") & "' " ↵
        & "Where objectClass='computer' and " & ↵
        "DNSHostName ='" & DNSHostName & "'"
Set objRecordSet = objCommand.Execute
objRecordSet.MoveFirst
Set computer = getObject(objRecordSet.Fields(0).value)
```

Остается найти группу, привязанную к компьютеру. Используем нашу договоренность об именовании – группа должна называться так же, как рабочая станция, с суффиксом _cg и находиться в той же организационной единице (или дочерней, поскольку область поиска запросов – поддерево).

Такой группы может и не быть. Поэтому после выполнения запроса проверяем свойство EOF, результирующей выборки (objRecordSet).

```
group = null
objCommand.CommandText = "select ADSPath from '" & ↵
        computer.parent & "' where objectClass = ↵
        'group' and cn = '" & computer.cn & "_cg'"
```

Рисунок 5. Запрет на использование ресурса

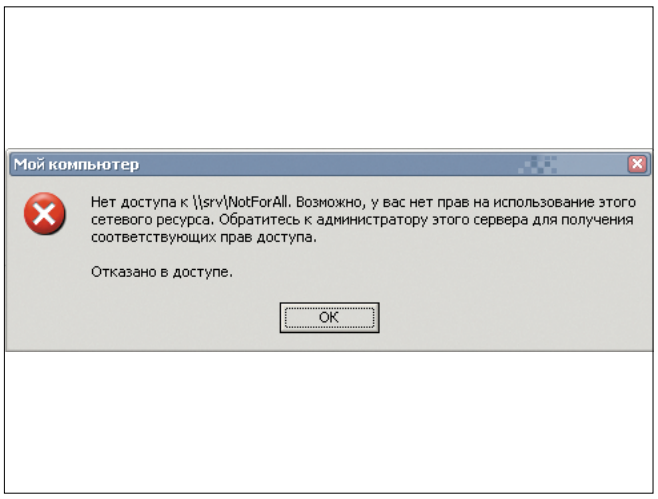
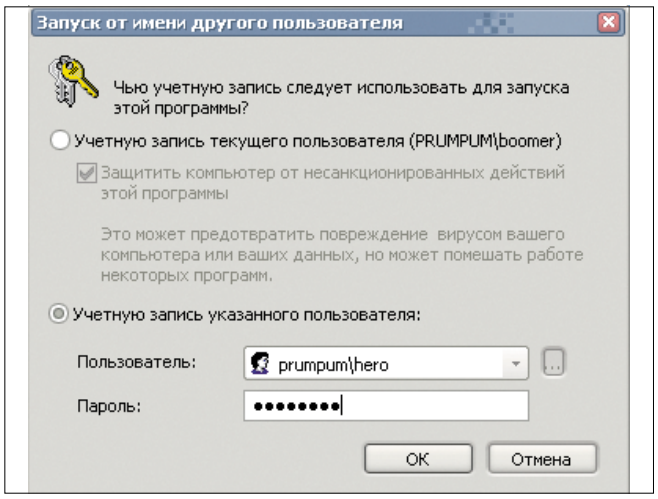


Рисунок 6. Вторичный вход в систему



```
Set objRecordSet = objCommand.Execute
if not objRecordSet.EOF then
  objRecordSet.MoveFirst
  set group = getObject(objRecordSet.Fields(0).value)
end if
```

Если группа была найдена и пользователь уже ей принадлежит (если он в предыдущий раз заходил с этого компьютера), то функция завершает работу.

```
if not isNull(group) then
  if group.isMember(user.ADSPath) then exit function
end if
```

Затем вне зависимости от того, была найдена присоединенная группа или нет, требуется удалить пользователя из всех групп других рабочих станций. Здесь он удаляется из групп, имя которых оканчивается на _cg.

```
for each ug in user.groups
  if len(ug.cn)>3 then
    if right(ug.cn,3) = "_cg" then
      ug.remove user.ADSPath
    end if
  end if
next
```

Если группа была найдена, пользователь в нее добавляется.

```
if not isNull(group) then group.add(user.ADSPath)
End Function
```

Испытания

Посмотрим теперь, как это работает. Сначала нужно зарегистрировать наш класс в системе (с помощью контекстного меню) и установить прослушивание событий. Все готово к работе. Никаких особых инструментов мы не добавляли. Все что требуется – это оснастка Active Directory Users and Computers и стандартные средства системы управления правами доступа операционной системы.

Пусть в домене будет две рабочие станции: защищенная под названием secure, для которой ограничения не нужны, и доступная всем insecure. В том же контейнере создадим группу insecure_cg. Она и будет содержать работающих на этой машине пользователей (см. рис. 3).

Затем создадим открытую папку, с содержимым которой запретим работать с компьютера insecure (см. рис. 4).

Все. Теперь если пользователь, наделенный достаточными правами, обратится к папке с рабочей станции secure, никаких проблем не будет. Если тот же пользователь зайдет с insecure, то доступа уже не получит (см. рис. 5).

Более того, поскольку отслеживается процесс аутентификации пользователя, то корректно будут обрабатываться и случаи вторичного входа в систему (см. рис. 6). В группе, привязанной к рабочей станции, будет содержаться несколько пользователей.

Вам было представлено простое и удобное в использовании решение, которое позволит предоставлять права к ресурсам пользователям в зависимости от рабочей станции, на которой они находятся. Никаких дополнительных элементов и модификаций схемы Active Directory не требуется, поскольку используется лишь соглашение об именах. Отключать расширение нет необходимости – если нет групп, привязанных к компьютерам, ничего происходить не будет.

Информация обновляется при каждой аутентификации пользователя, поэтому проблемы, вызванные некорректным завершением работы, практически исключены. Но даже в случае сбоя ситуация может быть легко исправлена простым редактированием состава необходимых групп. EOF

1. Андросов В. Реализуем нестандартные правила управления доступом на основе архитектуры организации в Windows Server 2003. //Системный администратор, №10, 2007 г. – С. 48-58.
2. Андросов В. Синхронизация ACL и структуры организации. Часть 1. //Системный администратор, №12, 2007 г. – С. 36-41; часть 2. №1, 2008 г. – С. 56-61; часть 3. №2, 2008 г. – С. 82-87.
3. Андросов В. Делегируем права на перемещение учетных записей пользователей в Active Directory. Часть 1. Постановка задачи. //Системный администратор, №3, 2009 г. – С. 16-21.
   Андросов В. Делегируем права на перемещение учетных записей пользователей в Active Directory. Часть 2. Реализация основных функций. //Системный администратор, №4, 2009 г. – С. 24-30.
   Андросов В. Делегируем права на перемещение учетных записей пользователей в Active Directory. Часть 3. Реализуем необходимые операции. //Системный администратор, №5, 2009 г. – С. 30-37.
   Андросов В. Делегируем права на перемещение учетных записей пользователей в Active Directory. Часть 4. Завершение надстройки. //Системный администратор, №6, 2009 г. – С. 40-47.
   Андросов В. Проводим реализацию тонкого делегирования прав в Active Directory. //Системный администратор, №7, 2009 г. – С. 32-38.

Визитка

ИГОРЬ ШТОМПЕЛЬ, инженер, системный администратор. Сфера профессиональных интересов – GNU/Linux, функциональное программирование

Новые возможности Nmap 5.00 – программы для исследования безопасности сетей

В июле была представлена пятая версия Nmap с большим количеством изменений. Поговорим о наиболее значимых из них

Nmap 5.00 – первый стабильный после версии 4.76 и крупный выпуск с 2007 года. Как следует из официального заявления, «это наиболее значимый релиз с момента появления Nmap в 1997 году» [1]. Если перейти на язык цифр, то получается около 600 значительных изменений после выхода предыдущего крупного релиза 4.50.

Nmap (Network Mapper, сетевой картограф) – это программа, которая ориентирована на исследование и аудит безопасности сетей. Она представляет собой свободное программное обеспечение, лицензированное под GNU GPL v2. Nmap является кроссплатформенным и доступен для GNU/Linux, FreeBSD, OpenBSD, NetBSD, Sun Solaris, HP-UX, IRIX, Mac OS X, Amiga, Windows.

Несмотря на сегодняшнюю кроссплатформенность, первоначально Nmap разрабатывался для UNIX-систем. В основе его создания лежало недовольство возможностями и ограничениями существующих на тот момент сканеров безопасности (Julian Assange's Strobe, SATAN и другими). Гордону Лиону (Gordon Lyon), известному под ником Fyodor, хотелось иметь программу, которая обладала бы всеми необходимыми функциями. После ряда попыток модификации доступных сканеров он решился написать свою программу. Основная цель, которую преследовал разработчик, – создать быстрый сканер, поддерживающий все виды сканирования. После непродолжительного самостоятельного использования получившейся программы, Гордон Лион передал ее в журнал Phrack. Статья, вышедшая 1 сентября 1997 года в 51-м номере журнала и получившая название «The Art of Port Scanning» (Искусство сканирования портов), доступна по адресу http://www.phrack.org/issues.html?issue=51&id=11#article. А затем Nmap стал известной и успешной реализацией эффективного сканера безопасности сетей [2].

Установка Nmap 5.00

В используемый мной дистрибутив Ubuntu 9.04 включен Nmap 4.76. Поэтому я загрузил файл с исходными текстами программы версии 5.00 отсюда: http://nmap.org/dist/nmap-5.00.tar.bz2. Далее распаковал архив и установил Nmap:

```
./configure && make
sudo make install
```

После завершения установки программа вывела оригинальное изображение, показанное на рис. 1.

Nmap 5.00 – наиболее значимые изменения

На официальном сайте программы выделены следующие наиболее значимые новшества этого релиза: включение в Nmap – Ncat, появление Ndiff, улучшение производительности, выпуск книги «Nmap Network Scanning» – официального руководства по работе с программой, усовершенствование работы Nmap Scripting Engine. Рассмотрим их подробнее.

Ncat – это утилита, предназначенная для чтения и записи данных по сети с использованием командной строки. Она впервые была интегрирована в Nmap версии 4.85 BETA 1. Программа специально разработана для проекта Nmap как усовершенствованный вариант Netcat (http://sectools.org/#netcat). Ncat позволяет создавать «цепочку» из нескольких программ Ncat для перенаправления портов TCP и UDP на порты или хосты. Кроме того, поддерживается работа по протоколу SSL, подключения через прокси (с проверкой подлинности) с использованием SOCKS4 или HTTP (метод CONNECT) [3].

Ncat может быть использована как TCP/UDP/SCTP/SSL-клиент для взаимодействия с веб-серверами, серверами Telnet, почтовыми серверами и другими сервисами на базе TCP/IP. Другими словами, Ncat позволяет определять ошибки, находить уязвимости в системе безопасности в работе различных сервисов. Программа может выполнять роль TCP/UDP/SCTP/SSL-сервера и позволяет захватывать каждый байт, посылаемый клиентами. Ncat имеет функциональность, позволяющую выступать в роли прокси-сервера и перенаправлять трафик между портами или хостами с использованием протоколов TCP/UDP/SCTP. Например, запустить HTTP-прокси-сервер на порту 8080 локальной машины можно так:

```
ncat -l --proxy-type http localhost 8080
```

Если перейти на язык цифр, то получается около 600 значительных изменений

Кроме того, имеется возможность шифрования данных соединений с использованием SSL (поверх IPv4 или Ipv6).

В новой версии Nmap за сравнение результатов сканирования отвечает программа Ndiff (http://nmap.org/ndiff). Она сравнивает два XML-файла и на выводе отображает различия между ними. Ndiff покажет, какие хосты стали доступны, а какие, наоборот, нет; какие порты стали открыты или закрыты. Синтаксис команды следующий:

```
ndiff [options] {a.xml} {b.xml}
```

options – опции программы, полный перечень которых можно получить, дав команду:

```
man ndiff
```

a.xml и b.xml – первый и второй соответственно файлы с результатами сканирования для сравнения.

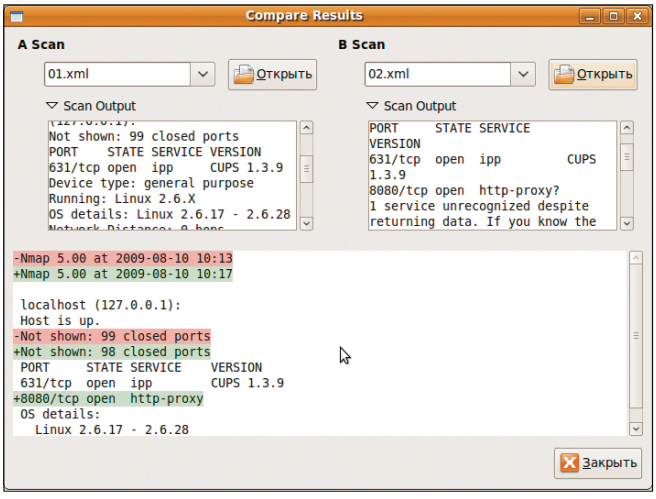
В качестве примера проведу сравнение двух файлов с результатами сканирования. Для этого запускаю Zenmap (после установки на моем компьютере в меню «Приложения → Интернет» появились две команды Zenmap и Zenmap (as root) с правами суперпользователя. Последний представляет собой GUI для Nmap, поставляемый с программой и разрабатываемый с использованием языка Python. После запуска Zenmap провожу первое сканирование: в поле Target прописываю локальный петлевой интерфейс (127.0.0.1), в поле Profile выбираю Quick scan plus, затем нажимаю кнопку Scan. Полученный результат сохраняю в файл 01.xml (Scan → Save Scan). Далее, как было описано, с использованием программы Ncat запускаю http-прокси на локальном порту 8080 и снова провожу сканирование. Результат сохраняю в файл с названием 02.xml. Для сравнения результатов сканирования выбираю команду из главного меню Zenmap Tools → Compare Results. В поле A Scan открываю файл 01.xml, а в поле B Scan – файл 02.xml. Всё, программа автоматически осуществит сравнение файлов.

На рис. 2 видно, что произведено сравнение двух результатов сканирования. Для наглядности основные моменты или изменения первого сканирования подчеркиваются ро-

Рисунок 1. Завершение установки Nmap 5.00

```
       If it is installed you can try the --with-openssl=DIR argument
checking pcap.h usability... no
checking pcap.h presence... no
checking for pcap.h... no
configure: creating ./config.status
config.status: creating Makefile
config.status: WARNING:  'Makefile.in' seems to ignore the --datarootdir setting
config.status: creating config.h
  ( )   /\ _                 (
   \ |  (  \ ( \.(            )                    _____
  \  \ \  `  `   ) \          (  ___               /     \
 (_`    \+   . x  ( .\          \/   \____-----------/ (o)   \_
- .-               \+  ;        (  O                          \____
                     )          \_____________ `                  \  /
(_                +- .( -'.- <. - _  VVVVVVV VV V\                 \/
(____           ._._: <_ - <- _   (--_AAAAAAA__A_/                  |
  .    /./.+-  . .- /  +--  - .     \_____________//_               \_______
  (__ ' /x  / x _/ (                                 \___'          \     /
 , x / ( '  . / .  /                                     |            \   /
   /  /  _/ /    +                                      /              \/
    '  (__/                                           /                 \
          NMAP IS A POWERFUL TOOL -- USE CAREFULLY AND RESPONSIBLY
Configuration complete.  Type make (or gmake on some *BSD machines) to compile.
```

Рисунок 2. Сравнение результатов сканирования с использованием Zenmap



зовым цветом, а второго – светло-зеленым. На этом примере видно, что программа нашла различия – во время второго сканирования был открыт порт 8080 (8080/tcp open http-proxy – результат использования утилиты Ncat, упоминавшийся выше), а количество закрытых портов соответственно уменьшилось на один – 98.

Еще одним новшеством стало увеличение производительности работы Nmap, а именно скорости сканирования. Была проведена работа по выявлению наиболее часто используемых портов путем сканирования десятков миллионов IP-адресов в Интернете. Теперь Nmap при сканировании по умолчанию обращается, как показано на рис. 3, к 1000 из них. Что значительно увеличивает скорость сканирования. Быстрое сканирование Nmap (с опцией -F) теперь проводится с обращением к 100 часто используемым портам. В Nmap 4.68, например, проводилось сканирование 1276 TCP-портов и 1017 UDP-портов.

Выпуск книги Nmap Network Scanning (http://nmap.org/book) разработчики также отнесли к значительным изменениям, сопровождающим выход Nmap 5.00. По их заявлениям, эта книга подойдет как новичкам, так и профессионалам в области сетевой безопасности (от основ сканирования до разработки специальных методов, в том числе изучение сети, администрирование, аудит безопасности и т.д.). Например, рассматриваются вопросы обхода брандмауэров и систем обнаружения, оптимизации производительности и автоматизации задач с использованием скриптов (о скриптах более подробно будет сказано ниже). В книге приведены все опции программы, примеры использования которых показаны во второй части. Более половины печатной версии доступно в онлайн-редакции издания по адресу http://nmap.org/book/toc.html. Интересно, что книга даже возглавляла списки продаж компьютерной литературы на Amazon (http://nmap.org/book/img/nns-top-seller-942x1024.png) [4].

Заключительным, пятым, наиболее значительным изменением в Nmap версии 5.00 стало усовершенствование NSE (Nmap Scripting Engine, скриптовый движок Nmap). NSE – это один из самый мощных и гибких инструментов сканера безопасности. Он дает возможность писать скрипты с целью автоматизации большого круга сетевых задач. Разработчиками было произведено улучшение существующих сценариев (оптимизация работы и т.д.) и добавление 32 новых, а общее количество скриптов выросло почти на 50% – до 59. Все скрипты имеют хорошую документацию, которая находится по адресу http://nmap.org/nsedoc.

Рон Боуз расширил количество библиотек NSE, добавив еще шесть – smb, smbauth, netbios, msrpc, msrpcperformance и msrpctypes, а также 14 новых скриптов: p2p-conficker, smb-brute, smb-check-vulns, smb-enum-domains, smb-enum-processes, smb-enum-sessions, smb-enum-shares, smb-enum-users, smb-os-discovery, smb-pwdump, smb-security-mode, smb-server-stats, and smb-system-info. Кроме того, стоит отметить, что благодаря двум из этих скриптов (smb-check-vulns и p2p-conficker) Nmap стал первым сканером, который удаленно обнаруживает червь Conficker.

Другие добавленные скрипты позволяют, например, подключаться к MySQL-серверу и выводить информацию о номере версии, статусе СУБД, возможностях и паролях (mysql-info); попытаться войти в учетную запись электронной почты (POP3) путем подбора имени пользователя и пароля (pop3-brute); получать информацию о возможностях POP3-сервера электронной почты (pop3-capabilities); проверить, уязвим ли веб-сервер для атак на получение доступа к /etc/passwd с применением методов обхода каталогов ../../../../etc/passwd (http-passwd).

Новые возможности Zenmap

Zenmap представляет собой официальный графический front-end для Nmap. Например, он позволяет легко сохранять результаты сканирования в файл (как было показано выше), что для часто используемых типов сканирований значительно облегчает жизнь ИТ-специалиста.

Zenmap может выступать в роли графического вьюера результатов сканирования. Хотя название Nmap (Network Mapper) – сетевой картограф, он до сих пор не мог рисовать карту сети. Новая функция Zenmap позволяет создавать интерактивную анимационную визуализацию хостов в сети и соединений между ними. Источник на такой карте, с которого производится сканирование, всегда располагается в центре, а узлы – на концентрических кругах. Линии между узлами соответствуют обнаруженным связям.

Рассмотрю картографирование сети с использованием Zenmap подробнее. Запускаю программу и в поле Target набираю через пробел:

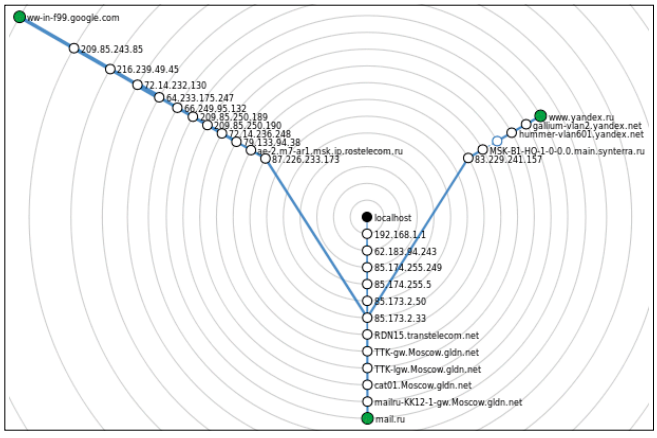
Рисунок 3. Nmap сканирует по умолчанию

```
root@ubtlnx:~# nmap 127.0.0.1

Starting Nmap 5.00 ( http://nmap.org ) at 2009-08-10 12:24 MSD
Interesting ports on localhost (127.0.0.1):
Not shown: 999 closed ports
PORT    STATE SERVICE
631/tcp open  ipp

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
root@ubtlnx:~#
```

Рисунок 4. Zenmap в роли сетевого картографа



```
www.yandex.ru www.google.com www.mail.ru
```

Далее в поле Profile выбираю Quick traceroute. В итоге в поле Command получается такая команда:

```
nmap -sP -PE -PS22,25,80 -PA21,23,80,3389 -PU -PO ↵
     --traceroute www.yandex.ru www.google.com www.mail.ru
```

Затем нажимаю кнопку Scan. Через несколько секунд Zenmap завершает выполнение команды, а я перехожу на вкладку Topology. Нажимаю кнопку Controls для управления отображением карты. В правой части окна программы отобразится боковая панель. Она позволяет сохранить изображение с сформированной картой (в формат .PDF, .PNG и другие), управлять элементами, отображаемыми на карте (IP-адрес, имя хоста, круги и т.д.), размером карты (Zoom) и растоянием между кругами (Ring gap). Далее я изменил параметр «Lower ring gap» (минимальное расстояние между кругами) для того, чтобы вся карта была на одном экране, и сохранил ее в файл, изображение из которого представлено на рис. 4. Кстати, щелкнув на любом хосте на карте, можно поместить его в центральный круг, а отображение остальной сети будет выстроено относительно этого хоста.

В Zenmap имеется функция агрегирования результатов сканирования, что дает возможность на одной карте представить несколько результатов сканирований. Например, если бы я проводил трассировку в приведенном выше примере сначала для www.yandex.ru, затем для www.google.com и www.mail.ru, то итоговый результат картографирования был такой же, как в указанном примере. Кроме того, переботан стандартный набор профилей сканирования с целью сделать его еще более удобным в работе. Если вдруг вам ни один из профилей не подойдет, можно с помощью редактора профилей (Profile → Edit Selected Profile) создать свой. Также разработчики добавили в Zenmap функцию расширенного поиска по результатам сканирования. Так, можно найти предыдущий результат сканирования с помощью таких критериев, как: порты, которые были открыты; ключевые слова в имени операционной системы, выявленной в ходе сканирования. Для начала поиска надо нажать клавиши <Ctrl>+<F> или выбрать в меню команду Tools → Search Scan Results.

***

На основании рассмотренного материала можно сделать вывод о том, что внесенные усовершенствования сделали Nmap еще более важным средством для исследования и обеспечения безопасности компьютерных сетей, которое значительно облегчает работу ИТ-специалистам, причем как начинающим, так и опытным. Большое количество изменений, улучшение производительности и развитие графического интерфейса Zenmap, а также свобода и кросплатформенность данного инструмента устанавливают высокую планку требований, которые будут предъявляться к подобным проектам. EOF

1. http://nmap.org/5/#changes.
2. http://www.xakep.ru/post/20972/default.asp.
3. http://nmap.org/ncat.
4. http://nmap.org/book.

Свободный полет

Если человек не боится жизни, проявляет искренний интерес ко всему, он становится гармонично развитой личностью

Оксана Родионова

Мне жизнь Артема Хазова, директора по продажам компании «Афлекс софтвер», видится как долгое путешествие воздушным транспортом. Аэропорты, яркое синее небо, низкая облачность, взлеты и посадки. Воспоминания всплывают и отходят в прошлое, как пейзажи под крылом самолета. А прелесть полета – в необыкновенном чувстве свободы.

Родное гнездо

...Пролетая над Уральскими горами, Артем Хазов не спутает их ни с какими другими. С любой высоты разглядит. Ведь здесь, на стыке Европы и Азии, лежит родной город с красивым именем Златоуст. Даже произносить приятно. Пусть климат не ахти – резко континентальный (зима продолжается до бесконечности, а лето прохладное), пусть экология хромает, зато природа – красота! Артем объясняет:

– Город наш находится прямо в горах. Выходишь на улицу, едешь на трамвайчике, а кругом горы, горы, горы. Кстати, я занимался в свое время альпинизмом, спелеологией. Лазить по пещерам и по горам там просто наслаждение. Уральские горы, конечно, не очень высокие, старые, но все же. Поскольку Златоуст находится в низине, а вокруг него горы, все тучи, которые проходят над ним, частенько зависают. Сыро, как в Лондоне.

А природа замечательная. Выходишь из родительского дома – сразу лес начинается. В детстве, конечно, часто в походы ходил.

Родители Артема поженились сразу после возвращения отца из армии. Через годик, когда родился сынок, папа с мамой уже учились на вечернем отделении в институте и работали на заводе. Первое время воспитывали мальчика бабушка с дедушкой. Отучились – стало полегче.

– Обычные инженеры, – улыбается Артем. Уточняет:

– Инженеры-технологи. Помню, дома постоянно чертежи были, родители спорили часто на производственные темы. Мама, например, всю жизнь отработала на одном заводе, практически чуть ли не в одном отделе.

Атмосфера дома была как в хорошем советском фильме, где родители всегда воспитывали своих детей собственным примером. В наше время такому можно только по-доброму, искренне позавидовать.

Прохлаждаться без дела мама с папой не любили, и сыну не давали. Детство у Артема было весьма насыщенное, о чем он не жалеет. Вспоминает о тех годах как о сплошном полете: «Скорее, скорее, успеть, не опоздать!»

– Весной того года, когда я должен был пойти в школу, у меня родился младший братишка. И я очень быстро повзрослел, потому что было много забот. Помню, днем возвращаюсь из школы, благо она была недалеко от дома, а мама уже в окошке меня выглядывает: «Артем, на «молочку»!». С банками-склянками бегу через дорогу на молочную кухню, возвращаюсь, отдаю: «Кушай, братишка!». Хватаю сумку со спортивным обмундированием и бегу на фигурное катание. А еще я плаванием занимался. Плюс музыкальная школа. Времени свободного не было абсолютно. Яркое воспоминание – возвращаемся с отцом с тренировки, зимний вечер, уже темно. Иду через двор, коньки висят на плече, а друзья в хоккей играют. Кричат: «Артем, выходи!». – «Не могу, я еще уроки не сделал...». Было обидно? Да нет, в общем-то, мне такая жизнь нравилась, она насыщенная была. Очень важно, что мама никогда не заставляла меня чем-то заниматься насильно. Приводила в секцию или в музыкалку и оставляла, только если мне нравилось.

В музыкальной школе Артем посвятил пять лет игре на аккордеоне, даже на областное телевидение в Челябинск как-то возили в компании с другими одаренными детьми. Но планы поменялись, а музыкальное образование осталось.

Чтобы вырастить способного человека, ему надо давать свободу действий. Немножко контролировать, однако не запрещать ему ничего

Теперь Артем очень благодарен родителям, что именно таким было его детство. Ведь его сверстники нынче не только высокие посты занимают, среди них есть и опустившиеся люди, наркоманы. Не хватило закалки. А у Артема она была и есть.

– Чтобы вырастить способного человека, ему надо давать свободу действий. Немножко контролировать, однако не запрещать ему ничего. Он должен сам выбирать. И слава богу, что у меня была такая свобода выбора. Мама, видимо, наблюдала, что во мне есть, и давала возможность правильно развиваться. А на глупости всякие времени не оставляла.

Блистающий мир

...Полет продолжается, но самолет повернул, и теперь летит встречать восход. Небо алеет. Прекрасно наблюдать из иллюминатора, как встает солнышко. Мир выглядит, наверное, так, как в первый день творения...

В старших классах Артем решил, что будет творить себя уже как технарь. Он перевелся в школу №10, где был физико-математический класс. Параллельно занимался в заочной физико-технической школе при Московском физико-техническом институте. Добросовестно делал задания, которые ему присылали по почте, и отправлял в столицу.

Жизнь, как всегда, немного подкорректировала намерения и желания – наградила искушением. Дело в том, что в новой школе был интереснейший клуб самодеятельной песни «Гринландия». Вот так и пишется, через «и», это не опечатка. Потому что вдохновил его создателей знаменитый писатель-романтик Александр Грин, автор «Алых парусов» и «Бегущей по волнам». Клуб был создан в 80-м году, тогда же и первую награду получил, на городском юношеском фестивале. Так и шло – пели бардовские песни, сочиняли свои, выступали, становились лауреатами городских, областных, всероссийских фестивалей, даже знаменитого Грушинского. Ставили спектакли, только особые, с музыкой, с песнями. И какие спектакли! «Тень» по Шварцу, «Король-олень» по Гоцци, «Снежная королева» по Андерсену. Как было устоять Артему, когда приглашают в такой коллектив. Он был парень скованный, замкнутый (в это сегодня трудно поверить!), но пошел с удовольствием.

Наверное, во многом благодаря учителю, ведь все в конечном итоге решает личность. Филолог, заслуженный учитель России, почетный гражданин Златоуста Наталья Анатольевна Галавтина, которая много лет руководила «Гринландией», была не только городской поэтессой, но и математиком по образованию. Можно сказать, мистическое совпадение: и у Артема в судьбе объединились физика и лирика.

Свою единственную, но значимую роль в спектакле по пьесе Григория Горина «Тот самый Мюнхгаузен» Хазов помнит до сих пор. Он играл

Артем Хазов, директор по продажам компании Aflex software – официального представительства западных разработчиков программных решений на территории России и СНГ. Окончил МИФИ в 2003 году. В компании Aflex software работает с момента её основания, начинал с должности менеджера по работе с партнёрами.

героя-любовника, адвоката Рамкопфа. Эту роль в одноименном фильме исполняет Александр Абдулов. Помните, в сцене развода он в качестве защитника баронессы заявляет, осуждая Мюнхгаузена: «Господа! Всякий муж, возвращаясь домой после недельного отсутствия, пытается обмануть жену, но не всякий додумается до того, чтобы утверждать, что он был на Луне!» С этим спектаклем, кстати, ребята стали лауреатами Всероссийского юношеского фестиваля в Димитровограде в 1996 году.

Конечно, и роль, и песни, и вся атмосфера КСП наложили отпечаток на старшеклассника Артема Хазова. Он хотел одно время поступать в театральный, даже после того, как стал студентом совсем другого вуза. Не решился. Но, во всяком случае, вкус к хорошей музыке, поэзии остался на всю жизнь. И любовь к творчеству. И гимн «Гринландии», который написала Наталья Анатольевна:

Отвергая сказки и мечтания,
Жизнь проходит взрослая моя,
Но порой опять страна Гринландия
Позовет в нездешние края.

На сайте школы есть страничка, посвященная легендарному КСП, и если заглянуть в список «гринландцев», в нем легко можно отыскать и Артема Хазова.

Зона турбулентности

...Школа осталась позади. Нет, самолет не вошел в штопор. В таких случаях командир корабля объявляет по громкой связи: «Мы входим в зону турбулентности». Еще бы, сменить пусть насыщенную, но провинциальную жизнь на ритмы столицы – это серьезная перемена. Потряхивает...

Хазов летел в Москву как на крыльях. Он студент Московского инженерно-физического института! Раньше, когда гостил в столице, гулял, сдавал экзамены, все радовало, притягивало. Казалось, даже запахи здесь другие, привлекательные, особые. Хотелось наконец почувствовать себя частью огромного города. Была еще одна причина, по которой выбрал МИФИ, а не уральский вуз.

> Я долго и переживал, почему у меня нет работы, от которой бы получал удовольствие. А сейчас я это даже работой не могу назвать, это часть моей жизни. Если делаешь то, что приносит удовольствие, то делаешь это хорошо

– У меня был пример – мой старший брат. Двоюродный, но он мне как родной. Я ему в детстве подражал. Он тоже окончил московский вуз, причем сходного профиля, МФТИ. В музыкалку, как я, ходил и на аккордеоне играет. Он для меня всегда был кумиром.

– Какое было ощущение, когда вы приехали в Москву?

– Когда я поселился в общежитии, вкусил все прелести Москвы, стало мне не по себе. Все чужое. Первые полгода я рвался обратно. В Златоусте остались друзья, любимая девушка, дом, все привычное. Что делать? Общежитие – маленькая комната, еще три соседа там живут и кот... А сейчас приезжаю к родителям и мне не хватает этого драйва, ритма. День я еще могу посидеть, в лесочек сходить, родственников обойти. Но на второй день мне уже надо куда-то бежать. Начинаю думать: «Мне в Москву надо, у меня дела срочные. Почему мне не звонят? Где мой ноутбук?».

Тогда, в студенческие годы, все постепенно вошло в свою колею. Общага, лекции, экзамены. Физикой Артем увлекался с детства, поэтому ему было интересно учиться.

– Я считаю, что на физику не надо смотреть как на набор формул. Это предмет, который объясняет жизнь, существование всего в нашем мире. Мне, слава богу, в жизни повезло. В вузе общался с одаренными и умными людьми. Например, зав. кафедрой квантовой электроники был Николай Басов, тот самый, который в 1964 году вместе с Александром Прохоровым и Чарлзом Таунсом получил Нобелевскую премию по физике за «фундаментальную работу в области квантовой электроники». В конце 50-х годов они сделали первый лазер. Я общался с их сокурсниками, с подобными людьми. И знаете, они гораздо больше, чем физики, у них огромный жизненный опыт, очень многому можно научиться, не только науке.

– Физика – это ведь не только расчеты, она лишь использует математику как инструмент?

– У нас есть такие задачи, которые без формул решаются. Нужно просто понять тот или иной процесс, как это происходит. Мне профессор на вступительном экзамене задал задачку. Есть два стакана, до краев наполненных водой, и есть листок бумаги. Как с помощью листка поставить стакан на стакан и чтобы один не провалился в другой? Тут никакие формулы не помогут. Задача на соображение.

– И какой ответ?

– Все очень просто. Нужно сложить бумажку гармошкой, ребра жесткости появляются. Положить на один стакан, а сверху поставить второй, он будет держаться. Это моя любимая задачка. Нужно просто немножко подумать.

Может быть, это закономерность такая – если человек летит по жизни, не прячется от ее сложностей, проявляет искренний интерес ко всему вокруг, он и становится тем, что называется «гармонично развитая личность»? А может быть, дело в физике? Ведь известно, что многие успешные физики были творческими личностями, литераторами и стихотворцами. Хотя и немногословных лабораторных затворников среди них немало. Наверное, все-таки многое зависит от конкретной личности. В ней – тайна.

Безоблачное небо

...Ведал или нет Артем, что «зона турбулентности» в его судьбе продлится и после окончания вуза, неизвестно. Но он стойко выдержал все напасти судьбы, несмотря на трудные

90-е годы, первый дефолт, кризис. Учился многому на практике, пробовал себя в разных видах бизнеса. Искал свое дело...

– Сначала не так гладко все было. Приходил на собеседования в разные компании, там смотрели на стаж, а нормального, официального стажа у меня не было, несмотря на то, что подрабатываю с первого курса. Я долго искал занятие по душе. Даже на американской фондовой бирже пытался работать по вечерам в Интернете. Было интересно, но по деньгам выходило не очень, поскольку работал на «дядю».

– Вам важно, чтобы работа была интересной? Но ради денег-то можно потерпеть и скучное дело?

– Не могу долго заниматься тем, что мне не нравится. Необходимость, конечно, некоторая существует, чтобы заработать денег, чтобы на хлеб с маслом хватало, но постоянно… не выдержу. После фондовой биржи трудился в новосибирской компании «Топ-книга». Они в Москве открыли сеть магазинов. Должность у меня была – менеджер отдела развития, я открывал книжные магазины, сейчас по Москве их много, у них разные лейблы, есть гипермаркеты «Ласт-Книгас», например. Это мое детище (улыбается). А в 2006 году меня пригласили в «Афлекс», его как раз организовали. «Афлекс» – это сейловое продажное подразделение нескольких вендеров. Сейчас планируется из «Афлекса» сделать самую лучшую продающую компанию в России. Мне кажется, правильно, когда ставятся такие высокие задачи, это стимулирует.

– Чем вас заинтересовала эта компания?

– Если честно, изначально я не понимал, куда и на что меня приглашают. Позвали менеджером по продажам – я пошел. Получится – не получится. Получилось. Я так понимаю, что ведение бизнеса, прогнозирование продаж и так далее – это важно, но если человек не умеет общаться с людьми, находить контакты, ничего не выйдет. Главное в моей работе – умение общаться с абсолютно разными людьми. С кем-то можно сразу найти общий язык, с кем-то наоборот. Но достучаться нужно до каждого. Я пытаюсь до сотрудников, менеджеров своих донести, что очень важно неформальное общение с людьми. Мы часто ездим на семинары, рассказываем о продукте. Самое сложное – подойти к человеку после семинара, начать с ним общаться. И запомниться ему не только тем, что ты хорошо знаешь свои продукты, но и тем, что можешь поговорить на любую тему, проявить искренний интерес к собеседнику. Чтобы с тобой человеку было приятно выпить чашку кофе.

– А если человек не хочет общаться?

– Надо прислушиваться к своей интуиции. И быть готовым к тому, что встреча не оправдает твои ожидания. Готовишься к одному, а получается все совершенно по-другому. Надо пообщаться с человеком, посмотреть ему в глаза, понять, какой он, и почувствовать, что с ним в данный момент происходит. Может быть, он просто себя сегодня плохо чувствует, зуб болит. Поговорить с ним. Достаточно нескольких фраз, чтобы более или менее понять, что ему интересно, и на эту тему начать с ним разговор.

– Вы изучаете психологию, читаете специальную литературу?

– Нет. Много полезного нахожу у наших классиков. Абсолютно все у них там написано между строк. Перечитываешь того же Довлатова, Пастернака (это мой самый любимый поэт и писатель), Достоевского, Гоголя. Кстати, Николая Васильевича в школе последний раз читал. И тут вдруг начал перечитывать «Мертвые души», и все совершенно по-другому стал воспринимать.

Книги, театры (самый любимый – Ленком), шашлыки с друзьями, общение с одноклассниками, многие из которых тоже перебрались в столицу, – это все редкие свободные минуты, сейчас их все меньше и меньше. Времена не те – надо выживать, держать фирму на плаву. Но все равно, даже кризис не может отменить ощущение счастья, которое не покидает Артема Хазова.

– Я долго искал, очень мучался и переживал, почему у меня нет той работы, от которой бы я получал удовольствие. А сейчас я это даже работой не могу назвать, это часть моей жизни. Утром просыпаюсь с мыслью о ней и засыпаю с этим. Постоянные семинары, выезды, общение – что называется, «в кайф». А я с детства привык: если делаешь то, что приносит удовольствие, то делаешь это хорошо.

Похоже, в жизни ничего не бывает просто так. Пролетая мысленно по воспоминаниям детства, ранней юности, можно многое о себе понять. Может, не было бы сейчас успешного директора по продажам компании «Афлекс софтвер» Артема Хазова, если бы когда-то мама не отдала его на фигурное катание, где волшебным образом соединяются тренировки до седьмого пота и искусство, если бы не учился он математике и физике, если бы не играл в школьном театре. Оказалось, все это нужно было, чтобы сегодня почувствовать себя счастливым человеком. Человеком, который любит летать.

– Когда долго нет поездок, я смотрю в небо и думаю: «Когда же я полечу?». Мне прямо не хватает чего-то. Причем для кого-то долгие перелеты на Дальний Восток – это кошмар, а мне нравится. Однажды летели из Хабаровска в Москву, практически всю Россию пролетели, на небе не было ни облачка, видны все речки, леса, города – вся земля, очень красиво! Я сидел, прилипнув к иллюминатору. Но когда подлетали к Москве, конечно, появился смог...

Ничего, смог – это ненадолго. Обязательно будет новая командировка и новый полет, который опять подарит это удивительное чувство свободы. EOF

Справка «СА»

ООО «Афлекс софтвер» – официальный представитель ведущих мировых разработчиков программного обеспечения. Компания тесно сотрудничает с зарубежными производителями программного обеспечения, представляя полный комплекс услуг по продвижению программных продуктов и представлению их интересов на территории Российской Федерации и стран СНГ. Сейчас партнерская сеть Aflex software насчитывает 300 компаний в регионах России и стран СНГ. Стратегическая цель Aflex software – развитие как бизнеса вендоров на российском рынке, так и всего рынка программного обеспечения в целом, через продвижение наиболее востребованных решений, ориентированных на частных пользователей и корпоративных клиентов.

Визитка

АРТЕМ ЧЕРНЕВСКИЙ, руководитель службы предпродажной поддержки по решениям Microsoft. Консультирует по ИТ-решениям, проводит тренинги для руководителей, имеет опыт руководства различными ИТ-компаниями. MCSE, Comptia A+Certified

Ищем сотрудников или партнеров?

Российские аналитики не раз прогнозировали, что собственные ИТ-департаменты будут заменены специализированными ИТ-компаниями. Но «завтра» наступит не скоро

Финансовый кризис вызвал очередную волну интереса к данной задаче, так как вынужденное сокращение ИТ-бюджетов побуждает искать наиболее экономичные способы поддержания ИТ-инфраструктуры компаний. Однако до сих пор нет ни понимания обсуждаемого вопроса, ни даже устоявшейся терминологии (см. врезку «Определения»).

Преимущества и недостатки ИТ-аутсорсинга

Так все-таки ищем сотрудников или партнеров? Начнем с экономики. На первый взгляд все просто, берем зарплату потенциального сотрудника, делим на 160 (примерное количество рабочих часов в месяц) и сравниваем с предложением партнера. Если для Москвы ставка опытного специалиста ИТ-поддержки начального уровня составляет около 50 тыс. руб., то получим 50 тыс. руб./160 ч. = 312 руб. 50 коп. в час. Но средняя ставка компании аутсорсера (не студента на вольных хлебах) – от 1500 до 3000 руб. за час. Минимум в пять раз больше. Жадность? Не торопимся с выводами. Посчитаем чуть внимательнее. Во-первых, существуют праздники, отпуска и больничные. Из 365 календарных дней в году рабочих – всего 225. Аутсорсеру мы платим только за фактическую работу, своему сотруднику же оплачиваем весь период. Во-вторых, существуют налоги на заработную плату – точную цифру назвать сложно, зависит от финансового управления вашей компанией, но в среднем около 30% от ставки. В-третьих, сотруднику по КЗоТ нужно 6 кв. м офисного пространства, да и компьютер, и ПО тоже необходимы. Кадровое агентство берет за свои услуги где-то 10% от годовой зарплаты за первый год работы сотрудника. Если вы ищете его сами, это может оказаться и дороже (время руководителя и HR-специалиста). Внутренняя цена вырастает минимум до 1000 руб. в час. Но все-таки непонятно, почему аутсорсинг дороже?

Все вышеизложенное оказывает сравнительно небольшое влияние на внутреннюю стоимость часа специалиста по сравнению с ключевым параметром сервисных бизнес-направлений: утилизацией. Ошибочно считать, что время, когда ваш сотрудник не играет в «Сапера», не висит в личной почте и не пьет кофе, он тратит на работу.

Первая проблема – самообучение. Если один специалист отвечает за множество различных систем, он должен учиться каждой из них, причем не только изучать литературу, но и экспериментировать с установкой, настройкой и поддержкой работоспособности (обучение в процессе работы). Специалист профильной компании может позволить себе узкую специализацию и высокую повторяемость задач, тем самым сильно сокращая время обучения. А что самое главное – вы не платите за то время, пока специалист аутсорсинговой компании учится.

Вторая проблема – управление проектами. Вряд ли весь бизнес вашей компании будет подстраиваться под ритм работы ИТ. Скорее наоборот. Это вызывает «процедурные» простои в работе ИТ-шников, когда они ожидают информацию от других подразделений, или поставку продуктов, или удобное «окно» в бизнес-процессах для выполнения работ. А вы все это время платите им зарплату. Чтобы оценить масштаб проблемы, можно привести следующую цифру: в консалтинговых подразделениях крупных западных вендоров хорошим уровнем утилизации является всего лишь 80%.

Точную оценку средней утилизации внутренних ИТ-специалистов по стране дать сложно, но, думаю, она должна быть в диапазоне от 10 до 40%. Таким образом, стоимость эффективного часа внутреннего специалиста – от 2 500 до 10 000 руб.

Кстати

Не так уж и мал рынок аутсорсинга, как следует из отчетов различных аналитических агентств. Ведь они опрашивают в основном только 500 крупнейших системных интеграторов и компаний-заказчиков (преимущественно обладающих своим ИТ-штатом). Оценить рынок аутсорсинга в малых и средних компаниях (зачастую «черный» и даже юридически неоформленный) невозможно, но это вовсе не значит, что он мал. Вспомните, какое огромное количество студентов физмата легко находят себе подработку в этой области, в десятки и сотни раз больше, чем проходящих стажировку в вендорах и крупных системных интеграторах.

Сокращение ИТ-бюджетов побуждает искать экономичные способы поддержания ИТ-инфраструктуры

Поразительный вывод, с учетом того, что на руки он получает всего 50 000 руб. При этом мы не обсудили качество выполняемой работы и вероятность успешного завершения проектов внедрения. Очевидно, что у компании, ранее успешно выполнившей ряд аналогичных проектов, качество и вероятность успеха значительно выше, чем у только что созданного внутреннего ИТ.

Почему же рынок аутсорсинга так мал?

Основная проблема – высокие профессиональные требования и к заказчику, и к исполнителю. Проблема исполнителей – отсутствие обязательного лицензирования деятельности, очень низкая стоимость стартапа своего бизнеса и огромное количество студентов физмата на рынке труда приводят к появлению большого числа ИТ-фирм-однодневок. Они зачастую не выполняют взятые на себя обязательства. Размер компаний и слабость нашей законодательной базы не дадут заказчику взыскать с исполнителя компенсацию понесенных убытков. Да и за счет небольшого времени существования ИТ-компаний они не могут получить преимущества от многократной повторяемости типовых операций.

У нас практически нет вузов, которые готовят специалистов в области управления ИТ-проектами и организации ИТ-бизнеса. Усугубляет ситуацию и то, что коллектив талантливых технарей редко может создать устойчивый бизнес.

С заказчиками тоже не все просто. Свой специалист – «раб», послушно бегущий в нужном направлении. Сменилось направление – бежит в другую сторону и он. У меня был случай, когда ИТ-директор крупного холдинга экстренно покинул важную встречу, так как ему позвонили и сказали, что завтра их холдинг открывает новый филиал на 200 рабочих мест. Первые сотрудники должны выйти на работу в течение недели (без утвержденного дополнительного бюджета, СКС в здании, компьютеров и договора с провайдером). Вариант с аутсорсером требует четкого планирования работы и бюджета, а также знания всех числовых ключевых параметров собственного бизнеса. Чем, к сожалению, может похвастаться не каждый российский руководитель.

ИТ-аутсорсинговая утопия

Как бы выглядела экономически обоснованная модель ИТ-инфраструктуры для компании, скажем, в 300 рабочих мест, разбросанных по трем городам и пяти офисам? Нашлось ли бы там место ИТ-аутсорсингу? Давайте посчитаем.

Детальная проработка задачи

Ясно, что 300 рабочих мест очень разнородны по выполняемым задачам. От типовых операций в «три кнопки» до многоцелевого использования компьютера для коммуникаций, переговоров и аналитической работы. Классический подход – всем все одинаково и по высшему разряду. Очевидно, что дорого и избыточно. Давайте выделим 40-50 «продвинутых» пользователей – руководство, аналитики и т.д. Остальным нужен только самый базовый функционал – офисные приложения, почта, Интернет, специализированное ПО. С точки зрения отказоустойчивости и доступности данных первая группа, очевидно, имеет приоритет.

Можем ли мы использовать хостинг?

Формально нет, на 300 станций свои серверы дешевле, но если мы учтем, что 200-250 пользователей могут обой-

Претендентов много, профессионалов мало

Профессионализм, высокая ответственность и исполнительность сотрудников – основные факторы, на которых держится стабильность бизнеса по поддержке ИТ-инфраструктуры заказчика. Очень сложно найти специалистов-профессионалов, которые обладают разносторонним опытом внедрений и поддержки десятков или даже сотен проектов, способностью быстро и точно вникать в суть бизнеса заказчика, а также могут поддерживать и развивать различные проекты. Большинство кандидатов, считающих свой профессиональный уровень достаточным для работы в аутсорсинге, приходят с опытом работы штатным системным администратором в различных фирмах, однако при проверке «в боевых условиях» часто показывают полную неспособность самостоятельно решать проблемы, разбираясь в них досконально. Помимо этого, очень немногие специалисты готовы работать в разъездном графике.

Роман Марков, директор компании ООО «Оптимум», г. Санкт-Петербург

Определения

Аутстаффинг – термин и идея пришли из области управления кадрами. Многие компании для поддержания имиджа предлагают большой соцпакет, арендуют дорогостоящие бизнес-центры, несут большое налоговое бремя, не жалеют средств на административные расходы. Некоторые западные компании при этом еще имеют и ограничения на количество штатных сотрудников в офисе. Выход предлагают специализированные компании, трудоустраивающие у себя специалиста и на основе годового контракта передающие сотрудника в оперативное управление заказчику. Формат работы практически не меняется, но подход позволяет сократить накладные расходы на персонал. На рынке есть ряд ИТ-компаний именно с таким подходом к поддержке заказчиков. В чем их преимущества перед традиционными аутсаффинговыми компаниями широкого профиля? Первое преимущество – помощь в подборе персонала. ИТ – специализированная область, поэтому оценить кандидата могут только профессионалы. Второе – обучение, особенно в процессе работы. Третье преимущество – легкая замена даже редкого специалиста на время его болезни или отпуска. Минусы подхода – невысокий процент экономии (заметный только в компаниях с известным брендом), а также низкая мотивация «контрактников» по сравнению с постоянным персоналом заказчиков. Большое распространение услуга вряд ли получит, так как в кризис накладные расходы на персонал были сокращены, экономический эффект модели сократился.

Техподдержка – подразумевает реактивную поддержку специалистами аутсорсера возникающих технических проблем заказчика и, возможно, проведение профилактических работ. Оплата проводится на основе фиксированной ежемесячной суммы, стоимости использованных часов специалистов или комбинации этих параметров. Самый распространенный и, к сожалению, один из самых неэффективных способов взаимодействия. Минусы – запредельно высокая стоимость, замедленная реакция в случае нестандартных запросов (требует отдельного согласования стоимости, выделения бюджета, длинная цепочка согласования). Главный недостаток этого подхода – фатальный разрыв между задачами ИТ и задачами бизнеса заказчика.

Аутсорсинг. Ключевое отличие от техподдержки в том, что компания-подрядчик отвечает за работоспособность бизнес-функции ИТ. Например, «возможность сотрудников отдела продаж общаться с клиентами компании посредством электронной почты». Обратите внимание: именно не «работоспособность почтового сервера», как в случае техподдержки. Очень хорошо такой подход описывает методология ITIL и частично MOF. Первое преимущество – возможна огромная совокупная экономия, так как ИТ начинает понимать бизнес. (Не будет закуплено самое дорогое решение с избыточной функциональностью, ИТ-специалисты не станут углубляться в технически интересные, но бесполезные для бизнеса области). Второе преимущество – возможность оценить вклад ИТ в бизнес, соответственно спланировать инвестиции и адекватный бюджет ИТ-подразделения. И ключевое достоинство – проактивная позиция ИТ-аутсорсера, являющегося неким ИТ-бизнес-консультантом заказчика, постоянно анализирующим новые технологии с точки зрения полезности для бизнеса клиентов. Пример: небольшое турагентство, 6-8 сотрудников. Заключен контракт на техподдержку с поставщиками компьютеров и ПО. Оплата почасовая, включая время дороги. Буквально за месяцы работы сотрудники этих компаний начинают тихо ненавидеть друг друга. Каждый вызов очень дорог, сотрудники клиента опасаются обращаться к партнеру без острой необходимости (Не проходят в почте большие фотографии отелей и нет удобного способа их уменьшения. Однако операторы турагентства предпочитают фотографии или не отправлять, или отправлять на следующий день, обработав на домашнем компьютере. Очевидна потеря клиентов, вынуждающая увеличивать расходы на их привлечение, затраты ИТ-бюджета в несколько раз!). При этом в случае очевидных сбоев заказчик не может оценить, объективны ли вызвавшие их причины или сам аутсорсер (по злому умыслу или вследствие недостаточной квалификации) их организует. Что бы сделал в данной ситуации грамотный ИТ-аутсорсер? Первое – за счет бекапирования, создания образов дисков, настройки удаленной поддержки, сокращения прав пользователей или других подходов максимально сократил трудозатраты на техподдержку (чтобы сократить свои расходы в первую очередь). Второе – проведя хотя бы полдня в офисе заказчика, обратил внимание на то, что менеджеры плохо знают Excel и тратят лишнее время на проработку заказов. И продал бы руководству клиента идею платного обучения сотрудников Microsoft Office. Обе компании были бы в выигрыше, увеличив свою прибыль. Минусы такого подхода – он требует высокой квалификации сотрудников как заказчика, так и подрядчика.

Хостинг – многие заказчики отказываются от собственного почтового сервера в пользу аренды почтовых ящиков у провайдера. Плюсы – низкая стоимость (для небольших компаний), высокая стабильность работы, доступность из любого места, централизованная система бекапирования. Минусы – низкий функционал решения, ограниченная возможность настройки, неоднозначная безопасность данных. Но в последнее время в связи с развитием предложений по хостингу и классических «корпоративных» серверов можно на хостинге построить для небольшой компании систему с функциональностью, полностью идентичной ИТ-инфраструктуре громадных корпораций. Возможности настройки при этом сохраняются, особенно при использовании сервисов российских хостинг-провайдеров. Единственное – требуется детальное обоснование необходимости изменений и тестирование работоспособности. Сотрудник компании не может по своему желанию бесконтрольно менять настройки сервера. Безопасность данных, пожалуй, основной психологический барьер для распространения этой услуги. Естественно, что непрофессионально настроенный сервер, обслуживаемый низкооплачиваемым специалистом широкого профиля в помещении заказчика, никак не безопаснее размещения данных у провайдера. Да и средств воздействия (особенно коллективом потребителей) на хостинг-провайдера больше, чем на сотрудника вашей компании. Но далеко не все бизнес-руководители это понимают. На горизонте – предоставление услуг хостинга от разработчиков ПО и развитие рынка хостинга клиентских приложений, что делает инвестиции в данный вариант развития ИТ еще более интересными. При этом стоимость услуг специалистов хостинг-провайдера неявно включается в абонентскую плату, что тоже является неким вариантом техподдержки/аутсорсинга.

тись простейшими услугами, то получаем выигрыш в энное количество рублей при использовании хостинга (см. таблицу). При этом кардинально сократим время развертывания инфраструктуры и требования по поддержке.

Какие ИТ-задачи у нас останутся?

Внутренние серверы (скорее всего, налоговые и учетные системы придется оставить в офисе), рабочие станции пользователей, ответы на вопросы пользователей, разработка ИТ-стратегии (связь бизнеса и ИТ), взаимодействие с партнерами. Кстати, немного.

Сколько собственных ИТ-специалистов нам нужно?

Не более трех (высокооплачиваемых!). ИТ-директор должен обеспечивать связь с бизнесом (проводя больше времени в других отделах, анализируя их работу). Его заместитель обеспечит бесперебойность работы (подмена на время отпусков и т.д.) плюс сможет помочь с решением задач на-

чального уровня (известный факт – опытный каменщик и подмастерье работают с той же эффективностью, что и двое опытных каменщиков). Плюс один технический специалист нужен на приемку от исполнителей выполненных работ (организацию приемо-сдаточных испытаний лучше не делегировать, хотя вполне нормально пригласить тестеров низкого уровня другого исполнителя).

С какими компаниями-аутсорсерами нам нужны контракты?

Очевидно, что по каждому функциональному направлению нужно два-три хороших предложения – для заменяемости в случае возникновения проблем. Первое – бухгалтерские и учетные системы, второе – специализированное ПО, третье – обработка обращений пользователей (кстати, все такие службы давно сидят за МКАД – значительно сокращая стоимость), четвертое – оперативное устранение проблем с рабочими станциями (отличная возможность в разных городах привлекать более дешевых региональных партнеров – хостинговая инфраструктура не требует их серьезной подготовки и глубокого знания вашей сети), пятое – учебный центр для пользователей. Отличный вариант – подписание контракта еще и с компанией, занимающейся ИТ-консалтингом – анализ возможностей, предлагаемых динамично развивающимися ИТ-технологиями, для получения конкурентного преимущества в бизнесе компании.

Сколько своих ИТ-шников нам бы потребовалось?

Для упрощения берем также вариант с хостингом, чтобы четко выделить преимущества именно аутсорсинга. В каждом филиале – минимум два (заменяемость) на поддержку рабочих станций и пользователей. Минимум по два (та же заменяемость) на каждое функциональное направление в центральном офисе, а их, как мы помним, пять. Плюс ИТ-директор и заместитель. Получили 22 специалиста. С учетом компенсации кадровой ротации и того, что в столь большом отделе потребуются уже промежуточные руководители – 25-30 специалистов при том же уровне поддержки.

Так сколько мы выиграем на аутсорсинге?

Работа 25 внутренних ИТ-специалистов (средняя ставка те же 50 тыс. руб., налоги, рабочие места) – примерно 2 млн руб. в месяц. Аутсорсинг посчитать значительно сложнее, так как есть огромная разница в цене предложений различных компаний и нет сформированного рынка с определенным прейскурантом, но если примерно оценивать «с запасом», то:

- **Бухгалтерские и учетные системы** – 300 000 руб./мес.
- **Специализированное ПО** – 100 000 руб./мес.
- **Служба поддержки пользователей** – 100 000 руб./мес.
- **Решение проблем с рабочими станциями** – 300 000 руб./мес.
- **Обучение пользователей** – 100 000 руб./мес.
- **Зарплата (с дополнительными расходами) ответственного за приемо-сдаточные испытания** – 100 000 руб./мес.

Итого: 1 млн руб. Чистый выигрыш – 1 млн руб. в месяц или 12 млн руб. в год. Естественно, все вычисления очень грубые и напоминают «среднюю температуру по больнице», но на рынке есть и заказчики с похожими параметрами ИТ-отдела, и исполнители с похожими ценовыми предложениями, так что для определенного сегмента эти вычисления справедливы.

Сравнительный анализ стоимости размещения базовой инфраструктуры

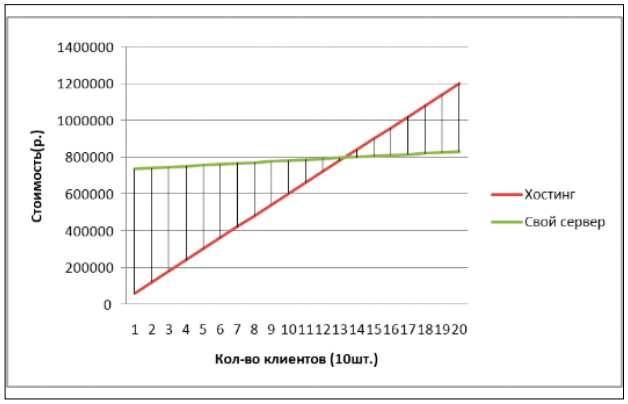
Почтовый сервер Exchange, контроллер домена, портал в случае хостинга или собственного сервера. Приведенные цены могут рассматриваться лишь для целей данного примера, так как стоимость продуктов может очень различаться в зависимости от программы закупки, региона, особенностей компаний, требований к надежности системы:

Инвестиции	Стоимость	Приведенная стоимость в год (с учетом амортизации и т.д.)
Серверное аппаратное обеспечение	300 000 руб.	100 000 руб.
Серверное ПО	150 000 руб.	50 000 руб.
Размещение сервера (вкл. ИБП, стойку, аренду места, кондиционирование и т.д.)	300 000 руб.	100 000 руб.
Фонд заработной платы системного администратора	40 000 руб. в месяц	480 000 руб.
	Итого:	730 000 руб.

При этом стоимость клиентской лицензии в случае своего сервера составляет порядка 500 руб. в год, а в случае хостинга – около 6000 руб.

Рисунок наглядно иллюстрирует, что до 100 рабочих мест при заданных параметрах хостинг оказывается дешевле.

Сравнение стоимости своего сервера и хостинга



Что в итоге?

Все изложенное не является истиной в последней инстанции, приведенные вычисления нельзя переносить на конкретные компании без адаптации к конкретным параметрам. Но в целом механизм анализа прозрачен и прост в использовании. Показанные преимущества ИТ-аутсорсинга говорят о том, что в определенных случаях он является хорошим решением. При каждом планировании ИТ-стратегии, особенно в условиях сокращения расходов, имеет смысл проанализировать данную возможность. И внести свой вклад в становление зрелого, экономически оправданного российского ИТ-рынка.

P.S. Автору было бы интересно услышать ваше мнение о статье на форуме журнала http://www.samag.ru/forum. EOF

Визитка

АНДРЕЙ ЛУКОНЬКИН, ведущий инженер-программист ОАО «НижегородАвтоДор». Занимается автоматизацией производства, бухгалтерского, управленческого и кадрового учета

Электронная проходная своими руками

Надоело записывать в тетрадочку у охраны время прихода и ухода сотрудников? Возьми инициативу в свои руки и автоматизируй этот процесс

Задача автоматизации учета рабочего времени сотрудников – типовая для средних и крупных организаций. Руководителю и отделу кадров важно знать время прихода и ухода, а также время присутствия или отсутствия человека на территории предприятия. Иногда также нужно контролировать время доступа на определенные объекты. Имея данные, можно принимать решения о премировании и депремировании, оплате сверхурочных часов, а также облегчить работу кадровой службы по ведению табелей учета рабочего времени.

Можно купить и поставить «вертушки», прилагающийся софт, закупить магнитные карточки и принтер для печати на них. Но на логичный вопрос руководства: «А почему выходит так дорого?» – можно скромно ответить: «Давайте за пару дней сэкономим несколько десятков или даже сотен тысяч рублей и создадим свою систему». Затраты минимальные: компьютер (подойдет даже старый), сканер штрихкодов (дешевый однолучевой Metrologic, подключаемый в разрыв клавиатуры) и работы по созданию программы и печати пропусков.

Итак, начнем с подготовительных работ. В любой базе «1С:Предприятия» имеется справочник «Физические лица», нам нужно добавить туда строковый реквизит «Штрихкод». Формировать штрихкод можно как вручную, так и автоматически, используя непериодический регистр сведений с одним измерением (сотрудник) и одним ресурсом (штрихкод). Формат штрихкода лучше выбрать самый распространенный – EAN13, начинать его с внутреннего префикса, используемого только на данном предприятии (например, 22). Также нам понадобится регистр сведений для фиксирования времени входа и выхода сотрудников. Назовем его «ПриходУходСотрудников», установим периодичность в пределах секунды и создадим следующую структуру: измерение «ФизЛицо» (справочник физических лиц), ресурсы «Вход» и «Выход», тип данных у которых Булево. Создадим обработку с реквизитами: «Штрихкод» (строка длиной 13), «НайденныйСотрудник» (справочник физических лиц), «СписокПрошедших» (таблица значений). Замысел прост: при сканировании напечатанного уникального штрихкода производится поиск нужного человека, затем проверяется, входит он или выходит. Информация отображается на экране и записывается в регистр сведений (см. рис. 1). В модуле формы обработки размещаем процедуру, срабатывающую при сканировании пропуска с напечатанным штрихкодом.

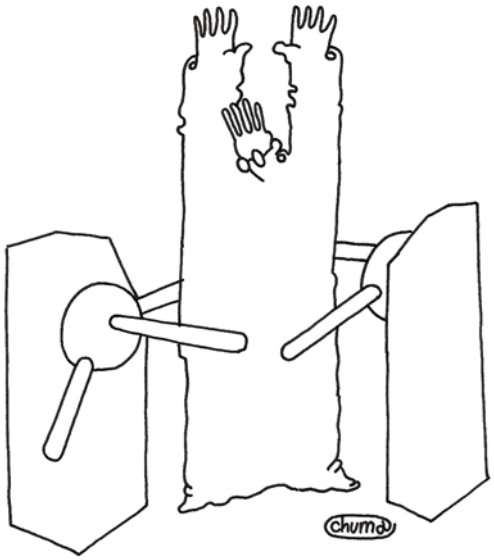
```
Процедура ШтрихкодПриИзменении(Элемент)
    ЭлементыФормы.НадписьОшибкаСотрудника.Заголовок = "";
    ЭлементыФормы.НадписьНайденныйСотрудник.Заголовок = "";

    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   ФизическиеЛица.Штрихкод,
    |   ФизическиеЛица.Ссылка
    |ИЗ
    |   Справочник.ФизическиеЛица КАК ФизическиеЛица
    |ГДЕ
    |   ФизическиеЛица.Штрихкод = &Штрихкод";
    Запрос.УстановитьПараметр("Штрихкод", Штрихкод);

    Выборка = Запрос.Выполнить().Выбрать();
    Если Выборка.Следующий() Тогда
        НайденныйСотрудник = Выборка.Ссылка;
        ЭлементыФормы.НадписьНайденныйСотрудник. ↵
            Заголовок = НайденныйСотрудник.Наименование;
        Штрихкод="";
        ЭтаФорма.ТекущийЭлемент = ЭлементыФормы.Штрихкод;
    Иначе
        ЭлементыФормы.НадписьОшибкаСотрудника.Заголовок = ↵
            "Не найден сотрудник с таким штрихкодом! ↵
            Обратитесь к администратору.";
        Штрихкод="";
        ЭтаФорма.ТекущийЭлемент = ЭлементыФормы.Штрихкод;
        возврат;
    КонецЕсли;

    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   ПриходУходСотрудниковСрезПоследних.ФизЛицо,
    |   ПриходУходСотрудниковСрезПоследних.Вход,
    |   ПриходУходСотрудниковСрезПоследних.Выход,
    |   ПриходУходСотрудниковСрезПоследних.Период
    |ИЗ
    |   РегистрСведений.ПриходУходСотрудников. ↵
            СрезПоследних(&ТекДата, ФизЛицо.Ссылка = ↵
            &Найденный) КАК ↵
            ПриходУходСотрудниковСрезПоследних";

    Запрос.УстановитьПараметр("ТекДата", ТекущаяДата());
    Запрос.УстановитьПараметр("Найденный", ↵
```



Давайте за пару дней сэкономим несколько сотен тысяч рублей и создадим свою систему

```
        НайденныйСотрудник.Ссылка);

    Выборка = Запрос.Выполнить().Выбрать();
    Если Выборка.Следующий() Тогда
        //Проверим, последний раз входил или выходил
        Если Выборка.Вход Тогда
            //Значит выходим
            НовСтрока = СписокПрошедших.Добавить();
            НовСтрока.Сотрудник = НайденныйСотрудник;
            НовСтрока.Выход = ТекущаяДата();
            НовЗапись = РегистрыСведений. ↵
                ПриходУходСотрудников. ↵
                СоздатьМенеджерЗаписи();
            НовЗапись.ФизЛицо = НайденныйСотрудник;
            НовЗапись.Выход = Истина;
            НовЗапись.Период = ТекущаяДата();
            НовЗапись.Записать();
            ЭлементыФормы.СписокПрошедших.ТекущаяСтрока = ↵
                НовСтрока;
        Иначе
            //Входим
            НовСтрока = СписокПрошедших.Добавить();
            НовСтрока.Сотрудник = НайденныйСотрудник;
            НовСтрока.Вход = ТекущаяДата();
            НовЗапись = РегистрыСведений. ↵
                ПриходУходСотрудников. ↵
                СоздатьМенеджерЗаписи();
            НовЗапись.ФизЛицо = НайденныйСотрудник;
            НовЗапись.Вход = Истина;
            НовЗапись.Период = ТекущаяДата();
```

```
            НовЗапись.Записать();
            ЭлементыФормы.СписокПрошедших.ТекущаяСтрока = ↵
                НовСтрока;
        Конецесли;
    Иначе
        //Не нашли, значит вход
        НовСтрока = СписокПрошедших.Добавить();
        НовСтрока.Сотрудник = НайденныйСотрудник;
        НовСтрока.Вход = ТекущаяДата();
        НовЗапись = РегистрыСведений. ↵
            ПриходУходСотрудников.СоздатьМенеджерЗаписи();
        НовЗапись.ФизЛицо = НайденныйСотрудник;
        НовЗапись.Вход = Истина;
        НовЗапись.Период = ТекущаяДата();
        НовЗапись.Записать();
        ЭлементыФормы.СписокПрошедших.ТекущаяСтрока = ↵
            НовСтрока;
    Конецесли;
КонецПроцедуры
```

При начале интерактивной работы, при открытии формы сформируем необходимые колонки таблицы значений и активизируем поле ввода штрихкода, чтобы обработка была готова для работы.

```
Процедура ПриОткрытии()
        СписокПрошедших.Колонки.Добавить("Сотрудник");
        СписокПрошедших.Колонки.Добавить("Вход");
        СписокПрошедших.Колонки.Добавить("Выход");
```

Рисунок 1. Рабочий режим обработки

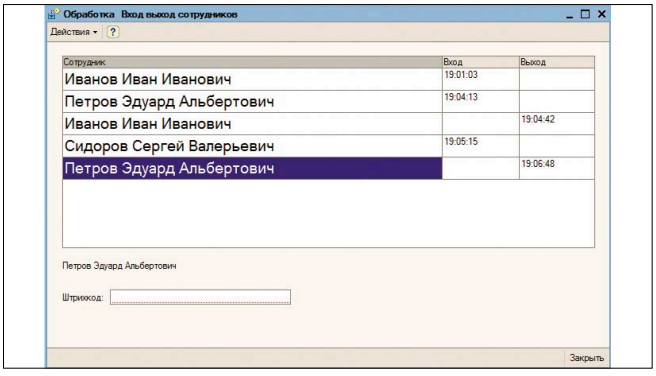
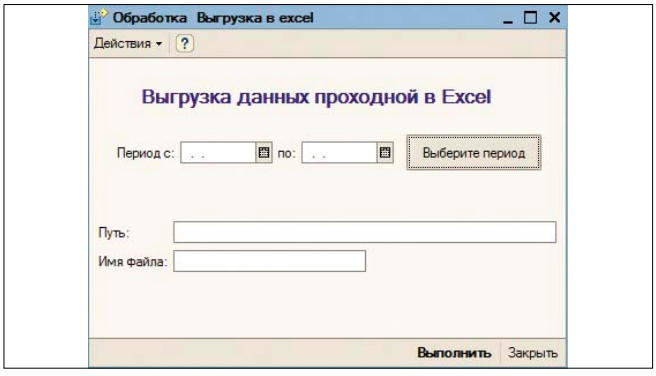


Рисунок 2. Обработка выгрузки сведений в файл Excel



```
        ЭтаФорма.ТекущийЭлемент = ЭлементыФормы.Штрихкод;
КонецПроцедуры
```

Пример обработки – минимальный каркас, функционал которого можно наращивать. Например, можно добавить вывод фотографии сотрудника, его подразделения и должности. Результатом нашей обработки будет являться сформированный набор записей, удобный для дальнейшего анализа (см. рис. 3).

Как можно использовать полученные данные? Зная графики работы, можно отслеживать несанкционированный выход за территорию предприятия, можно подсчитать общее рабочее время, можно выявлять опоздавших сотрудников. Также возможно автоматически формировать табель рабочего времени.

Как анализировать данные? Каждый выбирает свой способ. Идеальный вариант, если кадровый учет ведется в той же программе, в которой мы и запускали электронную проходную. Если же это другая база данных, то придется или подключаться через COM-соединение, или выгружать сведения в файл для дальнейшей загрузки и обработки.

Приведу пример выгрузки данных за определенный период в файл Excel (см. рис. 2).

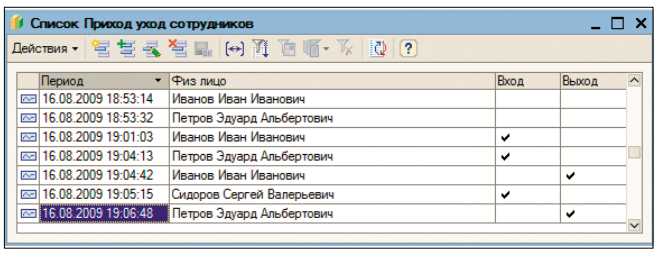
```
Процедура КнопкаВыполнитьНажатие(Кнопка)
    Если (НЕ ЗначениеЗаполнено(НачПериода))
        ИЛИ (НЕ ЗначениеЗаполнено(КонПериода)) Тогда
        Предупреждение("Выберите период!");
        Возврат;
    КонецЕсли;
    //Создание объекта MS Excel
    Excel = Новый COMОбъект("Excel.Application");
    //Создаём новую книгу в Excel
    Excel.WorkBooks.Add();
    //Выводим текст заголовка документа
    Excel.WorkBooks(1).WorkSheets("Лист1").cells(1,1). ↵
        value = "Таб. №";
    Excel.WorkBooks(1).WorkSheets("Лист1").cells(1,2). ↵
        value = "Сотрудник";
    Excel.WorkBooks(1).WorkSheets("Лист1").cells(1,3). ↵
        value = "Дата";
    Excel.WorkBooks(1).WorkSheets("Лист1").cells(1,4). ↵
        value = "Время";
    Excel.WorkBooks(1).WorkSheets("Лист1").cells(1,5). ↵
        value = "Подразделение";
    Excel.WorkBooks(1).WorkSheets("Лист1").cells(1,6). ↵
        value = "Событие";
    //Заполняем со 2-й строки
    Счетчик = 2;
    //Выберем записи о входах-выходах за период
    Запрос = Новый Запрос;
    Запрос.Текст =
    "ВЫБРАТЬ
    |   ПриходУходСотрудников.Период,
    |   ПриходУходСотрудников.ФизЛицо,
    |   ПриходУходСотрудников.Вход,
    |   ПриходУходСотрудников.Выход
    |ИЗ
    |   РегистрСведений.ПриходУходСотрудников КАК ↵
            ПриходУходСотрудников
    |ГДЕ
    |   ПриходУходСотрудников.Период МЕЖДУ &НачПериода ↵
            И &КонПериода";
    Запрос.УстановитьПараметр("НачПериода", ↵
            НачалоДня(НачПериода));
    Запрос.УстановитьПараметр("КонПериода", ↵
            КонецДня(КонПериода));

    Выборка = Запрос.Выполнить().Выбрать();
    Пока Выборка.Следующий() Цикл
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,1).NumberFormat = "@";
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,1).value = ↵
            Строка(Выборка.ФизЛицо.ТабНомер);
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,2).NumberFormat = "@";
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,2).value = Строка(Выборка.ФизЛицо);
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,3).NumberFormat = "@";
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,3).value = ↵
            Строка(Формат(Выборка.Период,"ДЛФ=D"));
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,4).NumberFormat = "@";
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,4).value = ↵
            Строка(Формат(Выборка.Период,"ДЛФ=T"));
        Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
            cells(Счетчик,6).NumberFormat = "@";
        Если Выборка.Вход Тогда
            Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
                cells(Счетчик,6).value = "Вход";
        ИначеЕсли Выборка.Выход Тогда
            Excel.WorkBooks(1).WorkSheets("Лист1"). ↵
                cells(Счетчик,6).value = "Выход";
        КонецЕсли;
        Счетчик = Счетчик + 1;
    КонецЦикла;
    //Записываем файл Excel
    Попытка
        Excel.ActiveWorkBook.SaveAs(СокрЛП(Путь)+ИмяФайла);
    Исключение
        Сообщить("Неудачная попытка сохранения файла");
    КонецПопытки;
    //Закрываем книгу Excel
    Excel.ActiveWorkBook.Close();
КонецПроцедуры

Процедура ВыбПериодНажатие(Элемент)
    НастройкаПериода = Новый НастройкаПериода;
    НастройкаПериода.УстановитьПериод(НачПериода, ↵
        ?(КонПериода='0001-01-01', КонПериода, ↵
        КонецДня(КонПериода)));
    НастройкаПериода.РедактироватьКакИнтервал = Истина;
    НастройкаПериода.РедактироватьКакПериод = Истина;
    НастройкаПериода.ВариантНастройки = ↵
        ВариантНастройкиПериода.Период;
    Если НастройкаПериода.Редактировать() Тогда
        НачПериода = НастройкаПериода.ПолучитьДатуНачала();
        КонПериода = НастройкаПериода. ↵
            ПолучитьДатуОкончания();
    КонецЕсли;
    ИмяФайла = Строка(Формат(НачПериода,"ДЛФ=D"))+".xls";
КонецПроцедуры

Процедура НачПериодаПриИзменении(Элемент)
    ИмяФайла = Строка(Формат(НачПериода,"ДЛФ=D"))+".xls";
КонецПроцедуры
```

Рисунок 3. Набор записей для анализа



Итак, задача по автоматизации учета рабочего времени выполнена, финансовая экономия для организации очевидна, получены структурированные данные о времени прихода и ухода сотрудников. Теперь остается только использовать эти сведения на благо предприятия. EOF

Уязвимость при обработке ICMP-сообщений в Cisco Firewall Services Module

Программа: Cisco Firewall Services Module версии до 3.1(16), 3.2(13) и 4.0(6).
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за ошибки в одном из сетевых процессоров (Network Processor) при обработке определенного ICMP-трафика. Удаленный пользователь может с помощью специально сформированных ICMP-сообщений потребить все доступные потоки и заставить модуль прекратить перенаправление трафика и отключить получение административного трафика.
URL производителя: www.cisco.com.
Решение: Установите последнюю версию 3.1(16), 3.2(13) или 4.0(6) с сайта производителя.

Отказ в обслуживании в CA Host-Based Intrusion Prevention System

Программа: CA Host-Based Intrusion Prevention System 8.1, возможно, другие версии.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за неизвестной ошибки в драйвере kmxlds.sys (версии до 7.3.1.18). Удаленный пользователь может с помощью специально сформированных пакетов вызвать аварийное завершение работы системы.
URL производителя: www.ca.com/us/products/product.aspx?id=5785.
Решение: Установите исправление Cumulative Fix 1 RO10298 с сайта производителя.

Отказ в обслуживании в Squid

Программа: Squid 2.7.STABLE3, возможно, другие версии.
Опасность: Низкая.
Наличие эксплоита: Нет.
Описание: Уязвимость существует из-за ошибки в функции strListGetItem() при проверке внешних заголовков аутентификации. Удаленный пользователь может с помощью специально сформированного заголовка, содержащего запятую, вызвать зацикливание приложения. Для успешной эксплуатации уязвимости требуется, чтобы опция external_acl_type содержала разграничитель, отличный от запятой.
URL производителя: www.squid-cache.org.
Решение: В настоящее время способов устранения уязвимости не существует.

Множественные уязвимости в Adobe ColdFusion и JRun

Программа: Adobe ColdFusion 8.0.1 и более ранние версии; Adobe JRun 4.0.
Опасность: Средняя.
Наличие эксплоита: Нет.
Описание: 1. Уязвимость существует из-за недостаточной обработки входных данных в URL в CFIDE/wizards/common/_logintowizard.cfm, CFIDE/wizards/common/_authenticatewizarduser.cfm и CFIDE/administrator/enter.cfm и в параметре startRow в CFIDE/administrator/logviewer/searchlog.cfm. Удаленный пользователь может с помощью специально сформированного запроса выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.

2. Уязвимость существует из-за недостаточной обработки входных данных в ColdFusion и в консоли управления JRun. Удаленный пользователь может с помощью специально сформированного запроса выполнить произвольный код сценария в браузере жертвы в контексте безопасности уязвимого сайта.

3. Уязвимость существует из-за недостаточной обработки входных данных в параметре logfile в сценарии logging/logviewer.jsp в консоли управления JRun. Удаленный пользователь может с помощью специально сформированного запроса, содержащего символы обхода каталога, просмотреть содержимое произвольных файлов на системе.

4. Уязвимость существует из-за ошибки в ColdFusion, связанной с двойным кодированием нулевого символа. Удаленный пользователь может получить доступ к важным данным на системе.

5. Уязвимость существует из-за ошибки фиксации сессий в ColdFusion. Удаленный пользователь может выполнить некоторые действия в приложении с повышенными привилегиями.
URL производителя: www.adobe.com.
Решение: Установите последнюю версию с сайта производителя.

Отказ в обслуживании в Kaspersky Anti-Virus и Internet Security

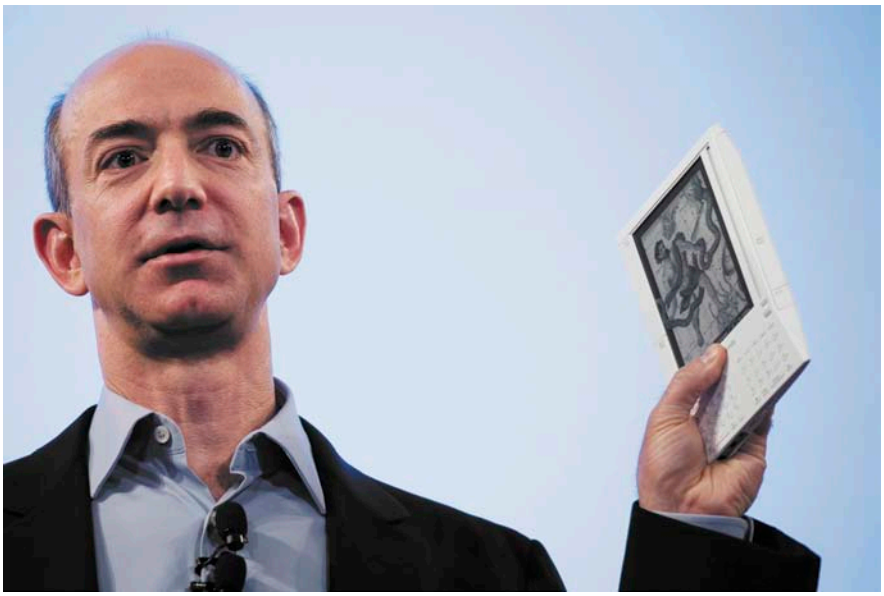
Программа: Kaspersky Anti-Virus 9.0.0.463, возможно, более ранние версии; Kaspersky Internet Security 9.0.0.459, возможно, более ранние версии.
Опасность: Низкая.
Наличие эксплоита: Да.
Описание: Уязвимость существует из-за ошибки при обработке HTTP-запросов к адресам, содержащим множественные точки. Удаленный пользователь может с помощью специально сформированного веб-сайта или почтового HTML-сообщения вызвать зависание процесса avp.exe и отключить сетевые подключения.
URL производителя: www.kaspersky.com.
Решение: Установите исправление с помощью автоматического обновления.

Составил Александр Антипов

Сетевой книгоноша

Джефф Безос — гений интуиции, педант, сорвиголова, профи и лентяй

Владимир Гаков



Основатель, президент, CEO и председатель правления компании Amazon.com в одном лице превратил никому не известный стартап «в папином гараже» в одну из самых парадоксальных и успешных ИТ-компаний! И к сорока годам заработал первый миллиард. Причем на книготорговле!

Сисадмин от сохи

Никаких особых «генов» к занятию высокотехнологичным бизнесом Джефф Безос не имел. Его предки по материнской линии прибыли в Техас вместе с первопоселенцами и на протяжении нескольких поколений содержали там ранчо. Хотя дед занимал в течение некоторого времени пост регионального директора Федеральной комиссии по атомной энергии в Албукерке (штат Нью-Мексико), он, рано выйдя на пенсию, также вернулся «к землице» – на семейное ранчо. На нем же прошло детство и родившегося в 1964 году Джеффри Престона Безоса. Кстати, Безос – фамилия его отчима-кубинца, за которого мать, прожившая с первым мужем чуть больше года, вышла замуж, когда Джеффу было пять лет.

В том же 1969 году семья перебралась в Хьюстон, где отчим получил место инженера в нефтяной компании Exxon. Постоянно помогая дедушке на ранчо, Джефф загорался только тогда, когда представлялась возможность что-то мастерить. По семейной легенде он, еще будучи ползунком, с помощью забытого взрослыми сверла пытался разобрать свою кроватку. А позже соорудил систему электрической сигнализации, чтобы младшие братья не мешали ему заниматься своими экспериментами в детской. И наконец в школьном возрасте полностью оккупировал отцовский гараж, устроив там настоящую лабораторию.

Окончив с отличием среднюю школу, Безос поступил в престижный Принстон. Сначала на физический факультет, а затем перевелся на инженерный – сын фермеров всерьез «запал» на компьютеры. Он дни и ночи просиживал в университетском компьютерном центре. Однокурсники вспоминали, что в Принстоне Безос прославился как изобретатель «пивпонга» (beer pong). Эта игра отличалась от обычного пинг-понга тем, что стол был уставлен кружками, наполненными пивом. Когда мячик попадал в кружку, игрок, на чьей половине это случилось, вынужден был выпивать ее до дна. Так что с «креативностью мышления» у студента Безоса, как видим, все было в порядке.

После Принстона он некоторое время работал на Уолл-стрит сисадмином (хотя тогда их так никто не называл), а затем создавал компьютерную сеть для глобальной телекоммуникационной компании Fitel. В 26 лет Безос занял пост вице-президента трастовой компании Bankers Trust, а затем перешел на пост старшего вице-президента хедж-фонда D.T.Shaw.

На последнем месте работы делавшего стремительную карьеру бизнес-администратора впервые посетила блестящая идея – торговать через Интернет. В те легендарные времена Всемирная паутина только-только начала опутывать земной шар, вырастая на 2000% ежегодно, и Джефф Безос одним из первых прозорливо разглядел ее поистине неограниченные коммерческие возможности. Товаров, ко-

торыми можно было торговать в Сети, он насчитал десятки, но почему-то выбрал самый рискованный и неперспективный – книги (позже расширив ассортимент на CD, DVD и всю сопутствующую продукцию – источники питания и т.п.). И, как мы уже знаем, не прогадал.

Тогда же дело, им затеянное, было во всех отношениях революционным. А руководство его компании, напротив, оказалось консервативным и не проявило никакого интереса к «бредовой идее» молодого сотрудника. Но Безос не унывал, решив реализовать свой проект самостоятельно. В 1994 году он покинул компанию и, погрузив имущество в пикап, вместе с женой отправился в путешествие через всю страну. Из Нью-Йорка – на Западное побережье, в Сиэтл.

Бизнес-план в бардачке

Как гласит корпоративная легенда, по дороге Безос, держа руль одной рукой, второй набрасывал на коленях бизнес-план того, что вскоре обретет название Amazon.com. Заехав по дороге в родной Техас, молодой человек занял у родителей 300 тысяч долларов в качестве стартового капитала. Заметьте, не выпросил, а взял в долг – Америка!

Первым помещением «книжного магазина в режиме онлайн» опять стал старый гараж – в таких же чуть раньше начинали свое дело другие гуру «электронной эры»: Хьюлетт и Паккард и два Стива, Джобс и Возняк. А «дорожная одиссея» Безоса стала тоже своего рода мифом для нового поколения – «поколения ИТ» (или, как его называют в Америке, Dotcom – «.com»).

Поначалу Безос решил назвать свою компанию Cadabra.com. Будучи человеком начитанным, он, конечно, знал, что слово «абракадабра» означает не только «белиберду», «словесную бессмыслицу», но и таинственное заклинание фокусников. Короче, с самого начала затевалось нечто чудесное и магическое! Потом, правда, пришлось отказаться от этого названия – уж очень оно было созвучно «кадавру» (по-французски – труп). Для вновь создаваемой компании хуже не придумаешь… Тогда возникла альтернатива – имя крупнейшей реки в мире. Тот же полноводный поток литературы, не сравнимый с ассортиментом обычного книжного магазина, символизирует и логотип, введенный в 2000 году: стрелка от буквы A (первой в латинском алфавите) к букве Z (последней) в слове Amazon.

Компания Безоса была среди пионеров – вместе с Google, Yahoo и eBay – абсолютно новой сферы бизнеса, обозначаемой теми же заглавными буквами ИТ. Но любопытно, что в отличие от «отцов-основателей» всех трех перечисленных компаний Джефф Безос остался единственным, кто сохранил пост CEO (генерального директора). Иначе говоря, единственным, кто смог справиться с проблемами «переходного периода», когда вчерашние подростки – компьютерные фанаты, которые относились к бизнесу как к новой увлекательной игре, внезапно оказывались руководителями огромных корпораций со штатом в несколько тысяч человек.

Маркетинговая программа новоиспеченной компании была проста и основана на том, что в просторечье называется «сарафанным радио». Безос составил список из трех сотен родственников и друзей, которых попросил заходить на свой сайт и рассказывать о нем всем, кому представится возможность. За первый же месяц, не истратив на рекламу ни цента, Amazon.com начала торговлю с десятками городов и стран. А к Рождеству 1995 года Безос подвел итоги – за первый год существования его компания продала книг на миллион долларов.

Когда скромный интернет-магазин, торговавший книгами, в год своего основания вышел на IPO и первый день продаж акций принес Безосу 500 тысяч долларов, новоиспеченный «полумиллионер» проживал в скромной квартирке в деловом районе Сиэтла и водил старенькую «хонду». А язвительные аналитики с Уолл-стрит пророчили, что предприятие Безоса скоро превратится в Amazon.toast (в смысле «пережаренный хлебец»). Тем более что изначальный бизнес-план не предполагал прибыли в течение первых четырех-пяти лет.

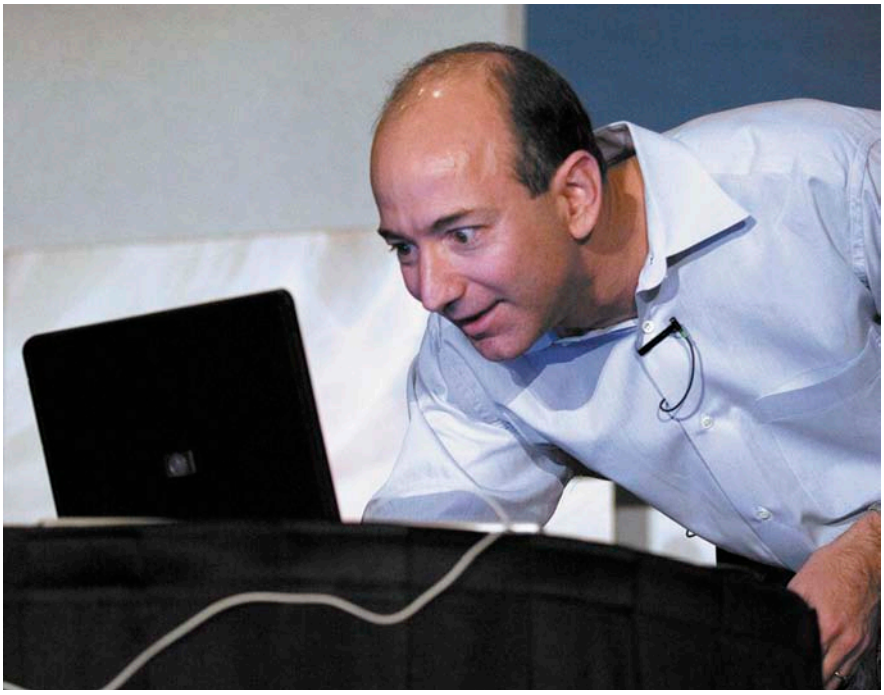
Однако эта стратегия увенчалась успехом. Пока в конце 1990-х другие ИТ-компании стремительно «надувались», компания Безоса росла медленно, но верно. Зато в 2000 году, когда случился всем памятный кризис на этом рынке, «пузыри» столь же стремительно стали лопаться. И многие уже не восстановились – но только не Amazon.com. Компанию Безоса тоже потрепало, но она, несмотря на тяжелые времена, сохранила свои позиции на рынке. В 2001 году начала снова приносить прибыль. Небольшую – всего 5 млн долларов (при обороте, превышающем 1 млрд доларов), но символическую: на тогдашнем ИТ-рынке и это было невероятным достижением. К ноябрю 2005 года Amazon.com снова вошла в список крупнейших (по капитализации) 500 компаний, составляемых рейтинговыи агентством Standard & Poor's.

> Вы завоевываете репутацию, пытаясь делать то, что сложно, и делать это так, как надо… А репутация — это то, что о тебе говорят за глаза

Репутация дороже акций

Безос по сей день проживает в Сиэтле вместе с семьей – женой-писательницей и четырьмя детьми. Их дом расположен на берегу живописного озера Вашингтон по соседству с легендарным «технологическим домом» Билла Гейтса стоимостью в 100 млн долларов.

Огромное здание штаб-квартиры компании со штатом 17 тысяч человек сегодня трудно не заметить – этот «дом на горе», построенный в стиле Art Deco, встречает всякого на въезде в Сиэтл. Что касается присутствия Amazon.com в виртуальном пространстве, то ее логотип ныне знаком миллионам пользователей, ежегодно покупающих в крупнейшем интернет-магазине товары на миллиарды долларов. Не только книги, но и диски, кассеты, программное оборудование, видеоигры, электронную технику – даже мебель, продукты питания и игрушки.



Основатель компании трепетно относится к своему бренду и собственной репутации, ставя между ними знак равенства: «Мне всегда казалось, что бренд компании в большей мере основывается на том, как она работает, а не на том, что она рассказывает о себе… Бренд для компании – это как репутация для личности. Вы завоевываете репутацию, пытаясь делать то, что сложно, и делать это так, как надо... А репутация – это то, что о тебе говорят за глаза».

На репутацию самого Безоса как бизнесмена и топ-менеджера уникального, что называется, от Бога, не повлияли даже трудности, которые испытала его компания во время приснопамятного «лопания пузырей» на ИТ-рынке. А накануне кризиса, в декабре 1999-го, когда Amazon.com уже успела потерять более полумиллиарда, журнал Time тем не менее назвал ее основателя и главу «человеком года». Особо отметив, что эта компания, как никакая другая, способствовала популяризации электронной коммерции.

Руководитель Джефф Безос столь же уникален, как и созданная им компания. Он умудряется сочетать стратегическое мышление «магната-олигарха» с дотошностью менеджера среднего звена. Безос известен своим умением ставить задачи на перспективу и вместе с тем вовлеченностью в мельчайшие детали, знанием каждой шестеренки созданного им гигантского корпоративного механизма. Обычно эти крайности трудно совмещать – на то и существуют обычные менеджеры и те, что с приставкой «топ». А вот Безосу – удается.

Журналисты отмечали склонность главы Amazon.com к рискованным решениям. Правда, некоторые из них на поверку оказываются тщательно просчитанными – или гениально «проинтуиченными». Хотя эти спонтанные решения босса, вроде бы не основанные на серьезном анализе, порой вызывают нервное потрясение у его сотрудников и партнеров. Обычно когда первые результаты подтверждают их худшие опасения! Но проходит время, и оказывается, что принятое решение было как раз самым удачным и своевременным из всех возможных. И вера Джеффа Безоса в успех его очередного начинания стоит тех бумажных кип с таблицами и графиками, в которые предпочитают слепо верить другие топ-менеджеры.

Для Безоса инновации – это не просто один из механизмов его бизнеса, а механизм главный, движущий, единственный. И не работающий в отсутствие риска. Для того чтобы поддерживать в компании дух инноваций, глава Amazon.com постоянно осуществляет, по его собственным словам, «разумную селекцию персонала»: «Есть люди, которые любят, когда все и повсюду меняется с огромной скоростью. Такие сотрудники обожают изобретать, что неизбежно связано с блужданием по неисхоженным тропкам, многие из которых заканчиваются тупиком. А другим по душе более стабильное окружение, которое обеспечивает уверенность в завтрашнем дне. Такие у нас, в Amazon.com, обычно не задерживаются».

Те же, кто задержался, повторяют, как священную мантру, слова своего босса: «Успех в краткосрочной перспективе – вовсе не индикатор того, что то же самое произойдет в перспективе долгосрочной. Не все можно просчитать – и тогда на помощь приходит вера».

Начальничек – обхохочешься!

Безос свято верит, что Amazon.com – самая «клиентоориентированная» компания в мире, и свою бизнес-стратегию строит на основополагающем принципе: «что хорошо для покупателя, хорошо для продавца». Пусть и не в ближней перспективе, но удовлетворение клиента когда-нибудь принесет удовлетворение (материальное) поставщику товаров и услуг. Для достижения этой цели Безос готов на все – принимать непопулярные (не обещающие быстрого «навара») решения, заимствовать удачные находки у конкурентов, прислушиваться к мнению критиков. И даже размещать их критические высказывания на сайте Amazon.com!

При всем при том внешне Безос производит впечатление человека бесхитростного, открытого и даже несколько безалаберного. И беспредельно, агрессивно оптимистичного – даже по американским меркам. Его жизненное кредо: «Work hard, have fun, make history» («Работай, получай удовольствие, твори историю») – не больше и не меньше. Те, кому доводилось с ним беседовать, особенно отмечают его неподражаемый смех по случаю и без, получивший название «гусиного кряканья». Это даже не смех, а оглушительный хохот, который словно призывает собеседника «быть проще», но на самом деле служит для Безоса его собственным «оружием массового поражения».

Могут сбить с толку и вопросы, которые задает глава Amazon.com претендентам на занятие руководящих постов в компании. Безос ни к селу ни к городу спрашивает, сколько всего окон в Сан-Франциско, и внимательно

наблюдает, как испытуемый будет выкручиваться. А затем разрядит тягучее молчание очередным взрывом хохота – мол, не парься, я пошутил!

Не только внешнее поведение Джеффа Безоса, но и его представления о том, как управлять большой организацией, весьма далеки от сложившихся корпоративных стандартов.

Всем известно, что чем больше компания, тем больше в ней должны быть развиты внутренние коммуникации. А Безос упрямо утверждает: «Нет, коммуникации – это ужас!» И настаивает на том, что современная ИТ-корпорация должна представлять собой предельно децентрализованную, атомарную структуру, в рамках которой отдельные малые группы сотрудников могли бы внедрять инновации и тестировать свои «визионерские» проекты независимо от влияния других групп и мнения начальства.

Эта философия нашла свое выражение в выдвинутом Безосом принципе «двух пицц». Христос, согласно библейской легенде, накормил целый народ семью хлебами. А новый гуру «децентрализованного менеджмента» считает, что если команду, «заточенную» на решение конкретной инновационной задачи, невозможно накормить всего двумя пиццами, значит, штаты непомерно раздуты, и их нужно сокращать.

Безбашенный Безос

В марте 2003 года Amazon.com вполне могла остаться без своего основателя и незаменимого лидера-харизматика. Отправившись в прогулку на вертолете по родному Техасу с целью присмотреть для себя ранчо, Джефф Безос попал в авиакатастрофу. К счастью, все обошлось – потерявший управление вертолет совершил вынужденную посадку на прибрежное мелководье, однако пилот и пассажир не пострадали. Атаковавшим его журналистам Безос ответил, что всегда верил в свою необычайную удачу, не подвела она его и на сей раз.

Полагаться на везение главе Amazon.com приходилось не раз – так было в начале карьеры, когда его детищу все единогласно прочили судьбу поджаренного тоста, и в черный для ИТ-рынка год наступления Миллениума. Однако несокрушимый оптимист Безос всегда воспринимал выпавшие на его долю испытания как вызовы судьбы как вновь открывшиеся удачные возможности. Он настолько неотделим от своей компании, что трудно представить, чем займется Безос, если ее вдруг не станет. Хотя о кое-каких ближайших альтернативах глава Amazon.com уже «проговорился».

В 2004 году СМИ облетела сенсационная новость – Безос основал компанию Blue Origin, целью которой станет организация коммерческого космического туризма. Для этого предполагается создание нового поколения космических челноков, которые будут доставлять состоятельных туристов на низкую орбиту и возвращать их на Землю. Вопреки сложившемуся имиджу человека открытого и словоохотливого, Безос на сей раз в общении с прессой был нем как рыба. Свой «заговор молчания» он прервал лишь в начале 2007 года, продемонстрировав журналистам видеопленку, на которой были запечатлены тестовые испытания прототипа космического челнока.

Взлет и посадка прототипа нового «шаттла» в автоматическом режиме осуществлялись на том самом ранчо, которое Безос присмотрел во время своего неудачного полета на вертолете. Сегодня мало кто сомневается, что одним из первых в орбитальный полет, запланированный на 2010 год, отправится сам руководитель и главный спонсор проекта – «рисковый» Джефф Безос. EOF

Законы амазонских джунглей (по Безосу)

1. Нанимай персонал крайне осторожно – своим выбором ты даешь эволюционный толчок «долгоиграющей» корпоративной культуре. (Безос предпочитает провести собеседование с полсотней претендентов и никого не выбрать, нежели взять на работу никчемного сотрудника.)
2. Будь упрям и одновременно пластичен. (Слишком большое упрямство со временем поставит крест на любых экспериментах. А отсутствие пластичности заставляет биться головой о стену, не замечая открытой рядом потайной дверцы.)
3. Будь одержим проблемами клиентов, а не коллег или конкурентов.
4. Знай, в какой момент следует забыть о корпоративных правилах. (Решения, основанные на фактах, не знают иерархии: младший клерк может оказаться «более правым», чем его босс. С другой стороны, интуитивное решение лучше получается у опытных руководителей, успевших основательно «заточить» свои инстинкты.)
5. Выслушивай добрые советы – и игнорируй их. («Нам никто не советовал начинать то, что мы затеяли. Мы слушали советы доброхотов, игнорировали их и на первых порах понимали, что поступали неправильно. Но с течением времени мы убеждались, что именно отказ от тех советов было лучшим, что мы могли сделать».)
6. Не гонись за «быстрым долларом». (Спад продаж на раннем этапе – вовсе не показатель того, что это будет продолжаться и впредь.)
7. Коммуникация – это ужас.
8. Не бойся колебаний.
9. Будь проще. (Когда невозможно принять решение, основанное на объективных данных, выручит обыкновенный здравый смысл, которым обычно руководствуются и клиенты.)
10. Собирай вместе малые достижения. (Что и демонстрирует успех Amazon.com – компания, успех которой состоялся не вдруг, а складывался по кирпичикам.)

Визитка

АНДРЕЙ УВАРОВ, инженер-разработчик, интересы: веб-разработка на Java, паттерны проектирования, функциональное программирование, Lisp-подобные языки

Основы Spring

Фреймворк Spring заслуженно называют архитектурным клеем, так как на его основе строятся очень сложные, но тем не менее изящные приложения

Spring – это фреймворк, который позволяет создавать модульные масштабируемые системы. Любой разработчик, непосредственно вовлечённый в процесс создания Enterprise-приложений на Java, просто обязан хорошо разбираться в технологии, ставшей стандартом де-факто в этой отрасли разработки ПО. Последняя стабильная версия имеет номер 2.5.6 и доступна на официальном сайте – http://www.springsource.org. Там же доступна очень подробная документация.

Структура

За время своего существования Spring оброс большим количеством функционала, выделяемого в отдельные модули. Давайте рассмотрим архитектуру этого фреймворка. На рис. 1 изображены модули, которые в совокупности и называются Spring Framework.

Core – ядро системы, реализованное в виде IoC-контейнера;

ORM (Object-Relationship mapping) – модуль, предоставляющий средства для взаимодействия с наиболее популярными ORM-фреймворками[1], как JPA, JDO, Hibernate и другими;

DAO (Data Access Objects) – данный модуль предоставляет возможности управления транзакциями (включая декларативное управление). Также включает в себя набор шаблонов для работы с JDBC;

JEE (Java Enterprsie Edition) – часть системы, которая реализует взаимодействие с технологиями, обобщёнными названием JEE. Вот только некоторые из них: JMS, JMX, JCA, EJB;

AOP (Aspec-Oriented Programming) – реализация Spring AOP и поддержка @AspectJ. AOP даёт нам дополнительные возможности физического разделения кода с целью сделать код более структурированным[2];

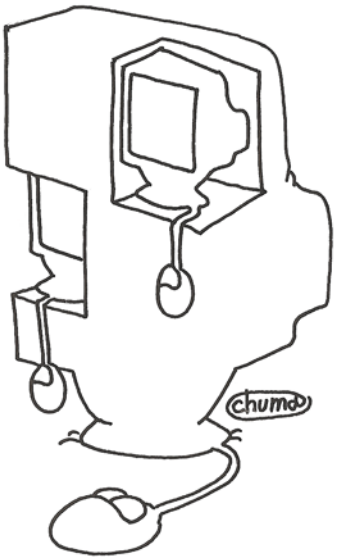
WEB – модуль, содержащий средства для взаимодействия с популярными веб-фреймворками (Struts, Tapestry), шаблонизаторами (Velocity, Freemarker) и многими другими полезными в веб-разработке вещами. Также включает в состав Spring MVC – собственную реализацию паттерна Model-View-Controller.

Core – теория

Все составные части прочно держатся на одной основе – core (ядре), которое представляет собой так называемый IoC-контейнер. IoC расшифровывается как Inversion of Control и является паттерном проектирования. Суть IoC заключается в уменьшении связности кода. Примером является паттерн Dependency Injection, следуя которому зависимости в нашем коде строятся на интерфейсах и абстрактных классах. И уже в ходе выполнения кода фреймворк берёт всю работу по инстанцированию на себя, следуя заранее определённым нами правилам.

В качестве примера можно привести следующую ситуацию: мы программируем действия вывода на печать, оперируя неким интерфейсом PrintService с единственным методом void print(String message), не задумываясь о том, какая реализация будет использована в ходе выполнения. А затем при помощи Spring мы конфигурируем свой код таким образом, что в некоторых случаях будет использована реализация, выводящая сообщения на экран, в других – на принтер и т.д. Так, если ещё нет никаких реализаций PrintService, то на время написания и тестирования своего кода мы мо-

1. ORM-технология, позволяющая связывать реляционное представление данных с проектируемой объектной моделью. То есть имея бизнес-объекты, мы не заботимся о том, как они будут сохраняться и извлекаться из базы данных. Связь бизнес-объектов с их представлением осуществляется декларативно, то есть мы определяем для каждого класса таблицу, в которой его экземпляры будут храниться.
2. При помощи AOP реализуется функциональность, которую сложно выделить в отдельные сущности. В качестве примера можно привести использование транзакций – при помощи AOP мы можем определить, что все вызовы методов, которые соответствуют маске «public com.example.dao.Hibernate*Dao.update*(..)», должны быть выполнены в транзакционном контексте.



Spring позволяет выйти на новую ступень разработки более гибкого кода

жем создать реализацию MockPrintService, которая будет выводить сообщения в отладочный лог-файл. А в момент, когда другой разработчик создаст нужную реализацию PrintService, мы без каких-либо изменений в java-коде легко сможем переключиться на неё.

Dependency Injection позволяет нам упростить код. Упрощение кода даёт выигрыш в скорости его разработки и простоте поддержки. Чем проще код, тем легче его тестировать, что также является плюсом. Положительным моментом является и то, что код не будет зависеть от фреймворка, что увеличивает его переносимость.

Это и есть теоретический минимум, необходимый для дальнейшего постижения Spring.

Core – практика

А теперь разберём очень простой и очень традиционный пример «Hello World».

Начнём с построения интерфейса:

```
package com.andrewdashin.examples.spring.beans;

public interface HelloBean {
    public void helloWorld();
}
```

Создадим реализацию нашего интерфейса:

```
package com.andrewdashin.examples.spring.beans;

public class HelloBeanImpl implements HelloBean {

    String hello;

    public void helloWorld() {
        System.out.println(hello);
    }

    public void setHello(String hello) {
        this.hello = hello;
    }
}
```

Следующим шагом определим правила, по которым будут создаваться объекты в нашем приложении, файл context.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"
    "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>

    <bean id="helloBean" class="com.andrewdashin. ↵
        examples.spring.beans.HelloBeanImpl">
        <property name="hello">
            <value>Hello World!</value>
        </property>
    </bean>

</beans>
```

Теперь непосредственно код приложения:

```
package com.andrewdashin.examples.spring;

import com.andrewdashin.examples.spring.beans.*;
import org.springframework.context.support. ↵
       ClassPathXmlApplicationContext;


public class Main {
    public static void main(String[] args) {
        ClassPathXmlApplicationContext context = ↵
                new ClassPathXmlApplicationContext( ↵
                "resources/beans.xml");
        HelloBean bean = (HelloBean) ↵
                context.getBean("helloBean");
        bean.helloWorld();
    }
}
```

Добавим файл manifest:

```
Manifest-Version: 1.0
Created-By: 1.5.0_13 (Apple Inc.)
Main-Class: com.andrewdashin.examples.spring.Main
Class-Path: lib/spring.jar lib/commons-logging-1.1.jar
```

Как вы, наверное, уже заметили, необходимо создать в корневом каталоге проекта папку lib и поместить туда spring.jar и commons-logging-1.1.jar. Дело в том, что Spring использует большое количество сторонних библиотек, и в частности – Apache Commons Logging. В связи с этим Spring Framework доступен в виде двух дистрибутивов: только фреймворк и фреймворк вместе со всеми зависи-

мостями. Если при скачивании был выбран второй вариант, то необходимый jar-архив можно найти в дистрибутиве, иначе – на официальном сайте http://commons.apache.org/logging.

Компилируется наш пример следующим образом:

```
javac -cp lib/spring.jar -d ./ ↵
  src/com/andrewdashin/examples/spring/Main.java ↵
  src/com/andrewdashin/examples/spring/beans/HelloBean.java ↵
  src/com/andrewdashin/examples/spring/beans/ ↵
  HelloBeanImpl.java
```

Запакуем в jar:

```
jar cvfm main.jar MANIFEST.MF ↵
  com/andrewdashin/examples/spring/Main.class ↵
  com/andrewdashin/examples/spring/beans/HelloBean.class ↵
  com/andrewdashin/examples/spring/beans/ ↵
  HelloBeanImpl.class resources/context.xml
```

И финальная стадия – практически проверим работу приложения (см. рис. 2).

Кроме ожидаемого результата, на экране появилось много дополнительной информации. Эта информация есть не что иное, как лог, ведущийся фреймворком и помогающий понять, что в данный момент происходит. Возможности логирования очень легко настраиваются под любые ситуации.

Конечно, для создания HelloWorld-приложений Spring не лучший выбор, т.к. является в данном случае весьма избыточным. Но чем больше кода в системе, тем более полезен данный фреймворк.

Вернёмся к нашему примеру. Итак, интерфейс и его реализация чрезвычайно примитивны, в связи с чем удостоим нашего внимания файл context.xml. Данный файл является описанием контекста приложения. В секции beans есть единственное определение bean с некоторыми свойствами. Bean – есть сущность, которая определяет правила создания экземпляров объектов. Параметр id определяет уникальный идентификатор, по которому будут создаваться экземпляры объектов, класс которых определён параметром class. Определяя свойство hello, мы заставляем Spring при каждом создании бина вызывать метод setHello(), передавая в качестве аргумента указанное нами строковое значение «Hello World!». Аргументами могут быть как значения, так и ссылки на другие бины.

Входной точкой нашего приложения является класс Main. Spring-контекст создаётся следующим образом:

```
new ClassPathXmlApplicationContext("resources/context.xml")
```

Контекстом является класс-фабрика, в данном случае ClassPathXmlApplicationContext, которая, руководствуясь заданной конфигурацией, создаёт необходимые нам объекты.

В приведённом примере экземпляр helloBean мы получали с уже автоматически заполненным свойством hello. В общем случае свойство hello будет являться классом, который в свою очередь также нужно будет инстанцировать.

Обычно в приложениях множество объектов зависят друг от друга. В качестве примера рассмотрим фрагмент конфигурации контекста:

```
<bean id="abstractController" abstract="true">
    <property name="commandClass" ↵
        value="org.example.Command"/>
</bean>

<bean id="requestDao"
      class="org.example.dao.RequestDaoImpl">
      <constructor-arg index="1" ↵
        value="org.example.domain.RequestEntity"/>
</bean>

<bean id="requestService"
      class="org.example.service.RequestServiceImpl">
    <property name="requestDao" ref="requestDao"/>
</bean>

<bean id="someRequestController"
      class="org.example.mvc.Controller" ↵
        parent="abstractController">
    <property name="requestService" ref="requestService"/>
</bean>

<bean id="otherRequestController"
      class="org.example.mvc.OtherController" ↵
        parent="abstractController">
    <property name="requestService" ref="requestService"/>
</bean>
```

Чтобы создать экземпляр someRequestController, необходимо прежде создать requestService, который в свою

Рисунок 1. Структура

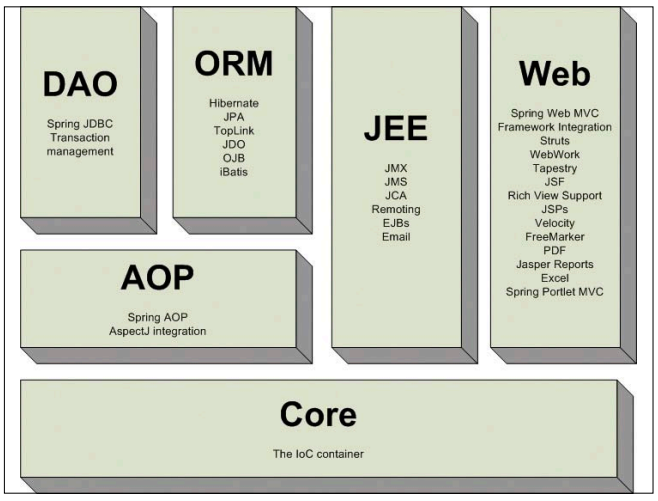


Рисунок 2. Результат

очередь зависит от requestDao. Порой графы таких зависимостей могут достигать достаточно больших размеров. И сложно представить, как можно было бы справиться вручную с созданием одного простого класса, если бы он зависел от множества других.

В примере использованы два типа удовлетворения зависимостей (Dependency Injection): посредством set-методов и при помощи конструкторов. Так, если мы пишем:

```
<property name="requestDao" ref="requestDao"/>
```

то Spring ищет метод с именем setRequestDao и пробует его выполнить, передавая в качестве аргумента указанное значение.

В случае с конструктором – вызывается конструктор класса, передавая аргументы, в указанном параметром index порядке. Если есть необходимость, то можно использовать сразу оба метода для конфигурирования одного объекта. В примере мы определили бин abstractController как абстрактный, а также определили для него свойство. Абстрактные бины в Spring выполняют функцию обобщения. Например, если у вас есть несколько бинов, которые имеют одинаковые зависимости (и могут быть абсолютно разными классами, как в случае с someRequestController и otherRequestController), то, вынося общие свойства в абстрактный бин, мы можем уменьшить повторение кода в конфигурации контекста.

Если скрупулёзно изучить код во время выполнения, то может обнаружиться, что someRequestController и otherRequestController используют один и тот же экземпляр класса requestService. То есть фреймворк один раз создал requestService, а когда появилась необходимость в повторном создании, он просто вернул уже существующий экземпляр.

Дело в том, что Spring реализует несколько способов создания бинов:

Синглтон – способ по умолчанию. Создаётся и используется в дальнейшем только один объект. Эта возможность является весьма полезной, так как помогает избежать реализации шаблона проектирования singleton в Java-коде (что упрощает написание тестов).

Прототип – каждый раз создаётся новый объект.

Запрос, сессия и глобальная сессия – эти способы используются в веб-приложениях. Будет создаваться лишь один экземпляр на каждый http-запрос или сессию.

Экземпляры всех бинов, объявленных как синглтон (явным или неявным образом), создаются на этапе инициализации контекста, что иногда может привести к существенному замедлению запуска приложения. Для подобных случаев предусмотрена возможность «ленивой» инициализации, т.е. бины будут инициализироваться лишь тогда, когда появится необходимость в их использовании.

***

В данной статье вы познакомились лишь с малой частью Spring, с его основами. Но даже используя только их, можно выйти на новую ступень разработки кода, более гибкого, переносимого, легко тестируемого (и как следствие, имеющего большее покрытие тестами) и без дополнительных затрат стратегически важного ресурса – времени. EOF

Визитка

ДМИТРИЙ ВАСИЛЬЕВ, больше 10 лет профессионально занимается разработкой ПО, принимает активное участие в различных проектах с открытым исходным кодом

Пишем первые модули на Erlang *

Продолжаем изучать Erlang — начнем писать программы, которые выполняются последовательно

Модули и функции

В Erlang программы состоят из функций, которые вызывают друг друга. Функции в свою очередь группируются и определяются внутри модулей. Модули в Erlang хранятся в файлах с расширением .erl, при этом имя модуля должно быть таким же, как и имя файла. Перед тем как запустить модуль, его нужно скомпилировать. Компилированные модули содержатся в файлах с расширением .beam.

Определение функции состоит из заголовка и тела функции. Заголовок функции состоит из имени функции, которое является атомом, за которым в скобках следуют формальные параметры функции. Количество параметров функции называется арностью (arity). Функции в Erlang уникально определяются именем модуля, именем функции и арностью, то есть две функции, находящиеся в одном модуле с одинаковыми именами, но с разной арностью, являются разными функциями. Стрелка (->) отделяет заголовок функции от ее тела.

Как уже было сказано, мы не можем определять функции в интерактивной сессии оболочки Erlang.

Давайте напишем наш первый модуль и рассмотрим его подробнее. Создадим файл с именем geometry.erl:

```
-module(geometry).
-export([area/1]).


% Функция для вычисления площади
area({square, Side}) ->
    Side * Side;
area({rectangle, Width, Height}) ->
    Width * Height;
area({circle, Radius}) ->
    3.1415926 * Radius * Radius.
```

В начале модуля находятся директивы модуля в следующем формате: -директива(значение). Директива module описывает имя модуля, которое должно совпадать с именем файла. Директива export описывает экспортируемые функции (которые будут доступны снаружи модуля) в виде списка в формате имя/арность. В данном случае наш модуль называется geometry и экспортирует одну функцию area с одним аргументом. Заметьте, что каждая директива заканчивается точкой.

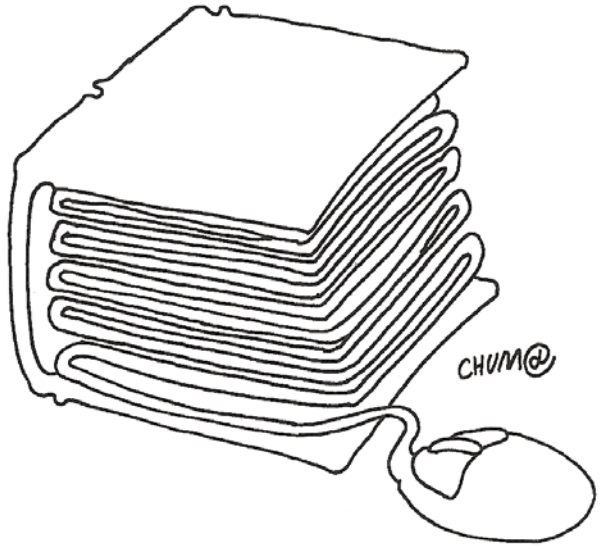
Строки, начинающиеся со знака %, являются комментариями, как мы уже рассматривали выше.

После комментария идет определение функции. В данном случае определение функции состоит из трех предложений, разделенных знаком «;» и последнее выражение функции заканчивается точкой. При вызове функции переданные аргументы последовательно сравниваются с шаблонами формальных параметров, пока не будет найдено нужное предложение. После того как найдено нужное предложение, выполняется выражение, находящееся в теле этого предложения, и возвращается результат этого выражения. В данном случае шаблоны параметров для предложений взаимоисключающие и порядок предложений не имеет значения, но в других случаях порядок предложений может быть важен.

Попробуем выполнить функцию из нашего модуля:

```
1> c(geometry).
{ok,geometry}
2> geometry:area({circle, 20}).
1256.63704
3> geometry:area({square, 20}).
400
4> geometry:area({rectangle, 10, 20}).
200
5> geometry:area({triangle, 10, 20, 30}).
** exception error: no function clause matching
geometry:area({triangle,10,20,30})
```

* В первой части статьи «Знакомьтесь, Erlang» (№8, 2009 г.) были рассмотрены особенности языка, основные типы данных, переменные и сравнение с шаблонами.



Подробную информацию можно найти в справочном руководстве по Erlang

В первой строке мы использовали функцию c(), определенную в оболочке для компиляции нашего модуля. Эта функция возвращает {ok, geometry}, что говорит об успешной компиляции модуля. Вне оболочки модуль может быть скомпилирован с помощью утилиты erlc.

После этого мы делаем несколько вызовов нашей функции, используя нотацию модуль:функция. Мы передаем различные кортежи в качестве аргументов, и выполняется тело того предложения функции, с шаблоном которого совпадает переданный аргумент. В пятой строке мы передали кортеж, который не совпадает ни с одним из шаблонов в определении функции, и получили ошибку.

Более сложный пример

Теперь рассмотрим более сложный пример с использованием ввода/вывода и рекурсии:

```
-module(persons).
-export([person_list/1]).


person_list(Persons) ->
    person_list(Persons, 0).

person_list([{person, FirstName, LastName} | Persons], N) ->
    io:format("~s ~s~n", [FirstName, LastName]),
    person_list(Persons, N + 1);
person_list([], N) ->
    io:format("Total: ~p~n", [N]).
```

Новый модуль называется persons и экспортирует функцию person_list/1 (с одним аргументом). Заметьте, что в модуле также есть функция person_list/2 (с двумя аргументами), но в данном случае она будет локальной для модуля. Функция person_list/1 вызывает вспомогательную функцию person_list/2.

Функции person_list/2 необходимо передать два аргумента: список пользователей и начальное значение для аргумента-счетчика. Функция person_list/2 состоит из двух предложений. В первом предложении функции мы отделяем первый элемент списка пользователей (заметьте, что мы отделяем имя и фамилию прямо в шаблоне аргумента). Затем используется функция format из библиотечного модуля io, чтобы вывести имя и фамилию пользователя, и после этого мы вызываем person_list/2 с оставшимися пользователями и увеличенным счетчиком пользователей.

Библиотечной функции io:format/2 нужно передать два аргумента – формат вывода и список аргументов. В данном случае формат вывода состоит из двух шаблонов для вывода строк ~s и перевода строки ~n. Модуль io содержит большое количество функций для работы со стандартным вводом/выводом.

Второе (и последнее) предложение функции print_list/2 вызывается, когда список пользователей оказывается пустым (обычно это происходит при окончании вывода пользователей), и выводит общее количество выведенных имен пользователей с помощью аргумента-счетчика. Во втором предложении мы также используем новый шаблон для io:format/2 – ~p, выводящий аргумент в формате, в котором это делает оболочка.

Давайте попробуем использовать наш модуль в интерактивной сессии:

```
1> c(persons).

{ok,persons}

2> persons:person_list([]).

Total: 0
ok

3> persons:person_list([{person, "Joe", "Armstrong"}]).

Joe Armstrong
Total: 1
ok

4> persons:person_list([{person, "Joe", "Armstrong"},
4> {person, "Mike", "Williams"},
4> {person, "Robert", "Virding"}]).

Joe Armstrong
Mike Williams
Robert Virding
Total: 3
ok
```

Мы видим, что функция работает, как мы и ожидали.

Ограничители

Часто сравнения с шаблоном для функций бывает недостаточно, и здесь на помощь приходят ограничители (guards), которые позволяют использовать простые тесты и сравнения переменных в шаблонах. Кроме функций, ограничители можно использовать в некоторых других конструкциях, которые мы рассмотрим далее, например, в конструкции case. Для функций ограничители должны быть расположены перед символами ->, разделяющими заголовок и тело функции. Например, можно написать функцию для нахождения максимального значения следующим образом:

```
max(X, Y) when X > Y ->
    X;
max(_X, Y) ->
    Y.
```

В первом предложении функции используются ограничители, начиная со слова when. Первое предложение выполняется только в случае, если X > Y, иначе выполняется второе предложение. Во втором предложении первая переменная называется _X – использование подчеркивания в начале имени переменной позволяет избежать предупреждения о неиспользуемой переменной, хотя этим нужно пользоваться с осторожностью, чтобы не пропустить ошибочные ситуации.

Ограничители представляют собой либо одно условное выражение, которое возвращает true/false, либо могут быть записаны как составное выражение следующим образом:

- Последовательность ограничителей, разделенных точкой с запятой (;), истинна, если хотя бы один из ограничителей в последовательности возвращает true;
- Последовательность ограничителей, разделенных запятой (,), истинна, только если все ограничители в последовательности возвращают true.

Не все выражения доступны для использования в качестве ограничителей, чтобы избежать возможных побочных эффектов. Вот список доступных выражений:

- Атом true (истина).
- Различные константы и переменные. В ограничителях все они представляют собой ложные значения.
- Функции для тестирования типов данных и некоторые встроенные функции, например: is_atom, is_boolean, is_tuple, size и др.
- Сравнение терминов, например =:=, =/=, <, > и т.п.
- Арифметические операции.
- Булевские операции.
- Булевские операции с короткой схемой вычисления (short-circuit).

Условное выполнение

В Erlang есть три формы условного выполнения, которые в большинстве случаев могут быть взаимозаменяемы. С первой формой мы уже познакомились при изучении функций – это использование сравнения с шаблонами в определении функций. Ниже мы рассмотрим еще две формы условного выполнения – конструкции case и if.

В конструкции case сначала выполняется выражение, и затем результат последовательно сравнивается с шаблонами. С шаблонами также можно использовать и ограничители. Рассмотрим пример:

```
case is_boolean(Variable) of
    true ->
        1;
    false ->
        0
end
```

В этом (достаточно надуманном) примере в качестве выражения case ... of выполняется функция is_boolean и шаблонами служат true и false. Два предложения разделены точкой с запятой, и конструкция заканчивается ключевым словом end. В случае если подходящий шаблон не будет найден, то будет выкинуто исключение.

Конструкция if использует только ограничители, которые последовательно выполняются, пока не будет получено значение «истина»:

```
if
    X > Y ->
        true;
    true ->
        false
end
```

В данном случае ограничитель true действует как конструкция «иначе» в других языках, то есть значением if будет false, если «X =< Y». В случае если ни один из ограничителей не даст значение «истина», будет выкинуто исключение.

Анонимные функции

Определяются с ключевым словом fun и похожи на определение обычных функций за исключением отсутствия имени. Рассмотрим пример:

```
-module(times).
-export([times/1]).


times(N) ->
    fun
        (X) when is_number(X) ->
            X * N;
        (_) ->
            erlang:error(number_expected)
    end.
```

Здесь функция times является функцией высшего порядка, так как возвращает другую функцию. Определение анонимной функции между ключевыми словами fun ... end состоит из двух предложений. В первом предложении с помощью ограничителя is_number мы определяем, что передано число (число может быть целым или вещественным), и умножаем его на аргумент, переданный в основную функцию. Во втором предложении мы немного забегаем вперед и используем генерацию исключений, которая будет рассмотрена в следующем разделе.

Попробуем выполнить функцию:

```
1> c(times).
```

```
{ok,times}
```

```
2> N2 = times:times(2).
```

```
#Fun<times.0.120017377>
```

```
3> N2(4).
```

```
8
```

```
4> N10 = times:times(10).

#Fun<times.0.120017377>

5> N10(4).

40
```

В строке 2 мы используем функцию times:times для получения функции, умножающей значение на 2, и в строке 4 создается функция, умножающая значение на 10.

Стандартный модуль lists экспортирует некоторое количество функций, которые принимают функции в качестве аргументов, например, функция lists:map вызывает функцию, используя каждый элемент списка по очереди:

```
6> Double = times:times(2).

#Fun<times.0.120017377>

7> lists:map(Double, [1, 2, 3, 4]).

[2,4,6,8]
```

Обработка исключений

Обычно исключения генерируются в случае обнаружения ошибки. Наиболее часто встречающиеся типы исключений – это исключения, связанные со сравнением шаблонов (мы уже встречались с ними раньше), и исключения, связанные с неверными аргументами функций. Теперь давайте рассмотрим, как можно перехватить и обработать различные типы исключений и как генерировать исключения в своем коде.

Исключения в своем коде можно создать, используя одну из встроенных функций:

exit(Why) – используется, когда нужно действительно прервать выполнение текущего процесса. Если это исключение не перехватывается, то всем процессам, присоединенным к данному, посылается сообщение {'EXIT', Pid, Why}.

throw(Why) – используется для генерации исключения, которое вызывающая сторона, возможно, захочет перехватить. Таким образом, мы документируем, что наша функция может генерировать данное исключение;

erlang:error(Why) – используется для аварийных ситуаций, которые не ожидает вызывающая сторона.

Теперь разберемся, как эти исключения обрабатывать. В Erlang существует два способа обработки исключений – выражение catch и конструкция try ... catch.

Выражение catch возвращает либо значение подвыражения, либо информацию об ошибке в зависимости от типа ошибки. Рассмотрим на примере:

```
1> catch 2 + 2.

4

2> catch 2 + a.

{'EXIT',{badarith,[{erlang,'+',[2,a]},
                   {erl_eval,do_apply,5},
                   {erl_eval,expr,5},
                   {shell,exprs,6},
                   {shell,eval_exprs,6},
                   {shell,eval_loop,3}]}}

3> catch exit("Exit").

{'EXIT',"Exit"}

4> catch throw("Throw").

"Throw"

5> catch erlang:error("Error").

{'EXIT',{"Error",
         [{erl_eval,do_apply,5},
          {erl_eval,expr,5},
          {shell,exprs,6},
          {shell,eval_exprs,6},
          {shell,eval_loop,3}]}}
```

В первой строке мы пробуем catch с выражением «2 + 2», которое успешно выполняется, возвращая 4. Во второй строке делается попытка сложить целое и атом, и catch возвращает описание ошибки в виде {'EXIT', {ошибка, стек вызовов}}. Следующие три строки показывают возвращаемые значения в зависимости от способа генерации исключений. Часто catch используют совместно с конструкцией case для обработки ошибок в выражениях.

Конструкция try ... catch позволяет обрабатывать только необходимые для обработки типы ошибок и даже может быть совмещена с конструкцией, похожей на case. Пример:

```
try 2 + a of
    Value ->
        ok
catch
    error:_ ->
        error
end.
```

Мы пытаемся выполнить выражение «2 + a», и шаблоны между of ... catch соответствуют шаблонам в выражении case. Шаблоны между catch ... end (в которых также можно использовать ограничители) используются для сопоставления с ошибками, где ошибка описывается как тип:значение.

Библиотечные модули

В состав Erlang включено большое количество стандартных библиотечных модулей. Их подробное описание можно найти по ссылке: http://erlang.org/doc/man_index.html.

Ниже описываются наиболее полезные модули:

erlang – содержит все встроенные функции Erlang. Большинство функций из этого модуля доступны без указания имени модуля, но к остальным нужно обращаться только по полному имени, с указанием модуля;

file – интерфейс к файловой системе, содержащий функции для работы с файлами;

io – интерфейс к стандартному серверу ввода/вывода. Содержит функции для чтения/записи файлов, в том числе стандартных устройств ввода/вывода;

lists – содержит функции для работы со списками;

math – модуль, содержащий стандартные математические функции;

string – содержит функции для работы со строками.

***

Мы рассмотрели основы последовательного программирования в Erlang. Более подробную информацию о функциях, модулях, ограничителях, условных выражениях, анонимных функциях и исключениях можно найти в справочном руководстве по Erlang: http://erlang.org/doc/reference_manual/part_frame.html. EOF

ВЛАДИМИР ГАКОВ, журналист, писатель-фантаст, лектор. Окончил физфак МГУ. Работал в НИИ. С 1984 г. – на творческой работе. В 1990-91 гг. – Associate Professor, Central Michigan University. С 2003 г. преподает в Академии народного хозяйства. Автор 8 книг и более 1000 публикаций

Компьютерный парк юрского периода

Его прообраз задумал испанский монах в XIV веке

Компьютеры начинались с громоздких электронных вычислительных устройств, которые обозначались с момента своего появления на свет английским словом computer

Официально компьютерная эра открылась 1 января 1946 года, когда в Пенсильванском университете членам комиссии американского министерства обороны был представлен первый электронный компьютер ENIAC (Electronic Numerical Integrator and Computer). Занимавшая целый зал тридцатитонная махина была детищем Джона Моучли (1907-1980) и Джона Преспера Эккерта (1919-1995), выполнивших заказ артиллерийского управления, которое остро нуждалось в средствах скоростного расчета таблиц стрельбы и бомбометания.

Вряд ли кто из присутствовавших на презентации военных чинов, да и ученых, отдавал себе отчет, что на их глазах зарождалась технологическая революция, изменившая облик цивилизации. Однако «первым в истории» компьютером ENIAC может быть назван лишь с оговорками – как и во всех случаях споров о приоритетах.

До потопа (информационного)

В начале XIV столетия в трактате Ars Magna («Великое искусство») испанский монах, поэт, философ и теолог Раймунд Луллий выдвинул идею логической машины, состоявшей из концентрических кругов с нанесенными на них буквенными символами, смысл которых разъяснялся в отдельных таблицах.

Век спустя идея начала приобретать материальные очертания. В 1623 году немецкий астроном Вильгельм Шикард (1592-1635) – друг, коллега и соотечественник великого Кеплера – соорудил замысловатый прибор, названный «часами со счетом». Этот 6-разрядный механизм был способен складывать и вычитать, а с помощью особых счетов на корпусе еще и умножать. Самой остроумной деталью машины был колокольчик, звонивший при «переборе», когда результат превышал «резервы памяти». Для начала XVII века, согласитесь, круто!

Но, как и многим светлым головам, Шикарду не повезло со временем рождения. В ту пору по немецким землям, лоскутным одеялом покрывавшим карту Европы, катком прошлась Тридцатилетняя война, и действующая модель и чертежи Шикарда были утеряны. Нашли их каким-то чудом только в 1935-м, после чего вновь потеряли и окончательно разыскали только спустя четверть века. В 1960 году по чертежам XVII века энтузиасты построили действующую модель.

В 1640-х годах счетной машиной «заболел» великий французский математик, физик и философ Блез Паскаль (1623-1662). Согласно легенде он впервые задумался над этим еще в девятилетнем возрасте, наблюдая за утомительными расчетами отца – сборщика налогов. Результатом размышлений гениального сына стал «паскалин» (Pascalin) – механизм, состоящий из шестеренок и связанных между собой колесиков с цифрами от 0 до 9, с помощью которых можно было складывать семизначные числа. Вычитать же агрегат не умел, да и вообще по многим параметрам уступал более простой системе Шикарда (о ее существовании французский ученый не знал). Тем не менее известность получила именно машина Паскаля.

И наконец в 1673 году (по некоторым источникам – в 1694-м) немецкий математик и философ Готфрид Вильгельм фон Лейбниц (1646-1716) изобрел ступенчатый калькулятор (Rechenmachine), способный производить арифметические действия с 12-значными числами. Возможности машины Лейбница для своего времени были потрясающими, а недостаток единственный – она требовала постоянного присутствия и вмешательства (на каждом этапе расчетов) пользователя. Зато не кто иной, как Лейбниц, первым додумался до двоичной системы изображения чисел!

По чертежам великого ученого парижский мастер Оливье построил действующую модель машины, которую после нескольких лет эксплуатации забросили куда-то на чердак, да там и забыли. Ее случайно обнаружили лишь в 1879 году, когда владельцы дома начали чинить протекавшую крышу.

Между тем уже шесть десятилетий (с 1820 года) успешно использовался арифмометр, который изобрел француз Шарль Ксавьер Тома де Кольмар (1785-1870). Эта машинка, занимавшая весь письменный стол, могла умножать и делить числа и являлась самым надежным калькулятором того времени, фактически первой массовой моделью.

Во втором десятилетии XIX века также увидело свет революционное изобретение соотечественника Кольмара Жозефа Мари Жаккарда (1752-1834) – механический ткацкий станок, способный выполнять заданный узор с помощью специально перфорированных карточек. Иначе говоря, первый «софт»!

Впрочем, если говорить все-таки о счетных машинах, то приоритет в изобретении программного обеспечения принадлежит не французу, а англичанке!

Компьютер на паровом ходу

Первую программистку в истории звали Адой Августой Кинг (1815-1852), в замужестве – графиня Лавлейс. Она была внебрачной дочерью знаменитого поэта Байрона и увлекалась математикой. Более того, долгие годы бескорыстно помогала (деньгами и расчетами) чудаку-изобретателю Чарлзу Бэббиджу (1791-1871), который всем заморочил голову безумным проектом какой-то дифференциальной машины, способной решать сложные математические уравнения.

Принципиальное устройство этого аппарата, обеспечивающего точность вычислений до восьмого знака после запятой, Бэббидж впервые описал в 1822 году, в возрасте 30 лет. А натолкнули его на мысль использовать для расчетов механическое устройство работы французского барона Гаспара де Прони. Во времена Великой французской революции тот получил задание правительства ввести метрическую систему во французский земельный кадастр, а это потребовало новых логарифмических и тригонометрических таблиц в небывалых дотоле объемах. Для их составления барон использовал единственную подручную ему «вычислительную машину» – людские ресурсы. Группа ученых ставила задачу, «среднее звено» следило за работой, а десятки счетчиков вели непосредственные расчеты, причем одни только складывали, другие только умножали...

В 1819 году Бэббидж был в Париже и там услыхал о предприятии де Прони. Познакомившись в местной Академии наук с результатами работы «вычислительной мануфактуры», англичанин решил всего-навсего заменить людей-счетчиков более эффективным механическим устройством. Сначала он построил простой механизм для составления таблиц полиномов – систему валиков и шестеренок, вращаемых с помощью рычага. После чего получил заказ от правительства на более мощное устройство для разработки навигационных таблиц.

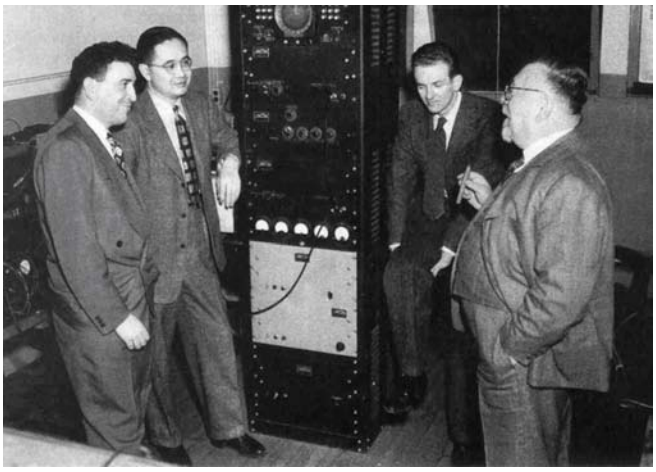
Это был грандиозный проект. По замыслу, машина, приводимая в действие паром, должна была занимать целую комнату и производить вычисления с точностью уже до 20-го знака! Спустя десять лет Бэббидж смог построить лишь один из ее блоков, и на этом дело застопорилось.

А теперь – внимание! В середине XIX века Бэббидж внятно описал целый ворох того, что в «железо» воплотили только век спустя – разве что терминологией пользовался подручной, сегодня вызывающей улыбку. Что касается сути, то улыбаться нечего – скорее изумиться. Тут и принцип центрального процессора («мельница»), и ввод программ («инструкций») с помощью перфорированных карт (первые программы «написала» упомянутая графиня Лавлейс), и блок памяти («склад»), и печатающее устройство, роль которого должен был выполнять печатный пресс. Единственное, чего не хватало паровому компьютеру, чтобы с полным правом называться прародителем современных ЭВМ, – это возможности хранения команд (stored-program) в том же ОЗУ, где содержатся исходные данные.

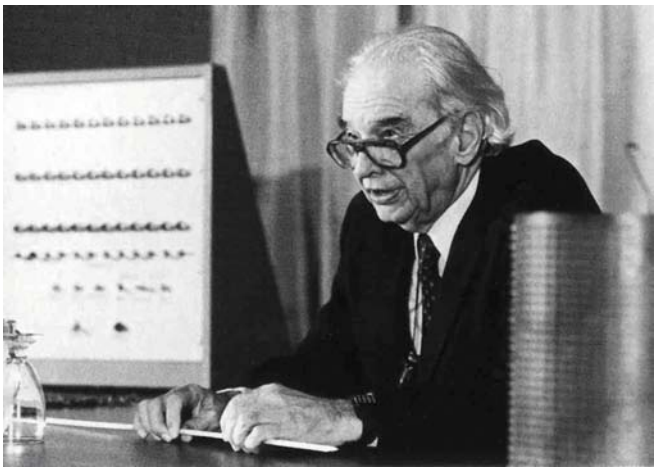
Быстродействие аналитической машины Бэббиджа, будь она построена, тоже вызвало бы у сегодняшнего пользователя улыбку: по замыслу изобретателя, одна операция сложения занимала 3 секунды, а умножения или деления – 2–3 минуты. Впрочем, и это было чистой фантастикой для эпохи, когда самым быстрым средством передачи информации все еще являлась упряжка добрых лошадей.

Машина Бэббиджа, способная, по словам Лавлейс, «ткать математические уравнения так же искусно, как жаккардовый станок – узоры из цветов и листьев», увы, так и осталась только в чертежах. Конец истории с его машиной столь же фантастичен, как и начало.

В 1989-1991 годах группа энтузиастов из Лондонского музея науки, используя современные материалы, но придерживаясь механических допусков, доступных во времена Бэббиджа, построила-таки придуманный им аппа-



Роберт Винер в MIT



Джон Винсент Атанасов

рат. И он работал, считая до 31-го знака после запятой!

Впередсмотрящие

Первую деревянную модель дифференциальной машины построили в 1843 году шведы отец и сын Шойтцы. Созданная Шойтцами конструкция успешно справлялась с уравнениями четвертого порядка.Два первых экземпляра машины купили обсерватория Дадли в Олбани, штат Нью-Йорк, и британское правительство. После чего ди-

Первую программистку звали Адой Августой Кинг, в замужестве — графиня Лавлейс. Она была внебрачной дочерью знаменитого поэта Байрона и увлекалась математикой

ректора обсерватории с треском уволили за экстравагантную покупку, а саму ее отправили пылиться на склад. Вторая же машина долго и надежно обслуживала более прагматичных англичан.

Последнее десятилетие «до потопа» – иначе говоря, до XX века, – связано также с именем американца Германа Холлерита (1860-1929), служащего Федерального агентства по переписи населения. На протяжении нескольких лет Холлерит пытался построить электромеханическую машину для оптимизации трудоемкого процесса обработки результатов переписи. И добился своего – в 1890 году такая машина, работавшая с использованием перфокарт, была построена и сразу же прошла боевое крещение. Если обработка результатов переписи населения в 1880 году потребовала семи лет и труда 1500 сотрудников, то изобретение Холлерита позволило его ведомству справиться всего за три года и усилиями 700 человек, заполнивших карточки на 63 миллиона американских граждан.

Результаты оказались настолько многообещающими, что спустя шесть лет Холлерит основал собственную фирму Tabulating Machine Company, главным направлением которой стали производство и продажа специального оборудования для перфорирования карточек. После многочисленных слияний и переименований компания в 1924 году окончательно сменила название на всем известное International Business Machines. Или кратко – IBM. Но это, как писали мои любимые Стругацкие, уже совсем другая история.

В первой половине XX века были заложены основы новой науки, название которой пришло позже, – кибернетики.

В 1936 году английский математик Алан Тьюринг (1912-1954) опубликовал статью, в которой доказал, что принципиально возможно создать универсальное цифровое вычислительное устройство, способное решить задачу любой степени сложности («машина Тьюринга»). Тьюринг предложил умозрительную схему – бесконечная бумажная лента с записанными символами и считывающая головка, которая с помощью последовательных запрограммированных операций была способна решить любую формализованную задачу. Чем не прообраз компьютера?

Собственно кибернетика родилась сразу же по окончании войны. Уже в 1946 году Джон Тьюки придумал термин «бит», а годом позже английский ученый Росс Эшби впервые ввел понятие «самоорганизующаяся система». Наконец, в 1948 году вышла революционная книга Норберта Винера (1894-1964) «Кибернетика, или Управление и связь в животном и машине», в которой были сформулированы основные положения новой науки об универсальных законах управления. Название ее было выведено Винером из греческого «кибернетос» – так древние греки называли лоцмана, навигатора, впередсмотрящего.

Динозавры компьютерной эры

Параллельно накоплению теоретического материала разворачивалась увлекательная гонка практиков: кому быть отцом современного компьютера. Однозначного победителя не определили до сих пор – первые модели компьютеров появились на свет почти одновременно в разных странах, а изобретатели часто узнавали о свершениях коллег с за-

Хорст Цузе и копия механической цифровой машины Z1

Вычислительная машина ABC

позданием. Одна из основных причин тому – война, и главное следствие ее – тотальная секретность.

Немцы не без оснований считают, что приоритет принадлежит инженеру Конраду Цузе (1910-1995). Еще в 1938 году выпускник Берлинского политехнического института на квартире родителей построил полностью программируемую механическую цифровую машину Z1. Она занимала площадь в 4 квадратных метра, действовала на основе двоичного кода, содержала блок памяти, а программа вводилась с перфорированной кинопленки. Модель была пробной, но через два года Цузе построил первый, как считают многие, электромеханический компьютер Z2.

Далее последовал Z3 (1941), состоявший из 2,5 тысяч телефонных реле (в этой машине впервые был реализован принцип программного управления), а в самом конце Второй мировой войны – усовершенствованная модель Z4. Тайно вывезенная из осажденного Берлина, она в разобранном виде пролежала три года в каком-то хлеву, после чего была доставлена в Цюрихскую высшую техническую школу.

У Цузе нашелся конкурент в Америке. За год до создания Z1 Джордж Стибитц (1904-1995) из Bell Telephone Laboratories разработал модель «двоичного сумматора» на телефонных реле. А в октябре 1940 года на заседании Американского математического общества он же продемонстрировал новую релейную (электромеханическую) машину – Complex Number Calculator, также претендующую на право называться первым цифровым компьютером. С помощью телетайпа, расположенного в зале заседаний (дело происходило в штате Нью-Гэмпшир), данные вводились в машину Стибитца, находящуюся в Нью-Йорке. А результаты вычислений передавались по телеграфу и выводились на печать в том же зале.

И только много десятилетий спустя обнаружился еще один американский «папа» ЭВМ – болгарин по происхождению Джон Винсент Атанасов (1903-1995), преподававший физику и математику в университете штата Айова. Атанасов независимо от Цузе совершил революционный переход к двоичной системе и вместе со своим студентом Клиффордом Берри в 1937-1939 годах построил прототип электронного цифрового компьютера ABC (Atanasoff Berry Computer). К 1942 году были созданы еще две ЭВМ, по образу и подобию которых ученый намеревался разработать более мощную машину, способную решать системы линейных алгебраических уравнений. Однако этому помешала война. Профессора призвали в армию, и работа была прервана на этапе отладки устройств ввода/вывода.

После демобилизации Атанасов узнал о первой демонстрации ENIAC и охладел к компьютерам настолько, что даже не поинтересовался, что из себя представлял аппарат Джона Моучли и Джона Преспера Эккерта.

А их машина между тем была до неприличия похожа на его собственную! В свое время Джон Моучли встречался с Атанасовым и позаимствовал у ABC ряд технических идей, которые и были использованы при создании его «первого компьютера». Однако упоминать об этом Моучли не счел нужным. Ответ на вопрос «Кто раньше?» был дан лишь в 1973 году, в результате судебного разбирательства между компаниями Sperry Rand, выкупившей патент на ENIAC, и Honeywell, которая оспаривала приоритет этого патента. Финальную точку поставил судья окружного суда Миннеаполиса Эрл Ларсон, юридически закрепивший за ABC право называться первым автоматическим цифровым компьютером.

Между прочим, еще в 1940-х годах его создатель безуспешно пытался заинтересовать своими идеями ведущие фирмы по производству механических счетных машин. Среди полученных Атанасовым формальных «отлупов»

«Наша компания никогда не станет осваивать электронный цифровой компьютер в своем производстве». Получен сей ответ был от… фирмы IBM!

Первый электронный компьютер ENIAC

Электромеханический монстр Harvard Mark I

особый интерес представляет один, заканчивавшийся решительной фразой: «Наша компания никогда не станет осваивать электронный цифровой компьютер в своем производстве». Получен сей ответ был от... фирмы IBM!

К тому времени руководство IBM окончательно склонилось к разработке альтернативного проекта – универсальной релейной цифровой машины, которую предложил Хоуард Эйкен (1900-1973), ничего не знавший о работах Атанасова. И в августе 1944 года электромеханический монстр Harvard Mark I (длиной 15 метров, высотой 2,5 метров и весом 5 тонн), состоявший из 3 миллионов узлов и 500 миль проводов и обошедшийся разработчикам примерно в $1 млн, заступил на трудовую вахту в Гарвардском университете, где проработал более 15 лет.

Эта махина производила операцию сложения менее чем за секунду, умножение совершала за 6 секунд, а деление – в два раза дольше. Сегодня даже элементарный калькулятор даст электронному динозавру сто очков вперед. Но лиха беда начало: компьютеры уже следующего десятилетия заставили скептиков прикусить языки.

Кстати, первые программы для машины Эйкена написала достойная продолжательница дела Ады Лавлейс – математик Грейс Марри Хоппер (1906-1992). Это была женщина во многих отношениях легендарная – автор популярных машинных языков (в частности, COBOL), президент ведущих компьютерных компаний и… контр-адмирал американских ВМС! Есть сведения, что знаменитая героиня научно-фантастического цикла Айзека Азимова о роботах – робопсихолог Сьюзен Келвин – списана с Грейс Хоппер.

Гиганты и мошки

Справедливости ради стоит отметить, что детище Моучли и Эккерта, создание которого обошлось в $400 тыс., на тот период было самой большой и мощной вычислительной машиной. Электронный гигант весил 30 тонн и состоял из 40 панелей, содержавших 18 тысяч ламп, 1500 реле и 70 тысяч резисторов и конденсаторов. Благодаря впервые примененным электронным триггерам ENIAC по быстродействию на три порядка опережал Mark I. Правда, электричества эта махина съедала также намного больше – 160 киловатт.

Кроме того, ENIAC страдал частыми сбоями, причины которых стали называть «жучками» (bugs) – термин, придуманный еще наладчиками электронных блоков радаров. Впрочем, «жучками» часто оказывались обыкновенные мотыльки – привлеченные теплом и светом, они заползали внутрь машины, что вызывало короткие замыкания.

Вслед за ENIAC Моучли и Эккерт создали UNIVAC (Universal Automatic Computer) – первую коммерческую ЭВМ и BINAC (Binary Automatic Computer) – первую машину, способную работать в режиме реального времени. А в Великобритании примерно тогда же появилась своя первая «большая» машина ACE (Automatic Computing Engine – «Ас»), и чуть позже сотрудники Манчестерского университета, куда перешел Алан Тьюринг, под его руководством создали машину Manchester Mark I (или MADAME – от Manchester Automatic Digital Machine) с рекордной в то время памятью.

Любопытно, что рвение английских разработчиков подстегнула самая обыкновенная профессиональная ревность к успехам американцев. А также «примкнувшего к ним» выдающегося математика Джона фон Неймана (1903-1957), эмигрировавшего с родителями из Венгрии (где он именовался Янушем Лайошем Нейманом) в США еще в 1930 году и в июне 1945-го разработавшего проект электронного компьютера EDVAC (Electronic Discrete Variable Computer).

В этой машине также имелось революционное новшество. Фон Нейман предложил вводить программу не на бумажной ленте, а с помощью перфокарты – прообраза будущего жесткого диска. Бесконечно далекий от каких бы то ни было коммерческих расчетов, математик опубликовал свои соображения в научном журнале, не потрудившись запатентовать идею, которой затем воспользовались и Моучли с Эккертом при постройке ENIAC. Вот и еще один претендент на право называться первым!

И, наконец, в 1947 году трое сотрудников американской компании Bell Telephone Laboratories – Уильям Шокли, Уолтер Браттейн и Джон Бардин – изобрели транзистор, давший возможность совершить решающий рывок вперед – к современному компьютеру. EOF

Грейс Хоппер, автор машинного языка COBOL

Джона Моучли (слева) и Джон Преспер Эккерт

Павел Александров:
«У «Системного администратора» нет альтернативы»

На вопросы «СА» отвечает бывший системный администратор, а ныне генеральный директор московской компании «КволиТек СП» и наш постоянный читатель

– Почему вы читаете «СА»?
– Потому что это интересно и полезно. Сначала читал, так как больше ничего полезного на русском языке не было. Нет, кроме шуток: не было нигде вменяемых статей по актуальным версиям программ на родном языке. А в английском я был ни бум-бум.

Сейчас для меня ваш журнал – источник готовых решений различных задач: развёртывание IP-АТС, инвентаризация, мониторинг серверов и прочее. Не надо думать, взял журнал, прочёл, сделал. Очень удобно и время экономит. Опять же всегда что-нибудь новое узнаю. Иногда бывает, что прочтёшь, задумаешься, а зачем это тебе? Но проходит время, кто-то из подчинённых сталкивается с какой-то проблемой, и ты понимаешь, что её решение уже нашёл в «СА», а значит, знаешь, как поступить.

– Какая изюминка, на ваш взгляд, должна быть у «СА»?
– Изюминка уже есть: у вашего журнала нет альтернативы. У новостных изданий есть, у тестовых и обзорных тоже есть. «СА» – это единственный отраслевой журнал для айтишников.

– Какие новые рубрики вы бы хотели увидеть в журнале?
– Я бы добавил что-нибудь вроде «Навыки общения с пользователями: вежливость и этикет для системных администраторов». А если серьёзно, то рубрику «Администрирование 1С» надо либо выкинуть, либо переработать. И назвать иначе: «Изучаем 1С», «Доработка 1С», «Разработка в 1С» (**от редакции** – по просьбе читателей уже переработали).

– Кто ваши друзья, чем увлекаетесь?
– Мои друзья – программисты, системные администраторы, рекламщики, маркетологи, геологи. А также музыканты и даже есть один повар. Среди них много туристов. С большинством мы вместе учились в геолого-разведочном институте. Сам я, зачастую с кем-то из друзей, ползаю по пещерам, хожу по лесам, горам, дорогам нашей страны, сплавляюсь по её рекам на байдарках. Катаюсь на велосипеде. Езжу по интересным местам на автомобиле, посещаю краеведческие музеи. Стреляю из лука по мишеням. Играю на губной гармошке. Время от времени делаю всякие разные штуки из железа или дерева. Жадно изучаю новое. Что угодно: полезное, бесполезное.

Люблю вкусное пиво, невкусное — не люблю. Физически зависим от чая

– Как сложилась ваша жизнь?
– Я женат уже почти десять лет, воспитываю дочь. Работаю аутсорсером. Начинал системным администратором. Это случилось еще на первом курсе института. Я был единственный, кто разбирался в компьютерах, поэтому меня нагрузили поддержкой сети студенческого профкома. Затем стал программистом и постановщиком задач в одном лице. После руководил разработчиками и системными администраторами. Сейчас руковожу и теми, и другими на пару с партнёром.

Свободное от этой вредной работы время стараюсь проводить с семьёй, благо дочка уже в том возрасте, когда её везде можно брать с собой.

– К чему вы стремитесь?
– Чертовски сложный вопрос! Вперед и вверх, вероятнее всего. EOF

Редакционнная 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 **копия заполненного и оплаченного бланка, отправленная в редакцию по факсу: (495) 628-8253, (доб. 120) или на emil: 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 2010 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 2010 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 агенства **«Роспечать»**

> Подписной индекс годовой – **88099**, полугодовой – **87836**
Объединенный каталог **«Пресса России»**
Адресный каталог **«Подписка за рабочим столом»**
Адресный каталог **«Библиотечный каталог»**

> Альтернативная подписка агенства:
«Интер-Почта» (495) 500-00-60, курьерская доставка по Москве
«Вся Пресса» (495) 787-34-47
«Курьер-Прессервис»
«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)

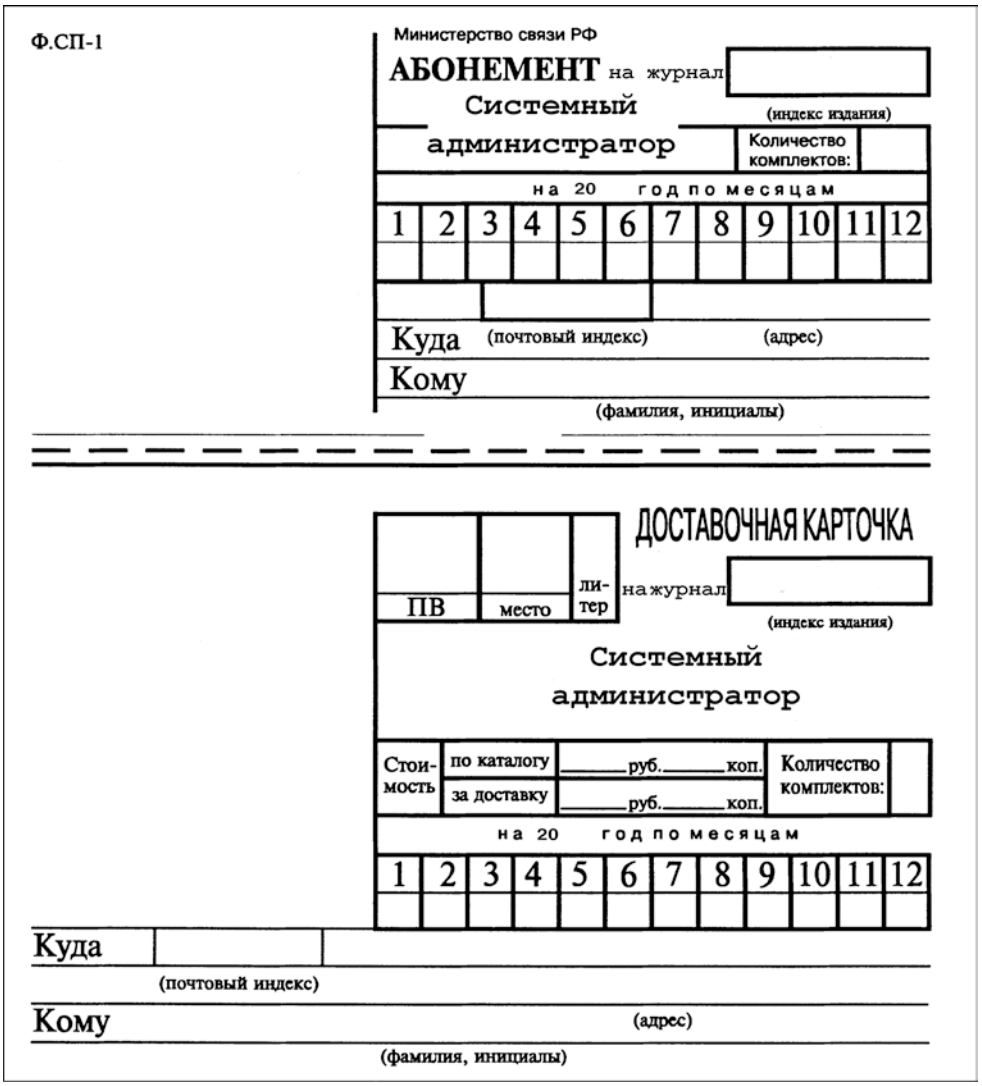
> **Узбекистан** – по каталогу российские издания через агентство по распространению печати **«Davriy nashrlar»** (7000029, г. Ташкент, пл. Мустакиллик, 5/3, офис 33)

> **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)

> **Украина** – 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 20 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 20 год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда (почтовый индекс)

Кому (адрес)

(фамилия, инициалы)

Подписные индексы:

20780*
+ диск с архивом статей 2009 года

81655**
без диска

по каталогу агентства «Роспечать»

88099*
+ диск с архивом статей 2009 года

87836**
без диска

по каталогу агентства «Пресса России»

* Годовой
** Полугодовой
*** Диск вкладывается в февральский номер журнала, распространяется только на территории России

Системный администратор

УЧРЕДИТЕЛИ ИЗДАНИЯ
Частные лица

Генеральный директор
Владимир Положевец
Главный редактор
Галина Положевец
chief@samag.ru
Шеф-редактор
Наталья Хвостова
sekretar@samag.ru
Технический директор
Владимир Лукин
Главный редактор электронного приложения «Open Source»
Дмитрий Шурупов
osa@samag.ru
Дизайн-макет
Марина Рязанцева
Дмитрий Бессонов
Иллюстрации
Виктор Чумачев

Над номером работали:
Алексей Барабанов, Александр Емельянов, Олег Щербаков, Кирилл Сухов

Рекламная и PR-служба
Дарья Зуморина, reklama@samag.ru,
Полина Гвоздь, expo@samag.ru,
тел./факс: (495) 628-82-53 (доб.120)

Распространение
Светлана Зобова
(495) 628-82-53 (доб.120)

Адрес редакции
107045, г. Москва, Ананьевский переулок, дом 4/2, стр.1,
тел./факс: (495) 628-82-53 (доб.120)
Сайт журнала: www.samag.ru

Издатель
ООО «С 13»

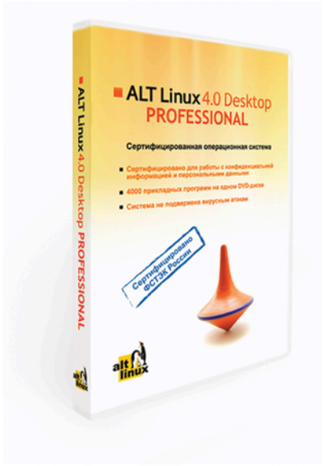
Отпечатано в типографии
ООО «Периодика»
Тираж 17000 экз.
Тираж электронной версии 62000 экз.

Сертифицированные продукты ALT Linux

Для кого предназначены сертифицированные продукты?

- Для **организаций**, которым необходимо иметь **сертифицированное ПО**. Это многие государственные учреждения, оборонные предприятия и т.д.;
- Для **организаций**, работающих **с конфиденциальной информацией и персональными данными**. Под эту категорию попадают практически все фирмы, имеющие базу данных паспортов, номеров сотовых телефонов и т.п. (туристические фирмы, страховые компании, банки и т.д.), фирмы, проводящие анкетирование.

ALT Linux 4.0 Desktop Professional
сертифицированный продукт для рабочих станций



ALT Linux 4.0 Desktop Professional сертифицирован Федеральной службой по техническому и экспортному контролю (ФСТЭК России).
Сертификат соответствия **№1649 от 23 июля 2008**:
- Классификация по уровню контроля отсутствия недекларированных возможностей (НДВ) — **4 уровень**.
- Показатели защищённости от несанкционированного доступа к информации (СВТ) — по **5 классу защищённости**.

ALT Linux 4.0 Desktop Professional — это:
- Удобная в работе операционная система, дающая пользователю возможность решать обычные задачи, не опасаясь вирусов и не затрачивая время на поиск нужных прикладных программ в сети Интернет и на полках магазинов;
- Дружественная программа установки, работа с которой будет особенно приятна начинающим пользователям;
- ALTerator — интуитивно понятный инструмент настройки и управления системой.

Рекомендуемая розничная цена: 3800 руб.

ALT Linux 4.0 Server Edition
сертифицированный продукт для серверов

Всё, что можно сделать по настройке сервера без вмешательства пользователя, уже реализовано в дистрибутиве **ALT Linux 4.0 Server Edition**.
ALT Linux 4.0 Server Edition сертифицирован Федеральной службой по техническому и экспортному контролю (ФСТЭК России).
Сертификат соответствия **№1501 от 8 ноября 2007**:
- Классификация по уровню контроля отсутствия недекларированных возможностей — **4 уровень**.
- Показатели защищённости от несанкционированного доступа к информации — по **5 классу защищённости**.

ALT Linux 4.0 Server Edition — серверный дистрибутив с широким спектром возможностей, включающий комплект готовых решений для актуальных задач организации: построения корпоративной сети и среды обмена информацией. Простые веб-интерфейсы управления, включённые в дистрибутив, позволяют существенно ускорить развёртывание корпоративного сервера.

Рекомендуемая розничная цена: 22000 руб.

www.altlinux.ru
По вопросам приобретения: zakaz@altlinux.ru

alt linux